V Jornadas Nacionales de Investigación en Ciberseguridad

Cáceres 5-7 de junio

Editores: Andrés Caro Lindo, Luis Javier García Villalba, Ana Lucila Sandoval Orozco

Universidad de Extremadura
Servicio de Publicaciones

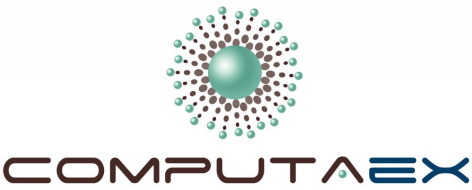
INSTITUTO NACIONAL DE CIBERSEGURIDAD

Actas de las

V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC 2019)

Junio 5–7, 2019

Cáceres, España



Editores:

Andrés Caro Lindo, Luis Javier García Villalba, Ana Lucila Sandoval Orozco

Universidad de Extremadura. Servicio de Publicaciones

Actas de las V Jornadas Nacionales de Ciberseguridad
Junio 5–7, 2019, Cáceres, España

Web de la Conferencia:
https://2019.jnic.es

Actas disponibles en:
https://2019.jnic.es/Actas_JNIC2019.pdf

Está permitida la descarga y la reproducción total o parcial de esta obra y su difusión siempre y cuando sea para uso personal o académico. Los derechos de autor de cada contribución individual corresponden a sus autores.

Editores:
Andrés Caro Lindo, Luis Javier García Villalba, Ana Lucila Sandoval Orozco
Junio 2019
ISBN *978-84-09-12121-2*

Diseño de Portada:
José Carlos Sancho Núñez

Publicado por:
Universidad de Extremadura. Servicio de Publicaciones
C/ Caldereros, 2 - Planta 3a. 10071 Cáceres (España)
Tel. 927 257 041; Fax 927 257 046
E-mail: publicac@unex.es http://www.unex.es/publicaciones

Bienvenida del Comité Organizador

Las V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) se celebran en Cáceres, del 5 al 7 de junio de 2019, organizadas por la Universidad de Extremadura (UEx), la Universidad Complutense de Madrid (UCM) y la fundación COMPUTAEX. Por primera vez estas jornadas están organizadas por más de una institución. Un buen espejo donde comprobar que la simbiosis entre grupos de diferentes universidades converge en puntos en común, aspecto fundamental en la filosofía de las propias JNIC desde su primera edición. La colaboración del Instituto Nacional de Ciberseguridad (INCIBE) en la realización de estas Jornadas se presenta también como fundamental para garantizar su éxito.

La V edición de estas Jornadas suponen su consolidación en el panorama nacional, después de la ilusión despertada en la edición inicial, ubicada en León (2015), las etapas de asentamiento vividas en Granada (2016) y Madrid (2017), y el fortalecimiento de la edición anterior en San Sebastián (2018).

Durante estos tres días, el programa previsto incluye trabajos de investigación en ciberseguridad relacionados con la detección de intrusiones, la monitorización de eventos de seguridad, la prevención, las políticas de seguridad, ataques, vulnerabilidades, análisis forense, cifrado... En segundo lugar, también se presentan trabajos de formación e innovación educativa. Por último, como en años anteriores, se continua con la línea de transferencia tecnológica, donde empresas e instituciones presentan retos científicos a los diferentes grupos de investigación. Todo ello combinado con actividades sociales para dar a conocer la gastronomía, naturaleza y cultura de Extremadura en general, y Cáceres en particular.

En esta edición de las JNIC, se han recibido 83 trabajos, de los cuales finalmente se han admitido 62 para su presentación en las jornadas (33 en formato de comunicación oral y 29 en formato de póster), además de dos trabajos de estudiantes, premiados como mejor Tesis y Trabajo Fin de Máster relacionados con la ciberseguridad. Como novedad en esta edición, se cuenta con Special Issues de revistas indexadas en el JCR en posiciones relevantes, donde los mejores artículos serán invitados a enviar versiones extendidas. Esta edición también presenta un reto CTF (Capture The Flag) dirigido a estudiantes universitarios y pre-universitarios, orientado a la detección de talento joven y, sobre todo, a fomentar la cultura de la ciberseguridad entre nuestros jóvenes.

Desde el comité organizador queremos agradecer a todos los que han hecho posible esta edición, desde los autores, asistentes, conferenciantes, revisores de los trabajos, miembros de los distintos comités, presidentes de los diferentes programas (Jesús Esteban Díaz Verdejo en el programa de investigación; Ana Isabel González Tablas en el programa de formación e innovación educativa; Juan Díez en el programa de transferencia), patrocinadores, voluntarios de la organización... Nuestro más sincero agradecimiento a todos ellos. Para finalizar, esperamos que todos los esfuerzos realizados para presentar la edición 2019 de las jornadas ayuden a consolidar a las JNIC como el mejor punto de encuentro en ciberseguridad.

Andrés Caro, Luis Javier García Villalba, José Luis González
General Chairs de las JNIC 2019

Índice general

Comité Ejecutivo

Marta Beltrán Pardo	Universidad Rey Juan Carlos
Andrés Caro Lindo	Universidad de Extremadura
Juan Díez González	INCIBE
Eduardo Fernández-Medina Patón	Universidad de Castilla-La Mancha
Luis Javier García Villalba	Universidad Complutense de Madrid
Gregorio Martínez Pérez	Universidad de Murcia
Guillermo Suárez-Tangil	King's College London
Urko Zurutuza Ortega	Mondragon Unibertsitatea

Comité Organizador

General Chairs

Andrés Caro Lindo	Universidad de Extremadura
Luis Javier García Villalba	Universidad Complutense de Madrid
José Luis González Sánchez	COMPUTAEX-CénitS

Program Chair (Investigación)

Jesús Esteban Díaz Verdejo	Universidad de Granada

Program Chair (Formación e Innovación Educativa)

Ana Isabel González-Tablas	Universidad Carlos III de Madrid

Program Chair (Transferencia Tecnológica)

Juan Díez González	INCIBE

Financial Chair

José Carlos Sancho Núñez	Universidad de Extremadura

Arrangement Chairs

Mar Ávila Vegas	Universidad de Extremadura
Alberto Bravo Gómez	Universidad de Extremadura
Pablo García Rodríguez	Universidad de Extremadura

Publication Chairs

Esteban Alejandro Armas Vega	Universidad Complutense de Madrid
Edgar González Fernández	Universidad Complutense de Madrid
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Miguel Sánchez Cabrera	Universidad de Extremadura

Publicity Chairs

Daniel Povedano Álvarez	Universidad Complutense de Madrid
Carlos Quinto Huamán	Universidad Complutense de Madrid
Felipe Lemus Prieto	COMPUTAEX-CénitS
Juan Pedro Torres Muñoz	Universidad de Extremadura

Webmaster

Antonio López Vivar	Universidad Complutense de Madrid

Comité de Programa de Investigación

Cristina Alcaraz	Universidad de Málaga
Félix Barrio	INTECO
Marta Beltrán Pardo	Universidad Rey Juan Carlos
Jorge Blasco Alís	Royal Holloway, University of London
Guillermo Calvo Flores	INCIBE
José Camacho	Universidad de Granada
Carmen Cámara	Universidad Carlos III de Madrid
Andrés Caro Lindo	Universidad de Extremadura
Jenny Alexandra Cifuentes Quintero	Universidad de la Salle
João José Costa Gondim	Universidad de Brasilia
Jesús Esteban Díaz Verdejo	Universidad de Granada
Rafael María Estepa Alonso	Universidad de Sevilla
Enaitz Ezpeleta	Mondragon Unibertsitatea
Eduardo Fernandez-Medina	Universidad de Castilla-La Mancha
Luis Javier García Villalba	Universidad Complutense de Madrid
Iñaki Garitano	Mondragon Unibertsitatea
Hugo Gascón	TU Braunschweig
Manuel Gil Pérez	Universidad de Murcia
Joao Gondim	Universidad de Brasilia
Félix Gómez Mármol	Universidad de Murcia
Alessandra Gorla	IMDEA Software Institute
Luis Hernández	Consejo Superior de Investigaciones Científicas
Julio César Hernández Castro	University of Kent
Mikel Iturbe	Mondragon Unibertsitatea
Nicolás Macia	Universidad Nacional de la Plata
José Márquez	Universidad del Norte
Srdjan Matic	IMDEA Software Institute
Héctor Menéndez	University College London
Lia Molinari	Universidad Nacional de la Plata
Alfonso Muñoz	BBVA Next
Robson de Oliveira Albuquerque	Universidad de Brasilia
Jesus Olivares Mercado	Instituto Politécnico Nacional
Hector Pérez-Meana	Instituto Politécnico Nacional
Sergio Pastrana	Universidad Carlos III de Madrid
Pedro Peris López	Universidad Carlos III de Madrid
Ricardo J. Rodríguez	Centro Universitario de la Defensa de Zaragoza, Academia General Militar

Fernando Román Muñoz	INDRA
Gabriel Sánchez	Instituto Politécnico Nacional
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Fermín J. Serna	Google
Alejandra Silva Trujillo	Universidad Autónoma de San Luis Potosí
José Soler	Technical University of Denmark
Guillermo Suárez-Tangil	King's College London
Rafael Timóteo de Sousa Júnior	Universidad de Brasilia
Ricardo Villanueva Polanco	Universidad del Norte
Urko Zurutuza	Mondragon Unibertsitatea

Comité de Programa de Formación e Innovación Educativa

Mar Ávila Vegas	Universidad de Extremadura
César Cáceres Taladriz	Universidad Rey Juan Carlos
Miguel Carriegos Vieira	Universidad de León
Noemí DeCastro García	Universidad de León
Félix Jesús García Clemente	Universidad de Murcia
José Antonio Gómez Hernández	Universidad de Granada
Ana I. González-Tablas	Universidad Carlos III de Madrid
Andrés Marín López	Universidad Carlos III de Madrid
José Carlos Sancho Núñez	Universidad de Extremadura
Ana Lucila Sandoval Orozco	Universidad Complutense de Madrid
Adriana Suárez Corona	Universidad de León

Comité de Programa de Transferencia

Marcos Arjona	ElevenPaths
Juan Díez	INCIBE
Fernando Román Muñoz	INDRA
Fermín J. Serna	Google
Víctor Villagrá	Universidad Politécnica de Madrid

Resúmenes de las Comunicaciones

Sesión I: Detección de Intrusiones

Presidente de Sesión:

Víctor Villagrá (*Universidad Politécnica de Madrid*)

Miércoles 5, 10:30 – 11:30 h

DeepConfusables: mejorando la detección de ataques basados en codificación Unicode

Alfonso Muñoz Muñoz, José Ignacio Escribano Pablos, Miguel Hernández Boza

---

En los últimos 5 años ha habido un interés creciente en el uso de la inteligencia artificial en el mundo de la ciberseguridad, especialmente en el ámbito defensivo y de detección de patrones, aunque, más sorprendente, es su intento de aplicación a seguridad ofensiva: pentesting, exploiting, etc. Nuestro trabajo se engloba en esta última tendencia revisando la utilidad del deep learning y transfer learning en la mejora de ataques basados en caracteres Unicode. Como resultado se publica el mejor diccionario de confusables existente a día de hoy, utilizable a través de nuestra herramienta DeepConfusables, y se revisan casos de uso, servicios importantes en Internet (Telegram, Whatsapp, Signal, Skype o Turnitin), donde las contramedidas a este tipo de ataques no están suficientemente bien resueltas, por ejemplo, phishing o productos de detección de autoría.

Evaluación de algoritmos de clasificación para la detección de ataques en red sobre conjuntos de datos reales: UGR'16 dataset como caso de estudio

Ignacio Díaz Cano, Roberto Magán Carrión

La evaluación y rendimiento de sistemas de detección de intrusiones normalmente se realiza mediante el empleo de conjuntos de datos previamente obtenidos dentro del mismo contexto, entorno y aplicación en donde se implantará el sistema final. En concreto, para NIDS, existe un gran número de soluciones al respecto que, sin embargo, basan su rendimiento en conjuntos de datos no adecuados, bien por que están obsoletos, o por su representatividad, o realismo, entre otros. En el presente trabajo se introduce la problemática anterior y se presenta un estudio y evaluación de sistemas NIDS basados en algoritmos tradicionales de clasificación supervisada utilizando, ahora sí, conjuntos de datos adecuados como el recientemente creado conjunto de datos de red UGR'16.

HIDS by signature for embedded devices in IoT networks

Bruno Vieira Dutra, João F. de Alencastro, Francisco Lopes de Caldas Filho, Lucas Mauricio Castro E Martins, Rafael Timoteo de Sousa Júnior, Robson de Oliveira Albuquerque

Cybersecurity in the Internet of Things (IoT) has become a major concern due to the huge number of vulnerable devices and the difficulty for the IoT middleware to completely block the interactions of these devices with other entities outside the IoT realm. Hence, one possible way to protect IoT is to enhance smart devices with intrusion detection capabilities considering their limited resources. With such assumptions in mind, this paper describes a host-based intrusion detection system (HIDS) by signature for IoT smart devices. In this HIDS the attack signatures remain in a central controller on the cloud, which is periodically consulted by the hosts/devices. The proposed system takes actions defined by the administrator to prevent vulnerable IoT devices to be compromised and thus to join botnets. In addition, the proposed system is also able to notify the IoT middleware about potential failure indicators.

Metodología para la detección de Botnets en la nube mediante técnicas de optimización por medio Grid-Search

David González-Cuautle, Gabriel Sánchez-Pérez, Aldo Hernández-Suárez, Ana Lucila Sandoval Orozco

En los recientes años las botnets se han convertido en una de las amenazas más serias para todo aquello que se encuentra en la nube debido a la gran implementación de servicios desplegados e información que circula en la misma. La dependencia de las infraestructuras y redes virtualizadas en la nube introduce una gran cantidad de riesgos y vulnerabilidades tales como la denegación de Servicio (DoS), correo no deseado, phishing, filtración de información y la detección apropiada de anomalías. Una solución para la detección rápida y eficaz de botnets es el análisis del flujo red por medio de aprendizaje automático para diferenciar entre tráfico de red benigno y malicioso. En este trabajo, se propone una metodología para comparar diferentes conjuntos de datos y mostrar a su vez el rendimiento de los algoritmos de aprendizaje supervisado más utilizados en el estado del arte y su optimización mediante hiper-parámetros con Grid-Search.

Sesión II: Monitorización de eventos de seguridad

Presidente de Sesión:

Eduardo Fernández-Medina (*Universidad de Castilla - La Mancha*)

Miércoles 5, 12:15 – 14:00 h

Detectando anomalías de integridad y veracidad en fuentes de datos IIoT

Iñaki Garitano, Mikel Iturbe, Enaitz Ezpeleta, Urko Zurutuza

El panorama de la seguridad en entornos industriales ha cambiado completamente en las últimas décadas. Desde las configuraciones primitivas iniciales, las redes industriales han evolucionado hacia entornos masivamente interconectados, desarrollando así el paradigma de Internet Industrial de las Cosas (IIoT). En IIoT, multiples dispositivos heterogéneos colaboran mediante la recopilación, el envío y el procesamiento de datos. Estos entornos controlados han hecho posible el desarrollo de servicios de valor agregado basándose en los datos, los cuales mejoran la operación de los procesos industriales. Así, la verificación de los datos entrantes resulta indispensable, debido a que las decisiones tomadas serán erróneas si los datos en los que se basan no son correctos. En este capítulo, presentamos un sistema de detección de anomalías IIoT (ADS), que audita la integridad y la veracidad de los datos recibidos de las conexiones entrantes. Para este fin, el ADS incluye datos de campo (magnitudes físicas basadas en datos) y metadatos de conexión (intervalo entre las conexiones entrantes y el tamaño del paquete) en el mismo modelo de detección de anomalías. El enfoque se basa en el control estadístico multivariante de procesos y se ha validado utilizando datos reales de una planta de distribución de agua.

Metodología supervisada para la obtención de trazas limpias del servicio HTTP

Jesús Díaz Verdejo, Rafael Estepa Alonso, Antonio Estepa Alonso, Germán Madinabeita

Disponer de datos adecuados para el entrenamiento, evaluación y validación de sistemas de detección de intrusos basados en anomalías representa un problema de índole práctica relevante. Las características requeridas para los datos plantean una serie de retos contrapuestos entre los que destaca la necesidad de disponer de un volumen significativo de datos reales que no contenga instancias de ataques. Esto implica un proceso de limpieza y supervisión que puede resultar muy costoso si se realiza manualmente. En este trabajo planteamos una metodología para automatizar en lo posible la adquisición y acondicionamiento de trazas del servicio HTTP para la detección de ataques basada en URI. Esta metodología se aplica con buenos resultados sobre una traza real como caso de estudio.

Extracción de conocimiento a partir de fuentes de datos reales procedentes de la monitorización de eventos de seguridad

Alberto Bravo Gómez, José Carlos Sancho Núñez, Andrés Caro Lindo

La monitorización de eventos de seguridad es una práctica cada vez más utilizada por las organizaciones, para detectar amenazas, vulnerabilidades y estimaciones de riesgos de seguridad. La gestión de eventos e información relacionada con la seguridad se realiza mediante sistemas comerciales que facilitan toda la información, procesando diferentes fuentes de datos. La posibilidad de desarrollar modelos alternativos que, en base a las mismas fuentes de datos, proporcionen información complementaria a los sistemas comerciales se plantea como un reto novedoso e interesante, no solo para las organizaciones, sino también para la comunidad científica. Este artículo presenta un novedoso sistema de extracción de conocimiento basado en la monitorización de eventos de seguridad que permite complementar la información de los sistemas comerciales y también predecir conductas futuras de riesgo que permitan anticiparse a situaciones de posibles riesgos.

Categorización automática de la severidad de un ciberincidente. Un caso de estudio mediante aprendizaje automático supervisado

Noemí DeCastro-García, Mario Fernández-Rodríguez, Ángel Luis Muñoz Castañeda

En este trabajo se presenta un caso de estudio en el que se construye un modelo que permite categorizar automáticamente la severidad de un incidente de ciberseguridad.Se ha aplicado la categorización a tres tipos de eventos diferentes. La metodología utilizada sigue las fases de la ciencia de datos. El modelo se ha realizado mediante técnicas de aprendizaje automático supervisado sobre una base real de registros de ciberincidentes. Los resultados muestran que los algoritmos ensamblados y aquellos basados en diagramas de construcción lógicos aportan los mejores modelos predictivos de clasificación, obteniendo un ajuste para cada evento con una tasa de acierto superior al 99

OSINT is the next Internet goldmine: Spain as an unexplored territory

Javier Pastor Galindo, Pantaleone Nespoli, Félix Gómez Mármol, Gregorio Martínez Pérez

Phenomenons like Social Networks, Cloud Computing or Internet of Thing are unknowingly generating unimaginable quantities of data. In this context, Open Source Intelligence (OSINT) exploits such information to extract knowledge that is not easily appreciable beforehand by the human eye. Apart from the political, economic or social applications OSINT may bring, there are also serious global concerns that could be covered by this paradigm such as cyber crime and cyber threats. The paper at hand presents the current state of OSINT, the opportunities and limitations it poses, and the challenges to be faced in the future. Furthermore, we particularly study Spain as a potential beneficiary of this powerful methodology.

Evaluación de características de fuentes de datos en ciberseguridad para su aplicabilidad a algoritmos de aprendizaje basados en redes neuronales

Xavier Larriva Novo, Mario Vega Barbas, Víctor Villagrá, Mario Sanz

Los algoritmos de inteligencia artificial ya tienen un papel protagonista en el ámbito de la ciberseguridad y la detección de ataques, pudiendo presentar mejores resultados en algunos escenarios que sistemas de detección de intrusiones clásicos como Snort o Suricata. Dentro de los algoritmos de aprendizaje automático, este artículo se centra en la evaluación de la aplicabilidad de uno de los más populares: las redes neuronales. Para ello, se plantea en primer lugar una categorización para datasets de ciberseguridad que divide sus características en varios grupos. Haciendo uso de dicha división, este trabajo busca determinar qué modelo de red neuronal (multicapa o recurrente), función de activación y algoritmo de aprendizaje arroja valores más elevados de precisión en función del grupo de características del que se disponga. Asimismo, y con estos resultados, se pretende deducir qué tipo de características presentes en un dataset son más relevantes y representativas para la detección y así, hacer más ligera la carga computacional de la red.

Sesión III: Formación e innovación educativa

Presidente de Sesión:

Ana Lucila Sandoval Orozco (*Universidad Complutense de Madrid*)

Miércoles 5, 15:30 – 17:00 h

Investigación en Ciberseguridad: Una propuesta de innovación docente basada en el role playing

Noemí DeCastro-García, Ángel Luis Muñoz Castañeda, Miguel Carriegos

El rápido crecimiento de las amenazas digitales, así como la aparición constante de retos tecnológicos, ponen de manifiesto la necesidad de incluir el desarrollo de competencias investigativas en los programas de formación en Ciberseguridad. De esta manera, el alumnado podrá adquirir habilidades que son fundamentales para mantener a la sociedad en la vanguardia del conocimiento. En este artículo se presenta una experiencia innovadora para la formación en investigación científica en el campo de Ciberseguridad. La metodología docente se basa en la creación de un entorno integral de investigación simulado y se ha diseñado en base a diferentes estándares internacionales potenciando la colaboración entre el sector académico y profesional. Además, la propuesta se ha implementado durante tres cursos escolares en el Master de Investigación en Ciberseguridad de la Universidad de León, tanto en modalidad online como presencial, presentando buenos resultados, así como posibilidades de mejora.

Diseño de actividad lúdica orientada a la enseñanza de métodos y técnicas de OSINT

Miguel Páramo, Víctor Villagrá

En este documento se expone el diseño de una actividad docente aplicada a la formación en ciberseguridad que incorpora de manera innovadora componentes narrativos (storytelling) y aspectos lúdicos (gamification) en una modalidad de ejercicio de “captura la bandera” orientado al aprendizaje de técnicas y métodos transversales propios del campo de Inteligencia de Fuentes Abiertas (OSINT). La actividad propuso un caso de estudio criminalístico ficticio con el objetivo de que, de manera exploratoria y aplicada, el alumno entrene y adquiera dichas habilidades transversales bajo el marco de la asignatura “Gestión de Riesgos y Operaciones de Ciberseguridad” del Máster en Ciberseguridad de la Universidad Politécnica de Madrid. La actividad se desarrolló en el curso académico 2018-2019 con motivo del estudio de operaciones preventivas en el campo de la ciberseguridad; en concreto la ciber-inteligencia y la vigilancia digital.

MOOC “Investigación en Informática Forense y Ciberderecho”, experiencia y resultados

Andrés Caro Lindo, José Carlos Sancho Núñez, Mar Ávila Vegas, Miguel Sánchez Cabrera

Las universidades utilizan los MOOC como escaparate para atraer alumnos a su oferta de títulos académicos. Sin embargo, su alto índice de abandono y la gran cantidad de recursos necesarios para su puesta en marcha, ponen en duda el beneficio de este tipo de cursos masivos. Esta contribución pretende acercar la motivación, experiencia y resultados del MOOC “Investigación en Informática Forense y Ciberderecho” celebrado por la Universidad de Extremadura en la plataforma Miríada X, entre los meses de octubre y diciembre del año 2018. Se exponen las claves que han llevado a este MOOC a tener una tasa de finalización del 25,65

Sesión IV: Prevención y políticas de seguridad

Presidente de Sesión:

Pedro García-Teodoro (*Universidad de Granada*)

Jueves 6, 10:00 – 11:30 h

Design and Development of a Translation and Enforcement Module for Cybersecurity Policies

Fernando Monje Real, Víctor Villagrá

Nowadays, cyber attacks constitute a bigger threat to organizations than before, given the higher sophistication of those attacks, their growing propagation velocity and the increase of their destructive capabilities. This problem requires solutions capable of answering in real time and automatically. The proposed solution is the development of a system capable of translating a high-level security policy designed by an organization into another low level policy, so that it can be interpreted by the elements of the network in charge of the security. In such a manner, it is possible to design the security policy independently of the network elements. The policy is implemented in real time accordingly to the dynamic risk of the organization. This risk calculation will be carried out using the data obtained by an Intrusion Detection System (IDS) monitoring the organization's network. System efficiency will be validated with two virtualized scenarios using different network topologies.

CyberSPL: Plataforma para la verificación del cumplimiento de políticas de ciberseguridad en configuraciones de sistemas usando modelos de características

Ángel Jesús Varela Vaca, Rafael Gasca, Rafael Ceballos, Pedro Bernáldez Torres

---

Los ataques de ciberseguridad se han convertido en un factor muy relevante que pueden contravenir el cumplimiento de las políticas de ciberseguridad de las empresas y organizaciones. Dichos ataques pueden estar provocados en gran medida por una ausencia de configuraciones de seguridad o de valores por defecto en la configuración de productos y sistemas. La complejidad en la configuración de productos y sistemas es un reto en la industria del software. En este artículo proponemos una plataforma, Cybersecurity Software Product Line (CyberSPL), basado en la metodología de diseño de líneas de productos de tal manera que a través de la definición de modelos de características podamos agrupar patrones de configuraciones de aplicaciones y sistemas relacionados con la ciberseguridad. Mediante el análisis automatizado de estos modelos permitiríamos la diagnosis de los posibles problemas en las configuraciones de seguridad y por tanto evitarlos. Como soporte para dicha plataforma se ha implementado una solución multiusario y multiplataforma que permite definir un catálogo de modelos de características público o privado. Además se han integrado mecanismos para determinar todas las configuraciones de un modelo, detectar si una configuración es correcta o no, además de diagnosticar las

causas de fallos dada una configuración determinada. Para validar la propuesta se usará un escenario real donde se plantea la configuración de un canal seguro de transmisión mediante el protocolo SSL/TLS, aplicado a un servidor de aplicaciones. En dicho escenario se analizaran dos modelos de características, se validarán diferentes configuraciones, y se diagnosticarán varias configuraciones con problemas.

Modelo Emergente Preventivo para producir software seguro

José Carlos Sancho Núñez, Andrés Caro Lindo, Pablo García Rodríguez, José Andrés Félix de Sande

La previsión del incremento de ciberataques y de su sofisticación podrían poner en jaque a sistemas e infraestructuras críticas de consumo humano. Por este motivo, se considera necesario introducir nuevos modelos emergentes que desarrollen software de forma segura por defecto. Esta contribución realiza un experimento comercial llevado a cabo en una empresa de desarrollo. Acerca de forma novedosa una comparativa de resultados entre dos escenarios de desarrollo, uno clásico cuyo enfoque de la seguridad es reactivo y otro, emergente y preventivo que aplica la seguridad por defecto durante todas las fases del ciclo de vida del software. La reducción de un 66 % de vulnerabilidades y la minimización del impacto temporal en la resolución de los fallos de seguridad encontrados, son las claves que evidencian que la propuesta presentada construye un software más seguro por defecto que el realizado utilizando modelos clásicos.

Mejora de la seguridad de esquemas de gestión de identidades federados mediante técnicas de User Behaviour Analytics

Alejandro García Martín, Marta Beltrán

Los esquemas federados para la gestión de identidades y accesos se han extendido espectacularmente en los últimos años en entornos web, cloud y móviles. Este tipo de especificación permite que un recurso, servicio o aplicación delegue en un proveedor de identidades externo los procesos de identificación, autenticación, autorización y auditoría (IAAA). A pesar de la madurez que están adquiriendo estas especificaciones todavía suponen amenazas para la seguridad de las infraestructuras en las que se incorporan. Este trabajo propone cinco estrategias diferentes para incorporar técnicas de User Behaviour Analytics (UBA) a los flujos IAAA federados, de manera que se puedan emplear tanto en prevención como en detección de incidentes de seguridad. Además, estas cinco estrategias se validan y evalúan en un caso de uso real.

Sesión V: Ataques y vulnerabilidades

Presidente de Sesión:

Jesús Esteban Díaz-Verdejo (*Universidad de Granada*)

Jueves 5, 12:30 – 14:00 h

Seguridad de redes y sistemas de información: de la Directiva 2016/1148 al Real Decreto-Ley 12/2018

Margarita Robles Carrillo

La seguridad de redes y sistemas de información ha sido objeto de regulación en la Directiva (UE) 2016/1148. Esta norma establece una serie de obligaciones que, en el caso de España, se han traducido en la adopción del Real Decreto-Ley 12/2018, de seguridad de redes y sistemas de información, y de la Estrategia Nacional de Ciberseguridad de 2019. El análisis de estas medidas y su comparación con los preceptos de la Directiva no arroja un balance completamente positivo. El incumplimiento de la misma tiene consecuencias mayores y más graves de lo que, generalmente, implica la falta de respeto de las normas internas o de otro tipo de normas internacionales.

Intelligence-Led Cyber Attack Taxonomy (C@T)

Francisco Luis de Andrés Pérez, Mildrey Carbonell Castro

Se presenta CAT (Cyber Attack Taxonomy) nuevo modelo para analizar y representar ciberataques en su fase estratégica de alto nivel que permitirá organizar las tácticas y técnicas utilizadas por los atacantes de modo estructurado. [1]. Permite la representación de ataques generados por cualquier tipo de actor o escenario, incluyendo, por ejemplo, los ataques internos originados por Insiders (persona que materializa la ciber amenaza desde el interior de las infraestructuras del objetivo) no contemplados en modelos anteriores. Asimismo, CAT podrá ser utilizado para el modelado de ejercicios de ataque; desarrollo de frameworks específicos para sectores de especial riesgo como las infraestructuras críticas con impacto sobre la población o los repositorios de datos personales entre otros. Esta taxonomía podrá ser utilizada para la representación de cualquier tipo de ciberataque, tanto los presentes como los futuros, permitiendo el modelado desde los más simples a los ataques dirigidos, pasando incluso por los desarrollados para comprometer entornos industriales o internet de las cosas (IoT). Del mismo modo, podrá ser utilizado para definir las infraestructuras de defensa ante ciberataques mediante el análisis de contramedidas organizadas para cada fase de la estrategia CAT, incluso contra tácticas o técnicas concretas de cara a facilitar la detección, engaño o destrucción del ataque entre otras medidas.

Sistema de Cálculo de Riesgo Dinámico en Dominios Administrativos Basado en Ontologías

Fernando Monje Real, Cristina Galván, Raúl Riesco, Víctor Villagrá

Con la creciente complejidad de las ciberamenazas, se hace necesario disponer de herramientas para conocer el contexto cambiante en tiempo real. En este documento se va a presentar una arquitectura y un prototipo diseñados para modelar el riesgo de dominios administrativos, ejemplarizándolo al caso de un país en tiempo real, concretamente el de España. Para llevar a cabo esta tarea se ha realizado un modelado de los activos y de las amenazas detectadas por diversas fuentes de información. Toda esta información se almacena como conocimiento haciendo uso de ontologías, que permita aplicar motores de razonamiento para así poder inferir nuevo conocimiento utilizable posteriormente en los siguientes razonamientos. Este modelado y razonamiento se ha enriquecido con un sistema dinámico de gestión de confianza de las distintas fuentes de información, y de capacidades de incremento de fiabilidad con la inclusión de información de inteligencia de amenazas adicional.

Mirror Saturation in Amplified Reflection DDoS

João J. C. Gondim, Robson de Oliveira Albuquerque

Over the last six years, there has been two major game changers in DDoS attacks: amplified reflection and IoT. Together, they motivated well-founded security concerns relating to IoT's offered attack surface, and how it could potentialize DDoS. In order to assess those concerns, the feasibility of IoT device abuse as reflectors was evaluated. Attacks abusing two protocols were tested showing a pattern: reflector saturates without sustaining maximum amplification rates, for very low injection rates (between 10 and 100 probe/sec). Hence, if on the one hand IoT devices, in general, would not be good reflectors, they would be good injectors. An attacker could thus use more injectors while maintaining low injection rates. This would certainly require greater coordination from the attacker but tends to hamper detection. It is expected higher sophistication in DDoS attack execution, as in carpet bombing and pulse attacks, with evolution of C2 incorporating orchestration and attack coordination.

SVCP4C: A tool to collect vulnerable source code from open-source repositories linked to SonarCloud

Razvan Raducu, Gonzalo Esteban, Francisco Javier Rodríguez Lera, Camino Fernández

There is a significant body of work to detect Buffer Overflow in the literature. From manual audition of the code to dynamic analyzers, many techniques have been proposed to find vulnerabilities. Particularly one of these techniques is Machine Learning, which has gained ground in this research field lately. By teaching a Machine Learning algorithm what a vulnerability looks like, the result can predict security threats without requiring human intervention. However, fulfilling such task requires the establishment of a proper dataset. The purpose of this paper is to present SVCP4C (SonarCloud Vulnerable Code Prospector for C), a bot written in Python for collecting vulnerable source code. The bot extracts files from open-source repositories linked to SonarCloud—an online tool that performs static analysis—, and tags in such all the lines that are vulnerable. This set of tagged files may later be used to extract features and to create training datasets for Machine Learning algorithms.

Cybersecurity on Brain-Computer Interfaces: attacks and countermeasures

Sergio López Bernal, Alberto Huertas Celdrán, Gregorio Martínez Pérez

In recent years, Brain-Computer Interfaces (BCI) have increased their presence in the medical field as well as in other sectors of the industry such as entertaining or authentication. This expansion has improved not only the subjects quality of life but also their quality of experience when using entertainment systems. Despite the benefits, new paradigms such as Brain-to-Internet or Brain-to-Brain, together with novel technologies and techniques missing security and privacy by design principles are influencing the emergence of cybersecurity challenges affecting subjects' safety and data privacy. In this context, this line of work aims to review the attacks on BCI disrupting physical safety, data availability, confidentiality and integrity, as well as propose proactive and reactive countermeasures to enable their protection.

Sesión VI: Análisis forense

Presidente de Sesión:

Andrés Caro Lindo (*Universidad de Extremadura*)

Viernes 7, 10:00 – 11:30 h

Algoritmo de Interpolación Cromática para la Detección de Zonas Manipuladas de Imágenes Digitales

Esteban Alejandro Armas Vega, Luis Alberto Martínez Hernández, Sandra Pérez Arteaga, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

---

Aunque históricamente ha habido confianza en la integridad de las imágenes, el avance de la tecnología ha comenzado a erosionar esta confianza. Este documento propone un metodo de autenticación de imagen digital basado en el error cuadratico medio del patrón de interpolación CFA estimado a partir de la imagen analizada. Los resultados de los experimentos demuestran la eficiencia del metodo propuesto. Cada uno de estos experimentos se ejecutó utilizando diferentes conjuntos de datos públicos desarrollados con fines de investigación.

Forensic Analysis Overview in the IoT Environment. A Windows 10 IoT Core Approach

Juan Manuel Castelo Gómez, José Luis Martínez Martínez

The huge development of the Internet of Things (IoT) and the sudden incursion of this network into our everyday world have drastically changed the application of technology in our lives. One of the main concerns arising from the IoT is security; the way the devices have been conceived has turned out to include a massive underestimation of the security requirements, which has led to a large-scale problem. In this article, a review of the state of security on the IoT is carried out, focusing on the forensics aspect. In addition, a case study is presented on how to perform a forensic analysis in an IoT-based operating system, namely Windows 10 IoT Core.

Análisis de la Estructura de los Contenedores Multimedia de Vídeos de Dispositivos Móviles

Carlos Quinto Huamán, Daniel Povedano Álvarez, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

---

En la actualidad, los dispositivos móviles se han convertido en el sustituto natural de la cámara digital, ya que capturan situaciones cotidianas de forma fácil y rápida, promoviendo que los usuarios se expresen a través de imágenes y vídeos. Estos vídeos pueden ser compartidos a través de diferentes plataformas quedando expuestos a cualquier tipo de manipulación, lo que compromete su autenticidad e integridad. Es común que los fabricantes no cumplan al 100 % con las especificaciones del estándar, dejando características intrínsecas del dispositivo que

generó el vídeo. Las investigaciones de los últimos años se centran en el análisis de contenedores AVI, siendo muy limitado, la literatura en el caso de contenedores MP4, MOV y 3GP. En este trabajo se realiza una técnica de análisis de la estructura de los contenedores de vídeos generados por dispositivos móviles y su comportamiento al ser compartido por las redes sociales ó manipulados por programas de edición. Como resultado del análisis se tienen los siguientes resultados: verificación de la integridad de los vídeos, identificación de la fuente de adquisición y diferenciación entre vídeos originales y manipulados.

Improving Speed-Accuracy Trade-off in Face Detectors for Forensic Tools by Image Resizing

Deisy Chaves, Eduardo Fidalgo Fernández, Enrique Alegre, Pablo Blanco

During forensic material analysis, accurate and fast face detection is required prior to facial recognition of fugitives or children in sexual abuse scenes. However, this is not easy due to common limitations in the image quality or face pose. Moreover, real-time performance is expected in some applications as the forensic ones. There are several methods to address the face detection problem, but most of them are not suitable for real-time applications due to its computational complexity. In this work, we propose a strategy based on image resizing especially valid for Child Sexual Abuse crimes, oriented to improve the trade-off between the speed and the performance of three deep-learning-based face detectors. The results showed that the proposed approach is able to speed up face detection with a small reduction in accuracy. The best speed-accuracy trade-off is achieved using images resized to 50 % of the original image size.

Localización de Manipulaciones en Imágenes Analizando Artefactos de Interpolación

Edgar González Fernández, Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

Diversos trabajos han abordado el problema de detección de manipulaciones en imágenes adquiridas desde dispositivos que emplean matrices de filtro de color, comunes en el mercado debido a los bajos costes de producción. Estos dispositivos emplean algoritmos de interpolación cromática durante el proceso de formación de la imagen, lo que permite realizar un análisis estadístico en los elementos generados a partir de este proceso. La mayoría de los trabajos centran sus esfuerzos en analizar la banda verde del filtro de Bayer, ya que contiene más información. La falta de métodos para analizar eficazmente las demás bandas o distintos filtros de color reduce la capacidad de detección de las herramientas conocidas. La finalidad principal de este trabajo es proveer una metodología general para la detección de manipulaciones en este tipo de dispositivos, además de proporcionar nuevas técnicas que permiten generalizar el análisis en una gran diversidad de sensores.

Sesión VII: Cifrado

Presidente de Sesión:

Rafael Estepa Alonso (*Universidad de Sevilla*)

Miércoles 5, 12:15 – 14:00 h

Herramienta Automática de Adquisición de Información de Ransomware

Antonio López Vivar, Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

---

Los ataques de ransomware reportados a las autoridades enfrentan la dificultad técnica de las dependencias de policía local para recopilar la información y ejecutar un análisis forense adecuado. En este trabajo se propone una herramienta de análisis forense que permite adquirir suficiente información para facilitar la posterior clasificación del ransomware. La herramienta propuesta combina la captura de ventana emergente que muestra el ransomware y a través de técnicas de reconocimiento óptico de caracteres, la obtención el mensaje de rescate junto con la dirección de pago y el valor. Además, extrae los ficheros generados por el ransomware y realiza un volcado de la memoria virtual del sistema para el análisis por parte del técnico forense. Para evaluar la precisión de la herramienta, se realizaron experimentos con distintas muestras de ransomware en un ordenador real, bajo un entorno controlado.

Guidelines Towards Secure SSL Pinning in Mobile Applications

Francisco José Ramírez López, Angel Jesus Varela Vaca, Jorge Ropero, Alejandro Carrasco

---

Security is a major concern in web applications for so long, but it is only recently that the use of mobile applications has reached the level of web services. This way, we are taking OWASP Top 10 Mobile as our starting point to secure mobile applications. Insecure communication is one of the most important topics to be considered. In fact, many mobile applications do not even implement SSL/TLS validations or may have SSL/TLS vulnerabilities. This paper explains how an application can be fortified using secure SSL pinning, and offers a three-step process as an improvement of OWASP Mobile recommendations to avoid SSL pinning bypassing. Therefore, following the process described in this paper, mobile application developers may establish a secure SSL/TLS communication.

A Review of Key Enumeration Algorithms for Cold Boot Attacks

Ricardo Villanueva Polanco

In this paper, we study the cold boot attack setting. In this setting, the attacker with physical access to a machine may recover cryptographic key information of a cryptographic scheme via this data remanence attack. We first

describe the attack setting and then pose the problem of key recovery in a general way and establish a connection between the key recovery problem and the key enumeration problem. The latter problem arises in the side-channel attack literature, where, for example, the attacker might procure scoring information for each byte of an AES key from a side-channel attack and then want to efficiently enumerate and test a large number of complete 16-byte candidates until the correct key is found. Therefore, we study several algorithms to solve the key enumeration problem, such as the optimal key enumeration algorithm (OKEA) and several other non-optimal key enumeration algorithms. Additionally, we proposed variants of some of them and make a comparison of all of them, highlighting their strengths and weaknesses.

Protocolos de clave pública en anillos de grupo torcidos

María Dolores Gómez Olvera, Juan Antonio López Ramos, Blas Torrecillas Jover

La Criptografía es la ciencia que estudia la seguridad en las comunicaciones. Actualmente, los protocolos que protegen nuestra privacidad se encuentran en un proceso de renovacion, y se están proponiendo nuevos algoritmos que puedan preservar nuestra seguridad, ante la aparición de nuevas amenazas. En el ámbito del álgebra no conmutativa se está investigando en este sentido, y en esta línea proponemos un intercambio de clave en un anillo de grupo torcido mediante un cociclo, con la intención de probar que es una buena estructura en la cual basar nuestros protocolos, y posteriormente implementarlos en ella.

Comunicaciones VoIP cifradas usando Intel SGX

Raúl Ocaña, Isaac Agudo

Cada día es mas frecuente encontrar servicios en internet gestionados desde plataformas online y con la expansión de la tecnología IoT, los smartphones, las smartTV y otros tantos dispositivos: la autenticación, la distribución y al fin y al cabo, la comunicación entre extremos puede verse seriamente comprometida si dicha plataforma es atacada. La inclusión de nuevas medidas de seguridad en este tipo de ecosistemas requiere de un cambios sustancial de la arquitectura subyacente en muchos casos, por lo que su avance es lento. En este trabajo se trata de forma concreta el desarrollo de una alternativa OpenSource a uno de estos servicios, la telefonía IP (VoIP), que esta expandiéndose cada día mas, empezando por redes locales y privadas y llegando a grandes centralitas de conmutación de tele operadoras, consiguiendo así una transmisión de voz segura extremo a extremo transparente para los servidores VoIP, que no requiera modificar la infraestructura subyacente.

Poster I: Detección y monitorización

Miércoles 5, 11:30 – 12:15 h

DarkNER: A Platform for Named Entity Recognition in Tor Darknet

Muhammad Wesam Al-Nabki, Eduardo Fidalgo Fernández, Javier Velasco Mata

In this paper, we introduce DarkNER, an application of Named Entity Recognition (NER) based on neural networks to identify six categories of named entities: Location, Person, Products, Corporation, Group, and Creative-work, in onion domains on the Tor network. The presented NER model is trained on the W-NUT-2017 dataset and tested on manually tagged samples of Tor hidden services. The experiments show the adaptability and effectiveness of neural networks models in detecting new textual entities, such as drugs names and weapons brands. The proposed application could help the authorities in filtering and monitoring the contents of the Tor domains.

Evaluación de mejoras en la monitorización estadística multivariante para la detección de anomalías en tráfico ciclo-estacionario

Noemí Marta Fuentes García, José Camacho, Gabriel Maciá Fernández

El tráfico de red tiene un claro carácter ciclo-estacionario (por ejemplo, ciclos día/noche o laborables/fines de semana). Esto hace que se puedan identificar patrones de comportamiento distintos dentro de ciertos intervalos temporales: el comportamiento de la red puede variar según las horas dentro de un mismo día. Por otra parte, estos mismos patrones se repiten de forma periódica: por ejemplo, el tráfico de red es similar todas las mañanas los días laborables. Esta particularidad hace más compleja la creación de modelos de normalidad adecuados para la detección de anomalías, así como la aplicación de técnicas que capturen estas dinámicas de manera adecuada sin generar una alta tasa de falsos positivos. Nuestro trabajo actual está centrado en evaluar la aplicación de distintas alternativas de detección de anomalías dentro del enfoque de monitorización de redes estadística multivariante (MSNM). En concreto, nuestro objetivo es mejorar el área bajo la curva (AUC) y garantizar así un elevado número de verdaderos positivos a la par que se reducen los falsos positivos.

MSNM-S: An Applied Network Monitoring Tool for Anomaly Detection in Complex Network Environments

Roberto Magán Carrión, José Camacho, Gabriel Maciá Fernández, Ismael Jerez Ibáñez

Recent forecasts predict a huge number of devices inter-connected by 2021. In this scenario, new applications and services are devised mainly focused on improving people's daily life. Monitoring devices, applications and the involved information to detect security events is a great challenge in such a complex environment. Because of that, new methods, algorithms and tools must be developed. In this work, we introduce the recently released MSNM-S tool which is able to carried out the monitoring and detection of security events in this kind of scenario.

Visualización y Análisis de Tráfico Móvil para la Securización de Redes y Sistemas

José Antonio Gómez Hernández, José Camacho, Pedro García Teodoro, Gabriel Macía Fernández, Margarita Robles Carrillo, Antonio Muñoz Ropa, Juan Holgado Terriza

Dado el creciente uso de dispositivos de usuario móviles en entornos de red corporativos, junto con la también creciente existencia de vulnerabilidades en estos dispositivos, se hace precisa la adecuada protección de este tipo de entornos y sistemas. A este fin, los autores están desarrollando un proyecto de investigación fundamentado en la monitorización de dispositivos y usuarios finales para provisionar un control de acceso dinámico que reduzca los riesgos del entorno. El diseño y despliegue de la propuesta científico-técnica requiere un marco de experimentación que permita validar los desarrollos realizados. El presente trabajo recoge una base de datos de tráfico móvil adquirida en el entorno de red de la Universidad de Granada, sobre la cual se lleva a cabo un análisis preliminar del comportamiento de este tipo de usuarios como paso previo para alcanzar los objetivos del proyecto. Para una mejor comprensión, dicho análisis se apoya en la herramienta de visualización Gephi.

Análisis de las Técnicas de Detección Automática de Pornografía en Vídeos

Jenny Alexandra Cifuentes, Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

El análisis forense digital ha surgido como una disciplina para enfrentar diversos tipos de delitos informáticos y cibernéticos. En particular, teniendo en cuenta el aumento del contenido pornográfico no restringido en Internet y la difusión de casos de distribución de material de abuso sexual infantil, hay una creciente necesidad de herramientas informáticas eficientes para la detección y/o el bloqueo automático de contenido pornográfico. El objetivo de este estudio es revisar las diferentes estrategias disponibles en la literatura para la detección de pornografía, específicamente en vídeos, e identificar brechas de investigación. Este trabajo muestra que las técnicas basadas en aprendizaje profundo detectan vídeos pornográficos con mayor precisión que otras estrategias de detección convencionales. La precisión de las estrategias reportadas en este trabajo depende de las técnicas de extracción de características, la arquitectura y los algoritmos finales de clasificación. Finalmente, se detallan algunas areas complementarias de investigación en la detección de vídeos pornográficos.

Aplicación de técnicas de transfer learning a problemas de ciberseguridad

David Escudero García, Ángel Luis Muñoz Castañeda

Cada vez es más común encontrarse con diferentes aplicaciones de machine learning a diferentes problemas, incluido el ámbito de la ciberseguridad, debido a que permite automatizar procesos importantes como determinar la maliciosidad de aplicaciones software, la detección de ataques de red, etc. No obstante, para construir una solución eficaz es necesario disponer de una cantidad abundante de datos ya clasificados (etiquetados), lo cual puede ser costoso en cuanto a tiempo y recursos. El uso de técnicas de transfer learning, que permite reutilizar el conocimiento derivado de un conjunto de datos para mejorar un modelo de machine learning para un un problema con un conjunto de datos distinto, puede ser una buena solución para paliar el problema de la escasez de datos. En este trabajo se pretende realizar una evaluación del rendimiento predictivo y computacional de técnicas de transfer learning en problemas de ciberseguridad para determinar su eficacia.

Poster II: Investigación ya publicada I

A Review of Anomaly-based Exploratory Analysis and Detection of Exploits in Android

Guillermo Suárez-Tangil, Santanu Kumar Dash, Pedro García-Teodoro, José Camacho, Lorenzo Cavallaro

Smartphone platforms are becoming increasingly complex, which gives way to software vulnerabilities that are difficult to identify. In this work we present CoME, an anomaly-based methodology aiming at detecting software exploitation in Android systems. CoME models the normal behavior of a given software component or service and it is capable of identifying any unanticipated behavior. To this end, we first monitor the normal operation of a given exploitable component through lightweight virtual introspection. Then, we use a multivariate analysis approach to estimate the normality model and detect anomalies. We evaluate our system against one of the most critical vulnerable and widely exploited services in Android, i.e., the mediaserver. Results show that our approach can not only provide a meaningful explanatory of discriminant features for illegitimate activities, but it can also be used to accurately detect malicious software exploitations at runtime.

Un resumen de "Aplicación de técnicas de compresión de información a la identificación de anomalías en fuentes de datos heterogéneas: análisis y limitaciones"

Gonzalo de La Torre Abaitua, Luis Lago Fernández, David Arroyo

La interconexión y heterogeneidad de los diferentes sistemas de información de la actualidad hacen que la ciberseguridad haya evolucionado desde la clásica clasificación basada en logs y listas, hacia enfoques de carácter integral que consideran otros factores como las redes sociales, foros de discusión o mensajes de correo. Esto hace necesario disponer de un mecanismo que pueda analizar de forma agnóstica esta amplia variedad de registros de actividad y de eventos de seguridad. Partiendo de la base de que todos estos registros contienen información textual, hemos explorado el uso de la distancia de compresión normalizada (NCD) para establecer una metodología capaz de trabajar con fuentes heterogéneas de información. En este sentido, hemos partido de una contribución propia en el campo de la detección de anomalías en HTTP y la hemos extendido a la detección de dominios generados mediante DGAs (Domain Generation Algorithms) y de spam en SMS. Los diversos experimentos confirman que la metodología tiene un rendimiento aceptable de acuerdo con el estado del arte. En este punto, cabe subrayar la ventaja de nuestra propuesta en términos de simplicidad y de capacidad de ser aplicada de modo general, al margen del formato de codificación de los datos. Asimismo, también se ha observado que se alcanzan resultados positivos utilizando menos datos de entrenamiento que los usados en otras aproximaciones a los tres problemas considerados.

A Review of "What did Really Change in the new App Release?"

Paolo Calciati, Konstantin Kuznetsov, Xue Bai, Alessandra Gorla

As the mobile app market is evolving at a very fast pace, users and market managers might have a hard time understanding what really changed in a new release and whether updating the app is recommendable or could pose a security and privacy threat. We propose a ready-to-use framework to analyze the evolution of Android apps. Our

framework extracts and visualizes various information —such as how an app uses sensitive data, which third-party libraries it relies on, etc.— and combines it to create a comprehensive report on how apps evolve. We perform an empirical study on 235 applications using our framework. Our analysis reveals that Android apps tend to have more leaks of sensitive data over time, and that API calls tend to have the corresponding dangerous permission already granted when added to the code.

A Review of Scalable Detection of Botnets Based on DGA

Mattia Zago, Manuel Gil Pérez, Gregorio Martínez Pérez

This conference article is a review of a work previously published by the authors and contains a summary of the identified challenges and proposed solution. The research navigates the state-of-the-art concerning the Machine Learning (ML) approaches to the detection of botnets based on Domain Generation Algorithms (DGAs), with a critical review of the existing frameworks in terms of algorithms and features used. The research aims to polish the data exploration process with a comparative analysis of those ML features presented in the state-of-the-art. The main results are several, starting with the creation of a common ground that enables future researches to focus on the study and design of new detection solutions instead of centre on the feature discovery process; following with the experimental demonstration that detection frameworks can be effective without harming user privacy; and, finally, by presenting multiple research challenges that can be exploited by future researches.

A Review of Improving the Security and QoE in Mobile Devices through an Intelligent and Adaptive Continuous Authentication System

José María Jorquera Valero, Pedro Miguel Sánchez Sánchez, Lorenzo Fernández Maimó, Alberto Huertas Celdrán, Marcos Arjona Fernández, Gregorio Martínez Pérez

Continuous authentication systems for mobile devices focus on identifying users according to their behaviour patterns when they interact with mobile devices. Despite the benefits of these systems, they also have open challenges such as the authentication accuracy and the adaptability to new users' behaviours. With the goal of improving these challenges, the main contribution of this paper is an intelligent and adaptive continuous authentication system for mobile devices. The proposed system enables the real-time users' authentication by considering statistical information from applications, sensors and Machine Learning techniques based on anomaly detection. Several experiments demonstrated the accuracy, adaptability, resilience, and resources consumption of our solution.

Poster III: Prevención y políticas de seguridad

Técnica de Autenticación de Imágenes Digitales Basada en la Extracción de Características

Esteban Alejandro Armas Vega, Carlos Quinto Huamán, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

---

En los últimos años, ha habido un gran crecimiento en el uso de imágenes digitales en la sociedad moderna. Esto, junto con la facilidad del uso de aplicaciones de edición de imágenes, compromete la autenticidad y veracidad de una imagen digital. Estas aplicaciones permiten manipular el contenido de la imagen sin dejar rastros visibles. Ademas de esto, la facilidad de distribuir informacion a través de Internet ha hecho que la sociedad acepte todo lo que ve como verdadero sin cuestionar su integridad. Este artículo propone una técnica de autenticación de imagenes digitales que combina el análisis de los patrones de textura locales con la transformada discreta Wavelet y la transformada discreta de coseno para extraer características de cada uno de los bloques de una imagen. Posteriormente, utiliza un SVM para crear un modelo que permita verificar la autenticidad de la imagen. Los experimentos se realizaron con imagenes falsificadas de bases de datos públicas y los resultados obtenidos demuestran la eficacia del método propuesto.

Guía Nacional de Notificación y Gestión de Ciberincidentes, Ventana Única e Indicadores

David Carlos Sánchez Cabello, Alberto Sánchez Del Monte, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

---

La Guía Nacional de Notificación y Gestión de Ciberincidentes, aprobada por el Consejo Nacional de Ciberseguridad en enero de 2019, se ha elaborado bajo un espíritu eminentemente práctico por parte de aquellos organismos de la Administración española involucrados en la ciberseguridad. Por ello, ha sido necesario aportar toda aquella experiencia adquirida en la gestión de ciberincidentes a lo largo de los últimos años por CSIRT y autoridades competentes en la materia. El documento se ha diseñado para conformar una metodología de trabajo compartida y establecer unos indicadores y parámetros unificados, tanto para Operadores de Servicios Esenciales (OSE) y Proveedores de Servicios Digitales (PSD) con obligatoriedad de notificación de acuerdo al Real Decreto-ley 12/2018, como para ciudadanos, Administraciones públicas y empresas privadas. Así pues, este Guía define, entre otros, conceptos como el nivel de peligrosidad e impacto asociado a un incidente, una ventanilla única de notificación y una taxonomía homogénea.

El Efecto de la Transposición de la Directiva NIS en el Sector Estratégico TIC de la ley 8/2011

David Carlos Sánchez Cabello, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

Las redes y sistemas de información desempeñan un papel crucial en la sociedad. Su fiabilidad y seguridad son esenciales para las actividades económicas y sociales, y en concreto para el correcto funcionamiento de los mercados tanto nacionales como internaciones. Debido a ese carácter transnacional, un incidente acaecido en esos sistemas informáticos ya sea o no deliberado, y con independencia del lugar en que se produzca, puede afectar a diferentes Estados miembros y a la Unión en su conjunto a través de su mercado interior. Esto permite entender que, la seguridad de las redes y sistemas de información es fundamental para el correcto funcionamiento del mercado. Tanto la directiva NIS como su transposición tienen por objetivo proteger todos estos sectores definiéndolos concretamente en 6. Toda esta nueva regulación ha de encajar con la regulación ya existente como la ley 8/2011 por la que se establecen medidas para la protección de las infraestructuras críticas y el Plan Estratégico Sectorial de las Tecnologías de la Información y la Comunicación (TIC).

CyberHeroes: Aplicación móvil para fomentar el buen uso de la tecnología e Internet en menores

Mario González, Gregorio López, Víctor Villagrá

El uso de la tecnología y el acceso a Internet de los menores es un asunto de capital importancia para la sociedad de hoy en día, que atrae cada vez más atención. Este artículo presenta un prototipo de juego de preguntas y respuestas desarrollado para iOS cuyo objetivo es precisamente concienciar y fomentar el buen uso de Internet y de la tecnología en menores. Asimismo, el artículo esboza futuras oportunidades y líneas de investigación a las que puede dar lugar esta primera iniciativa.

A Generic Solution for Authenticated Group Key Establishment From Key Encapsulation – a Compiler for Post-Quantum Primitives

Edoardo Persichetti, Rainer Steinwandt, Adriana Suárez Corona

We present a generic group key exchange construction that builds on a key encapsulation mechanism and a signature scheme. When applied to existing post-quantum proposals, the compiler provides a three-round solution for authenticated group key establishment that is quantum resistant and requires only one signature per user.

Seguridad y Privacidad en el Internet de las Cosas

Alejandra Guadalupe Silva Trujillo, Jesús Gerardo Heredia Guerrero, Pedro David Arjona Villicaña, Ana Paola Juárez Jalomo, Ana Lucila Sandoval Orozco

Las tendencias de la Industria 4.0 están fuertemente relacionadas con el Internet de las cosas (IoT). Varias áreas, como la industrial, biomédica, educativa y de entretenimiento, exigen cada vez más el uso de sistemas integrados para ofrecer una mejor experiencia de usuario a través de la conectividad y el uso efectivo de las tecnologías. Estos

dispositivos generan, procesan e intercambian una gran cantidad de información, de la cual gran parte se considera información confidencial. Los ataques a los dispositivos IoT son críticos porque pueden ocasionar daños físicos e incluso amenazar la vida de un ser humano. El objetivo de este documento es proporcionar una introducción al funcionamiento de los sistemas IoT, para conocer las opciones que se han desarrollado para la protección de información sensible y los desafíos que permanecen latentes, que es donde se necesita más investigación.

Poster IV: Investigación ya publicada II

A review of Behavioral Biometric Authentication in Android Unlock Patterns through Machine Learning

José Torres, Marcos Arjona, Sergio de los Santos, Efthimios Alepis, Constantinos Patsakis

---

Due to the ever-increasing deployment of services for which users need to authenticate, many applications require higher standards of security, such as drawn patterns and fingerprints, used mostly to authenticate users and unlock their smart devices. In this work we propose a biometrics-based machine learning approach that supports user authentication in Android to augment native user authentication mechanisms, making the process more seamless and secure. Our evaluation results show very high rates of success, both for authenticating the legitimate user and for rejecting an adversary who imitates the legitimate user. Finally, we showcase how the proposed solution can be securely deployed in non-rooted Android devices.

Formal verification of the YubiKey and YubiHSM APIs in Maude-NPA

Antonio González Burgueño, Damián Aparicio-Sánchez, Santiago Escobar, Catherine Meadows, José Meseguer

---

We have performed in [1] an automated analysis of two devices developed by Yubico: YubiKey, designed to authenticate a user to network-based services, and YubiHSM, Yubico's hardware security module. Both are analyzed using the Maude-NPA cryptographic protocol analyzer. YubiKey & YubiHSM are cryptographic Application Programmer Interfaces, involving a number of complex features: (i) discrete time in the form of Lamport clocks, (ii) a mutable memory for storing previously seen keys or nonces, (iii) event-based properties that require an analysis of sequences of actions, and (iv) reasoning modulo exclusive-or. In [1], we were able to automatically prove security properties of YubiKey and find the known attacks on the YubiHSM.

A review of Message Anonymity on Predictable Opportunistic Networks

Depeng Chen, Guillermo Navarro-Arribas, Cristina Pérez-Solà, Joan Borrell

---

We review the use of simple onion routing for message anonymity in deterministic opportunistic networks. We provide stochastic onion routing algorithms and anonymity measures for such scenarios.

A Review of "Characteristics and Detectability of Windows Auto-Start Extensibility Points in Memory Forensics"

Daniel Uroz, Ricardo J. Rodríguez

Memory forensics consists in dumping the memory of a computer to a file and analyzing it with the appropriate tools. Many security incidents are caused by malware that targets and persists as long as possible in a Windows

system within an organization. The persistence is achieved using Auto-Start Extensibility Points (ASEPs), the subset of OS and application extensibility points that allow a program to auto-start without any explicit user invocation. In this paper, we propose a taxonomy of the Windows ASEPs, considering the features that are used or abused by malware to achieve persistence. Many of these ASEPs rely on the Windows Registry. We also introduce the tool Winesap, a Volatility plugin that analyzes the registry based Windows ASEPs in a memory dump.

Design recommendations for online cybersecurity courses

Lorena González Manzano, José María de Fuentes

Nowadays, a significant amount of free online cybersecurity training courses are offered. When preparing further courses, the designer has to decide what to teach and how to do it. In this paper, we provide with a set of recommendations for both issues. Concerning topic selection, 35 free online courses are analysed using NIST's NICE reference framework. Thus, several training gaps are discovered. Concerning the way of preparing the course (or refining it after the first edition), a set of good practices is proposed based on students' performance and commitment in a cybersecurity MOOC with +2,000 initially active students. To foster further research in this area, an open-source framework is released to enable the analysis of students' performance in EdX MOOCs.

Poster V: Ataques y vulnerabilidades / Análisis Forense

Mitigación de amenazas a la privacidad en OpenID Connect mediante la introducción de un Privacy

Jorge Navas, Marta Beltrán

Las soluciones de Gestión Federada de Identidad están siendo adoptadas ampliamente en entornos móviles, web y cloud en los últimos años, y lo serán en el futuro en entornos como Internet of Things o Edge Computing. Empresas como Google, Facebook, Amazon, Linkedin, Microsoft o Salesforce, por mencionar algún ejemplo significativo, han apoyado la creación de estándares como OAuth u OpenID Connect convirtiéndose en muchos casos en proveedores de identidad. De esta manera resuelven los problemas de Identificación, Autenticación, Autorización y/o Auditoria (IAAA) de los usuarios finales en un sólo flujo. Sin embargo, las especificaciones de OpenID Connect no se encuentran exentas de amenazas en cuanto a privacidad: el proveedor de identidades almacena información sobre los usuarios y sus atributos y registra las peticiones de acceso que van realizando con sus identidades. Este trabajo en desarrollo propone la introducción de un nuevo agente en los flujos, el árbitro de privacidad, que mitigue o evite estas amenazas.

Boosting child abuse victim identification in Forensic Tools with hashing techniques

Rubel Biswas, Victor González-Castro, Eduardo Fidalgo Fernández, Deisy Chaves

---

In this work, we present a scheme to identify occluded faces using perceptual image hashing. Most of the existing methods for this problem focus on occlusion detection and further removal of the occluded area by training a facial model. In this paper, we propose a new hashing method which does not require prior training. Our model combines frequency coefficients and statistical image information to increase the recognition accuracy of occluded faces. The proposed method aims to improve face recognition accuracy in forensic tools such as victim identification in Child Sexual Abuse (CSA) materials. Experimental results showed that the proposed method outperforms the results obtained with perceptual image hashing for occluded face identification using the LFW database.

¿Cómo representar un Buffer Overflow? Una revisión literaria sobre sus características

Gonzalo Esteban, Razvan Raducu, Ángel Manuel Guerrero Higueras, Camino Fernández

---

Detectar vulnerabilidades como los Buffer Overflows generalmente implica el uso de herramientas de análisis de código fuente o de revisiones manuales por parte de expertos. En este campo, la inclusion de técnicas como el aprendizaje automático pretende mejorar dicho proceso abriendo la puerta a la predicción de vulnerabilidades. A grandes rasgos, tanto las herramientas de análisis de código como las técnicas de aprendizaje automático funcionan representando ciertas características del código fuente. Sin embargo, en la literatura académica no se ha abordado la cuestión de cuántas formas de representar un Buffer Overflow existen. Con el objetivo de cubrir tal carencia, este trabajo presenta una revision sistemática de la literatura. Fruto de ello ha sido la identificación

de 79 características, todas ellas recogidas en 8 representaciones distintas. Estos resultados podrían asentar las bases para futuras investigaciones en el campo de la predicción de Buffer Overflows. En concreto, tanto para crear nuevos conjuntos de características a partir de las ya identificadas en la literatura, como para evaluar o comparar si los conjuntos encontrados son efectivos a la hora de predecir y representar este tipo de vulnerabilidades.

Extended Abstract: Are You Sure They Are the Same? Identifying Differences Between iOS and Android Implementations

Daniel Domínguez Álvarez, Alessandra Gorla, Juan Caballero, Roberto Giacobazzi

Most mobile applications are available for multiple platforms, most often Android and iOS since they jointly cover nearly the entire market. While the functionality of the Android and iOS implementations of an application may be expected to be the same, in reality they may differ significantly due to misalignments during the application development process. This extended abstract presents an ongoing project whose goal is to identify the differences, in terms of functionality and security offered to the user, of the Android and iOS implementations of a mobile application. Our current approach focuses on differences in the network traffic. Our preliminary results show that some security functionality may be implemented in only one of the two platforms. In a extreme case, one application encrypts its network traffic in Android, but not in iOS. Other applications only implement TLS pinning on Android and may only check it in some parts of the application.

Ciberseguridad en entornos de generación eléctrica en parques renovables. Resumen extendido

Antonio Estepa Alonso, Jesús Díaz Verdejo, Estefanía de Osma Ramírez, Rafael Estepa Alonso, Germán Madinabeitia Luque, Agustín Lara Romero

Este documento presenta un proyecto en curso en el marco de ciberseguridad en entornos industriales de generación eléctrica. Por limitaciones de espacio y por motivos de confidencialidad, tan sólo se describirá el contexto de este proyecto, el alcance esperado y los requisitos que debe cumplir la solución de ciberseguridad. Por último se realiza una breve introducción al diseño inicial de la solución propuesta siguiendo la aproximación de Mínimo Producto Viable. Dicha solución se basa en la definición de Indicadores de Compromiso IoC para la detección anomalías y vulnerabilidades en la planta.

Proceso para la implementación de un ecosistema Big Data seguro

Julio Moreno, Manuel A. Serrano, Eduardo Fernández-Medina, Eduardo B. Fernandez

Un entorno Big Data es un potente y complejo ecosistema que ayuda a las compañías a extraer información relevante de los datos, la cual, puede ser utilizada para ayudar en la toma de decisiones y en el desempeño del negocio. En este contexto, y debido a la cantidad, la variedad y la sensibilidad de los datos gestionados por este tipo de sistemas, la privacidad y seguridad se vuelven cruciales. Sin embargo, asegurar un ecosistema Big Data no es trivial y no debe ser tratado desde una perspectiva parcial o aislada, sino que se debe adoptar un enfoque holístico que comience desde el momento en el que se definen los requisitos y políticas, y continúe hasta que se desarrolle e implemente el ecosistema. Por todo ello, en este artículo, presentamos una visión metodológica para resolver este

problema mediante la integración de la seguridad y privacidad en el proceso de implementación de un ecosistema Big Data. Nuestra propuesta se basa en los principales estándares y buenas prácticas internacionales. Además, nuestro proceso se encuentra soportado por una Arquitectura de Referencia de Seguridad para Big Data, la cual, establece los principales componentes de este tipo de tecnologías mientras incorpora conceptos de seguridad.

Vulnerabilidades en altavoces inteligentes

Raúl Marvan Medina, Alejandra Guadalupe Silva Trujillo, Luis Carlos Bacasehua Morales, Claudio Isauro Nava Torres, Ana Lucila Sandoval Orozco

El uso de dispositivos en el hogar conectados a Internet alrededor del mundo ha aumentado considerablemente en los ultimos años. Los hogares inteligentes tienen cada vez más dispositivos conectados. El IoT (Internet of Things) ha cambiado la forma de interactuar con nuestros dispositivos y por eso mismo, ha aumentado la preocupación sobre el trato que se le da a la información recopilada por parte de las empresas fabricantes. El objetivo de este trabajo es compendiar la evidencia de vulnerabilidades en la seguridad y privacidad de altavoces inteligentes, que pueden poner en riesgo la confidencialidad de los usuarios. Y derivado de ello, identificar los retos donde se requiere de mayor investigación.

Visión General de las Técnicas de Identificación de la Fuente de Vídeos Digitales

Raquel Ramos López, Elena Almaraz Luengo, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

Los dispositivos móviles han tenido un impacto enorme en nuestra sociedad, no sólo sirven para que las personas se comuniquen sino que gracias a la diversidad de aplicaciones que contienen, hacen que sea una práctica habitual la creación de vídeos digitales con este tipo de dispositivos dada su facilidad de grabación y distribución. Sin embargo, estos vídeos pueden mostrar en ocasiones actos ilegales por lo que el análisis forense de dispositivos móviles adquiere una trascendental relevancia para deslindar responsabilidades o como parte de la evidencia en procesos judiciales siendo el propósito de esta rama del análisis forense relacionar a un individuo con un dispositivo. En este documento se presenta un resumen de las principales contribuciones en el mundo de la identificación de la fuente de vídeos digitales de dispositivos móviles que utilizan el patrón de ruido PRNU, y como afecta la compresión de un vídeo al patrón PRNU.

Premios RENIC

Seguridad en Dispositivos Médicos Implantables

Carmen Cámara

La bioingeniería es un campo en expansión. Las nuevas tecnologías parecen proporcionar un tratamiento más eficaz de las enfermedades o de las deficiencias humanas. Los Dispositivos Médicos Implantables (DMIs) constituyen un ejemplo. Estos dispositivos poseen actualmente más capacidad de computación, toma de decisiones y comunicación. Varios trabajos de investigación en el campo de la seguridad informática han identificado serios riesgos de seguridad y privacidad en los DMIs que podrían comprometer el implante e incluso la salud del paciente que lo porta. La tesis examina los principales objetivos de seguridad para la próxima generación de DMIs, analiza los mecanismos de protección más relevantes propuestos hasta ahora, y plantea soluciones de seguridad, principalmente basadas en medidas biométricas, para la mitigación de los problemas de seguridad encontrados. Las propuestas de seguridad deben tener en cuenta las limitaciones inherentes de estos pequeños dispositivos implantados: energía, almacenamiento y potencia de cálculo. Por otra parte, las soluciones propuestas deben lograr un equilibrio adecuado entre la seguridad del paciente y el nivel de seguridad ofrecido, siendo la vida útil de la batería otro parámetro crítico en la fase de diseño.

Ciberseguridad aplicada a la automoción. Smart car cibersecurity

Pablo Escapa Gordón, Héctor Alaiz Moretón

El propósito general de este resumen extendido es explicar el desarrollo, objetivos y las conclusiones del trabajo final de máster denominado: “Ciberseguridad aplicada a la automoción” consistente en: describir los sistemas y redes de computación que equipan los automóviles modernos, la búsqueda de vulnerabilidades, la proposición de posibles soluciones para paliarlas y la realización de diversas pruebas de concepto en entornos reales. Los sistemas descritos son la base del futuro coche autónomo por eso debemos trabajar en securizar e intentar minimizar o mitigar los posibles ataques a los que puedan ser sometidos.

Comunicaciones

DeepConfusables: mejorando la detección de ataques basados en codificación Unicode

Alfonso Muñoz Muñoz
Head of cybersecurity lab.
BBVA Next Tecnologies
Av. de Manoteras, 44. Madrid. España
alfonso.munoz2.next@bbva.com

José Ignacio Escribano Pablos
Security Researcher.
BBVA Next Tecnologies
Av. de Manoteras, 44. Madrid. España
joseignacio.escribano.pablos.next@bbva.com

Miguel Hernández Boza
Security Researcher.
BBVA Next Tecnologies
Av. de Manoteras, 44. Madrid. España
miguel.hernandez2.next@bbva.com

***Resumen*—En los últimos 5 años ha habido un interés creciente en el uso de la inteligencia artificial en el mundo de la ciberseguridad, especialmente en el ámbito defensivo y de detección de patrones, aunque, más sorprendente, es su intento de aplicación a seguridad ofensiva: pentesting, exploiting, etc. Nuestro trabajo se engloba en esta última tendencia revisando la utilidad del deep learning y transfer learning en la mejora de ataques basados en caracteres Unicode. Como resultado se publica el mejor diccionario de confusables existente a día de hoy, utilizable a través de nuestra herramienta *DeepConfusables*, y se revisan casos de uso, servicios importantes en Internet (Telegram, Whatsapp, Signal, Skype o Turnitin), donde las contramedidas a este tipo de ataques no están suficientemente bien resueltas, por ejemplo, phishing o productos de detección de autoría.**

***Index Terms*—Machine Learning, Deep Learning, Unicode, Confusables, Seguridad, Phishing, UX**

Tipo de contribución: *Investigación original*

I. Estructura

El presente artículo está estructurado de la siguiente manera: en la sección II se introduce el machine learning y su aplicación a ciberseguridad, poniendo énfasis en las tendencias en seguridad ofensiva. En la sección III, nos centraremos en el tratamiento de los caracteres Unicode y los *confusables*, dando una perspectiva global de lo que ha realizado en esta dirección en los últimos años. En la sección IV describiremos la investigación realizada, la aplicación de deep learning para la extracción de características de los caracteres Unicode y lograr una mejor detección de *confusables*. Finalmente, en la sección V, se presentarán distintos ejemplos en distintas plataformas y cómo se muestran los *confusables*. Al mismo tiempo se señalarán las medidas de seguridad que estas plataformas implementan. En la sección VI se resume los principales resultados de la investigación.

II. Machine learning y ciberseguridad

En los últimos años, tanto el *machine learning* (ML), y en particular, el *deep learning* (DL) se han convertido en potentes herramientas con las que obtener patrones sobre grandes volúmenes de datos y ha sido aplicado a numerosas áreas del conocimiento, entre las que se encuentran la visión por computador [1–3], la detección y predicción de enfermedades [4–7], entre otras.

Un área donde el *machine learning* ha cobrado especial relevancia es el de la seguridad de la información, ya que es posible aplicarlo tanto a técnicas defensivas como ofensivas. En cuanto a seguridad defensiva, se han aplicado a esteganografía [8–10], detección de anomalías en tráfico de red [11–13], detección y clasificación de malware [14–18] o detección de vulnerabilidades [19–22], entre otros.

En el campo de la seguridad ofensiva, se han publicado numerosos ataques contra algoritmos de *machine learning* [23–25], ya que numerosos algoritmos de machine learning se han visto vulnerados mediante pequeños cambios imperceptibles en los datos de entrada, conocidos como *ejemplos adversarios*[1]. Así como propuestas de robos de modelos de machine learning [26], troyanos o backdoors en modelos [27] o diferentes herramientas y frameworks opensource para automatizar tareas de exploiting, cracking, test o pentesting[2,3,4,5].

Aunque es cierto, por lo anterior, que existen ejemplos de aplicación del ML a la seguridad ofensiva creemos que existen todavía un recorrido importante en su uso a casos concretos en el mundo de la ciberseguridad. Con especial aplicación en el mundo ofensivo y lógicamente por su utilidad para modelar mejor a un atacante y testear la seguridad de los sistemas e idealmente proponer contramedidas que los mejoren.

Con este enfoque se aborda la utilidad del deep learning y transfer learning a un nuevo caso en ciberseguridad, los ataques basados en codificación Unicode. Nuestro trabajo se centra especialmente en la creación automática de un diccionario de confusables que permita verificar mejor cómo de fácil puede ser engañar a sistemas con la utilización de los mismos, demostrando que la inteligencia artificial tiene recorrido en el ámbito de la seguridad ofensiva.

III. Machine learning y ataques contra Unicode

Las ataques a Unicode[6] han sido recurrentes a lo largo de los últimos años, aún así, siguen estando muy presentes en la actualidad: bloqueo de *iPhones*[7] con mensajes que incluyen caracteres extraños, el caso del chat de *Playstation 4*[8] o suplantar el origen de una llamada[9], son algunos de los ejemplos representativos más recientes.

[1] https://blog.openai.com/adversarial-example-research
[2] https://github.com/tensorflow/cleverhans
[3] https://github.com/13o-bbr-bbq/machine_learning_security
[4] https://github.com/brannondorsey/PassGAN
[5] https://github.com/13o-bbr-bbq/machine_learning_security/tree/master/DeepExploit
[6] http://www.unicode.org/standard/standard.html
[7] https://techcrunch.com/2018/02/16/iphone-bug-telugu-unicode-ios-mac-text-bomb/
[8] https://www.ign.com/articles/2018/10/16/ps4s-are-reportedly-being-bricked-and-sony-is-working-on-a-fix
[9] https://appleinsider.com/articles/19/01/04/new-phishing-scam-masquerades-as-apple-support-call

Figura 1: Uso de *confusables* en dominios. Nótese que el dominio falso contiene una a cirílica en vez de una a latina.

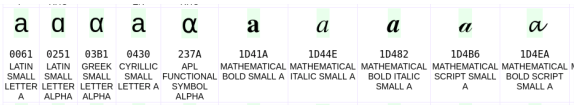


Figura 2: Algunos *confusables* para la letra a proporcionados por Unicode.

Muchos de estos ataques[10] están vinculados al mal tratamiento y validación de estos caracteres, a veces, extraños, normalizados en el estándar llamado Unicode. Estos ataques siguen funcionando y siendo viables en aplicaciones mal testeadas. Entre los tipos de ataques más comunes se encuentran: phishing con url similares, evasión de sistemas de protección de derechos de autor, fugas de información al saltarse las protecciones de *DLPs*, herramientas de esteganografía, introducción de fallos para alterar la fase de entrenamiento en algoritmos de machine learning, etc. La mayoría de los problemas anteriores se basan en el mal uso de confusables (ataques visuales basados en homoglifos), caracteres muy similares a otros que pueden ser confundidos con otros de diferente lenguaje.

Esto puede entender fácilmente con ataques de falsificación de URLs para provocar ataques dirigidos o spear phishing (Fig. 1). En los últimos años se han publicado diversas herramientas de open source con ese interés:

- Squatm3.[11]
- EvilURL.[12]
- Confusables.[13]
- homographs.[14]
- IDN Generator.[15]

En resumen, tener un buen diccionario, en calidad y cantidad, de *confusables* puede facilitar testar mejor las implementaciones de aplicaciones y servicios web ante peticiones o textos "maliciosos" (Fig. 1 y 2) minimizando ataques basados codificación Unicode.

Con este enfoque nuestras contribuciones son:

- Diseñar e implementar un sistema que permita obtener un diccionario completo de confusables de Unicode de forma semiautomática, en el paper se justificará porqué es mejor esta propuesta que una plenamente automática.
- Aplicar técnicas de deep learning para obtener confusables que no son obtenidos de forma clásica (a mano).

[10]https://capec.mitre.org/data/definitions/71.html
[11]https://github.com/david3107/squatm3
[12]https://github.com/UndeadSec/EvilURL
[13]https://github.com/wanderingstan/Confusables
[14]https://github.com/KlausBrunner/homographs
[15]https://github.com/phishai/idn_generator

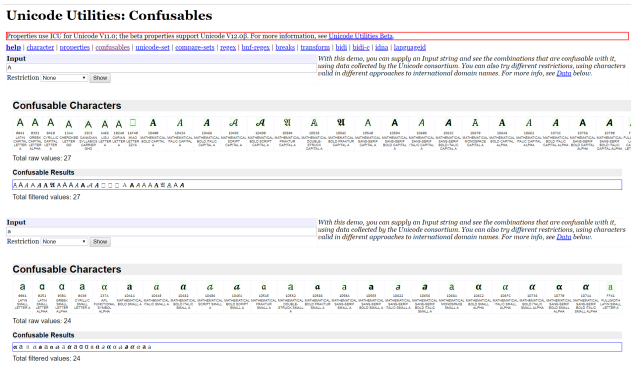


Figura 3: Confusables para los caracteres A/a provenientes del Unicode Consortium.

IV. Deep confusables: mejorando los ataques usando codificación Unicode

Desde el estándar Unicode se ha intentando facilitar un listado con los caracteres que visualmente son similares. Este listado de *confusables*[16] podría facilitar, a modo de lista negra, detectarlos con facilidad y minimizar ataques en su utilización. Esta lista se realiza mediante un proceso manual y por medio del consenso de la propia organización de Unicode (Fig. 3). Con lo que una actualización de nuevos caracteres Unicode requiere un esfuerzo importante en actualizar esta lista y, por nuestra experiencia (que se deduce el diccionario de confusables generado), no están contempladas todas las opciones visualmente similares. Es precisamente en este punto donde surje el interés de analizar el potencial del deep learning para generar un diccionario de confusables lo más completo, actualizado y automático posible.

IV-A. Preparación del dataset. Obtención de imágenes de caracteres Unicode

Nuestra propuesta para detectar como de similares son dos caracteres Unicode consiste en tratarlos como si fueran imágenes, de esta forma se pueden aplicar diversos principios para calcular la similitud. El proceso para generar un diccionario de confusables sofisticado requiere necesariamente de un repositorio de imágenes normalizadas. En nuestra investigación se toma como referencia las imágenes de la página web Unicode Table[17]. Dada esa tabla se procede a extraer carácter a carácter Unicode. Cada carácter se extrae como una imagen con dimensiones de 34x34 píxeles y en RGB. El proceso se ha hecho de forma semiautomática: para cada uno de los bloques Unicode[18], se ha tomado una captura de pantalla de la página web anterior, se han contado el número de caracteres de cada uno de los bloques y se ha ido recorriendo la cuadrícula y guardando cada uno de los caracteres. Tras un primer proceso de extracción, se han revisado de forma manual los bloques de caracteres reservados y/o vacíos para no incluirlos.

La base de datos creada cuenta con un total de 38 880 caracteres únicos, agrupados en 266 bloques. Hemos liberado esta base de datos y está disponible libremente en GitHub.[19]

[16]https://unicode.org/cldr/utility/confusables.jsp
[17]https://unicode-table.com
[18]https://unicode-table.com/en/blocks
[19]https://github.com/next-security-lab/unicode-images-database

IV-B. Generación de similaridad. Comparando caracteres Unicode

La arquitectura diseñada para extraer la similitud entre caracteres Unicode dado el dataset creado se refleja en la Fig. 4. Para cada par de imágenes se reescalan a 224x224 píxeles[20], y se obtienen las características de cada una a partir de un modelo de deep learning preentrenado. Uno de los atractivos del deep learning es ser capaz de extraer características de forma automática a partir de multitud de datos diferentes [28], [29]. Es precisamente esta idea la que se utilizará para extraer las características más relevantes de las imágenes de los caracteres y obtener posteriormente la similaridad de forma automática. Por lo tanto, las características de cada par de imágenes son pasadas a una función de similitud, que devuelve un valor en el rango de 0 y 1, lo que nos permitirá asignar probabilidad de coincidencia. Como función de similitud se ha utilizado la función del coseno de similitud definida como sigue:

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\|_2 \cdot \|y\|_2} \quad (1)$$

donde x, y son dos vectores, $\langle \cdot, \cdot \rangle$ designa el producto escalar usual y $\|\cdot\|_2$ designa la norma eucĺidea del vector.

IV-C. Tecnologías y redes neuronales utilizadas

La implementación de la arquitectura de similitud utiliza lenguaje de programación Python, el framework Scikit-learn [30] para la implementación de la similitud por función coseno y el framework de deep learning Keras [31], que permite correr otros frameworks de deep learning para otros frameworks como Tensorflow [32], CNTK [33] y Theano [34] compartiendo una misma API. En nuestro caso se ha utilizado Keras 2.2.0 y Tensorflow 1.5.0. En nuestro diseño no es necesario diseñar una red neuronal desde cero si no que es posible utilizar modelos preentrenados de Keras usando los principios de transfer learning para adaptarlo a nuestro escenario de comparación de caracteres Unicode. En concreto, Keras dispone de modelos preentrenados, entre los que se encuentran Xception [35], VGG16 y VGG19 [36], ResNet50 [2], Inception V3 [37], MobileNet [38], DenseNet [39], NasNet [40] y MobileNetV2 [41] que utilizan como base de datos ImageNet [42]. Los modelos analizados durante esta investigación han sido los siguientes:

- VGG 16 [36]
- VGG 19 [36]
- ResNet 50 [2]
- DenseNet [39]
- MobileNet [41]

Para todos los modelos anteriores se ha obtenido una matriz de similitud entre los caracteres Unicode siguiendo la estructura descrita en la Fig. 4. Para cada una de estas matrices se han obtenido los confusables de cada caracter, forzamos un mínimo del 50 % de similitud entre caracteres (Fig. 5). Se

[20]Este es el tamaño necesario para que las imágenes sean alimentadas por los modelos preentrenados de deep learning que se han utilizado en la investigación

Tabla I: Media de confusables por modelo

Modelo	Media de confusables
DenseNet	26 638
MobileNet	2 237
ResNet50	4 634
VGG16	1 046
VGG19	973

puede apreciar que el modelo que más confusables obtiene es DenseNet, seguido de ResNet50, MobileNet, VGG16 y VGG19, independientemente del grado de similitud.

El número medio de confusables por cada modelo se puede ver en la Tabla I. El modelo DenseNet no parece adecuado para la búsqueda de confusables precisos ya que obtiene un gran número de ellos (se acerca a la muestra inicial de todos los caracteres de partida, 38800). La gran mayoría de ellos son falsos positivos.

Si atendemos, además, al número de confusables por cada caracter de cada modelo (Figura 6), observamos que la inmensa mayoría de ellos tienen una gran cantidad de confusables, por lo que elegir el modelo más adecuado es fundamental. A mayor cantidad de ”potencialesçonfusables también mayor cantidad de falsos positivos pero si la detección es muy precisa en similitud podríamos omitir potenciales candidatos a confusables. Es necesario establecer un balance adecuado.

Por nuestra experimentación un buena elección consistiría en el uso del modelo VGG16 o VGG19, en un equilibrio razonable entre cantidad de confusables y calidad (razonablemente similares). En nuestra pruebas, y para los resultados posteriores, se decidió la utilización del modelo VGG16 poniendo foco en similitudes del 75, 80, 85, 90, 95 y 99 %.

IV-D. Diccionario de confusables - deepDiccConfusables

Para la creación del diccionario de confusables, se decidió utilizar el modelo VGG16 con grado de similitud 75 %. Diferentes pruebas indicaron que es un valor adecuado entre un número no excesivo de caracteres diferentes, falsos positivos, al caracter unicode objetivo (desde un punto de vista humano) y flexiblidad a la hora de detectar caracteres que puedan ser razonablemente similares (usar un grado de % de similitud alto impediría esto). Una vez generado dicho diccionario inicial, se procedió a analizarlo, compararlo y ampliarlo con el disponible, más amplio hasta el que hemos creado, publicado por Unicode [21].

Este proceso extrae alguna conclusiones interesantes que podrían ser de utilidad para otros procesos automáticos de seguridad basados en machine learning:

1. La asignación manual, como hace la organización Unicode Consortium, da resultados más precisos desde un punto de vista humano que la clasificación automática. Algo que en principio parece razonable.

2. La clasificación automática detecta mayor cantidad de posibles confusables que la asignación manual. Algunos de los cuales deberían haber sido cubiertos, por su similitud, en la asignación manual del consortium. Esto es un aspecto cualitativo, ya que depende de la percepción de un humano de si un caracter pasaría más o menos desapercibido. Nuestra

[21]https://unicode.org/cldr/utility/confusables.jsp

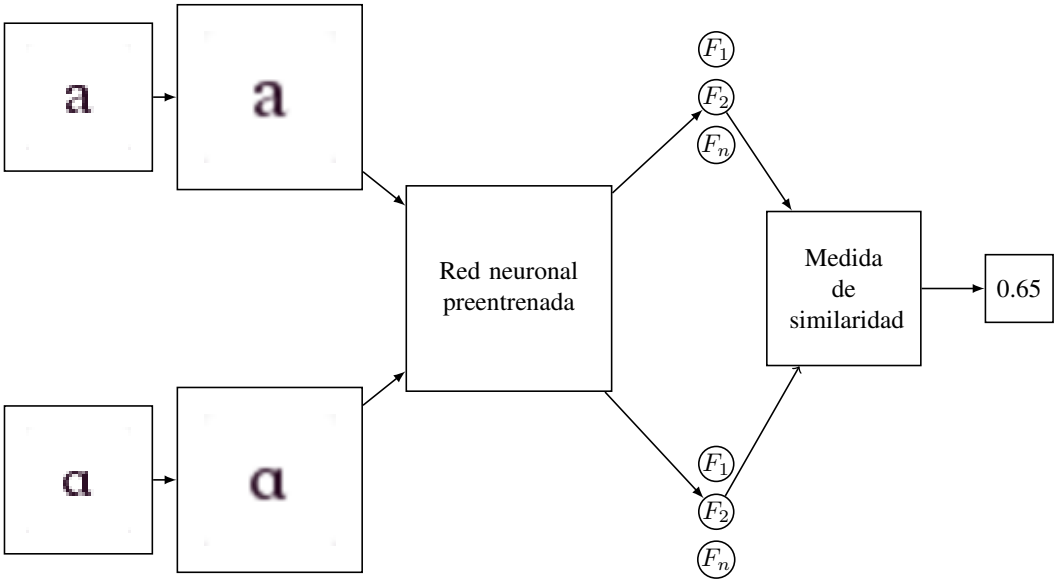


Figura 4: Arquitectura para comparación de imágenes que representa caracteres Unicode

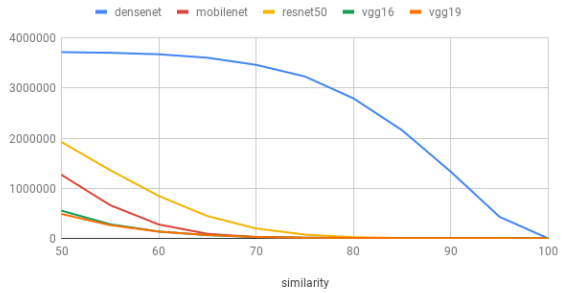


Figura 5: Ejemplo de cantidad de confusables comparando cada carácter Unicode Latin-1 con el resto de caracteres Unicode (38800). El valor del eje y es acumulado desde 50 % hasta 100 %, en pasos de 5.

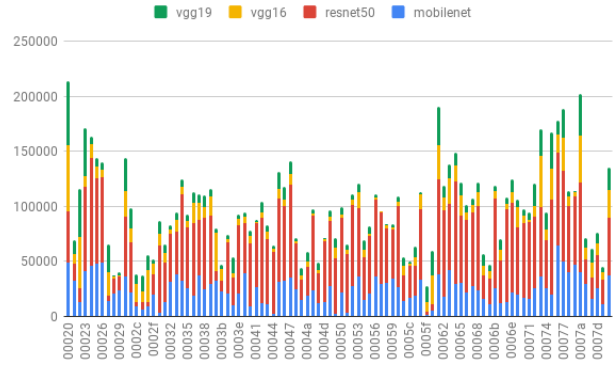


Figura 6: Confusables por carácter (no incluye DenseNet)

apreciación es que la clasicación automática complementa el diccionario de partida con un 10 % de confusables de calidad.

3. Las pruebas muestran que la clasificación automática permite detectar todos los confusables existentes en el listado del unicode consortium y detecta de media un 50 % de confusables adicionales por cada caracter analizado. La diferencia

Figura 7: Diccionario final de confusables creado por los autores

fundalmente con respecto a la asignación manual es que para generar ese resultado se introducen caracteres menos similares que podrían ser filtrados posteriormente a mano.

Finalmente, el diccionario creado dispone de media 54 confusables razonables por cada caracter. El diccionario completo puede descargarse desde GitHub[22].

IV-E. Herramienta Deep Confusables

Unido al desarrollo del diccionario se decidió implementar la herramienta `Deep Confusables` para que, adicionalmente a otros casos de análisis utilizando el diccionario, diferentes investigadores puedan verificar de mejor manera la seguridad y contramedidas de la aplicación de caracteres Unicode a urls. La herramienta está disponible en GitHub[23], siendo útil tanto para tareas ofensivas como defensivas, ya que se puede generar variaciones de dominios con confusables que puedan inferir en un error del usuario y registrarlos o detectarlos.

La herramienta permite utilizar el diccionario creado (lo recomendado) o variaciones de los caracteres en un función de un % de similitud (utilizando directamente el modelo de

[22]https://github.com/next-security-lab/deep-confusables-cli/blob/master/deep_confusables_lite/confusables.txt

[23]https://github.com/next-security-lab/deep-confusables-cli

```
Spain: www.melia.com, www.bankia.com, www.inditex.com, www.meliá.com or www.solmeliá.com.

Global domains: amazón.es, góogle.es, skypê.net, skÿpe.net, skÿpe.net, skypè.net, skypé.ne,
fâcebook.net, fâcebook.net, facêbook.net, facẽbook.net, minecraft.net, expædia.com.ph,
twïtter.com, t-mobile.com, aliexpress.com, applè.com, ïkea.com, brazzers.com,
instagram.com, netflix.com, facebook.com, theguardian.com, ebáy.com, americanexpress.com,
adidas.com, sex.com, whatsàpp.com, samsung.com, àirbnb.com, nytimes.com, baïdu.com, office.com,
microsoft.com, wikipédia.com, disneylandparis.com, xvideos.com, amazon.com, microsoft.com,
dropbox.com, youporn.com, vodafone.com, icloud.com, expedia.com.ph, pornhub.com or netflix.com
```

Figura 8: Algunos dominios falsos de nuestras pruebas

deep learning implementado). Adicionalmente, la herramienta permite verificar si el dominio ya existe o si contiene contenido malicioso (utiliza para ello Virustotal).

V. Problemas en el mundo real

El desarrollo del diccionario de confusables y la herramienta *Deep Confusables* nos ha permitido iniciar un proceso de evaluación en servicios y funcionalidades diversas usando ataques basados en codificación Unicode. Es un trabajo en curso pero en este apartado se refleja alguno de los resultados actuales.

V-A. Detección de dominios potencialmente fraudulentos

Con el diccionario desarrollado, es posible descubrir nuevos dominios fraudulentos basados en la codificación de Unicode. Para ello, se ha llevado a cabo un escaneo masivo de dominios, centrándose en dominios españoles, principalmente las compañías del IBEX35 y pymes con un total de más de 26000 dominios escaneados. También se ha escaneado los 10000 sitios más visitados del mundo. Es notorio que de media se usaron 54 caracteres para crear dominios equivalentes, que suplantan al dominio legítimo. Por ejemplo, para la url `http://www.abc.es` se generaron 330 dominios falsos.

Nuestras pruebas muestran 12 dominios activos para los dominios españoles y 27876 para los 10000 dominios más visitados. Algunos de los ejemplos se pueden ver en la Fig. 8.

V-B. Implementación incorrecta de validación de caracteres Unicode

La industria, principalmente los grandes proveedores de tecnología, desarrollaron mecanismos, especialmente en navegadores web, para proteger al usuario final contra ataques de urls falsa basadas en Unicode. Para ello utilizan la representación punycode [43] e IDNA [44], que pasa los caracteres Unicode a ASCII, permitiendo así su identificación en navegadores.

Estas contramedidas son eficaces en navegadores, en el siguiente punto se destacará algún problema, y permiten al usuario final darse cuenta al menos que la url seleccionada no es la original. Esto aunque necesario no impide la ejecución de esa url fraudulenta que podría inyectar código javascript en el navegador o ejecutar algún exploit público o no para una versión concreta de navegador. Para minimizar este vector de ataque se debería extender estas contramedidas a las aplicaciones y sistemas que son usados para compartir cadenas de texto en codificación Unicode. Por ejemplo, enviar enlaces en aplicaciones de mensajería instántanea.

En nuestra investigación, en curso, se analizaron las principales herramientas de mensajería instantánea, redes sociales, herramientas ofimáticas y plataformas de correo web. Para ello se utilizó, entre otros, el dominio falso `www.example.org` (*a* en cirílico en codificación unicode) que imita al dominio verdadero `www.example.org`. Si un servicio o producto tiene protección frente a ataques de confusables la primera

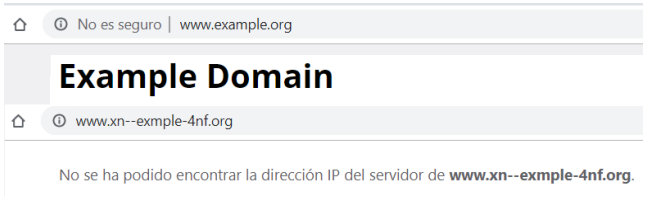


Figura 9: Visualizacion del dominio `www.example.org` (verdadero) y `www.example.org` (falso) donde aplica restricciones punycode.

Figura 10: Vulnerabilidad en Skype usando dominio falso por confusables: www.example.org (falso)

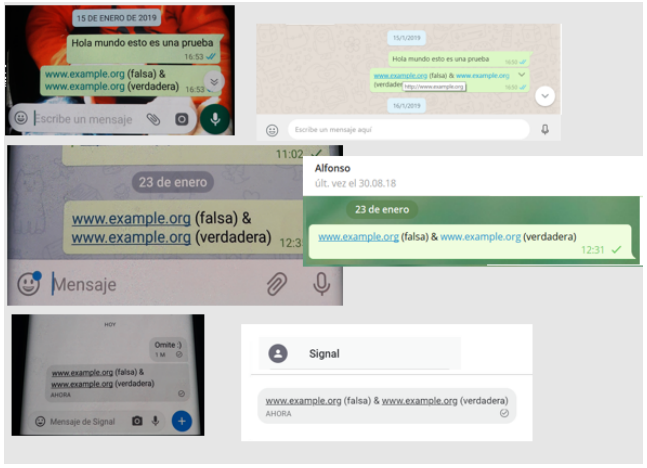


Figura 11: Vulnerabilidad en Whatsapp, Telegram y Signal usando dominio falso por confusables: `www.example.org` (falso)

dirección se debería mostrar de manera expandida como se puede ver en la Fig. 9.

En la actualidad, han sido detectados problemas que han sido comunicados a los fabricantes (bug bounties) para su corrección. Las úlimas versiones de Telegram (móvil y escritorio), Signal (móvil y escritorio), Whatsapp (móvil y escritorio), Skype (versión escritorio), Openoffice 4.1.6 y Foxit Reader pdf permiten engañar al usuario final mediante el uso de urls falsas usando *confusables* (Figs. 10, 11 y 12).

La ventaja del diccionario creado es la posibilidad de probar todos los confusables posibles de cada carácter para analizar la correcta validación de las entradas en servicios y

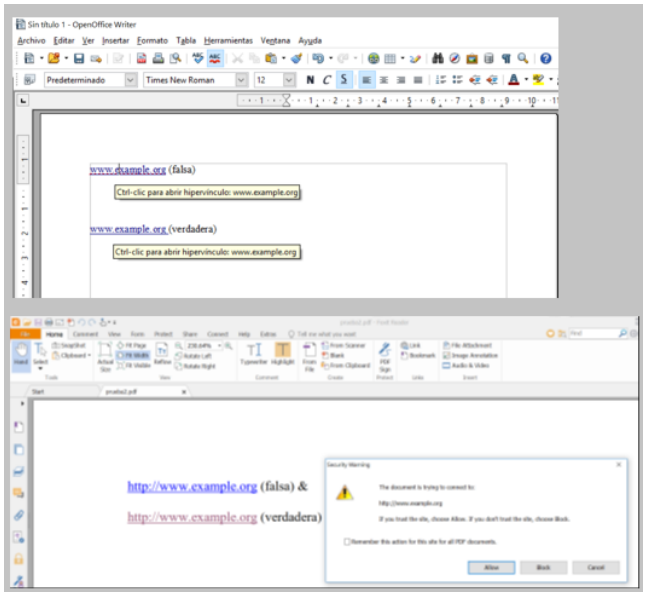


Figura 12: Vulnerabilidad en OpenOffice y FoxitReader usando dominio falso por confusables: `www.example.org` (falso)

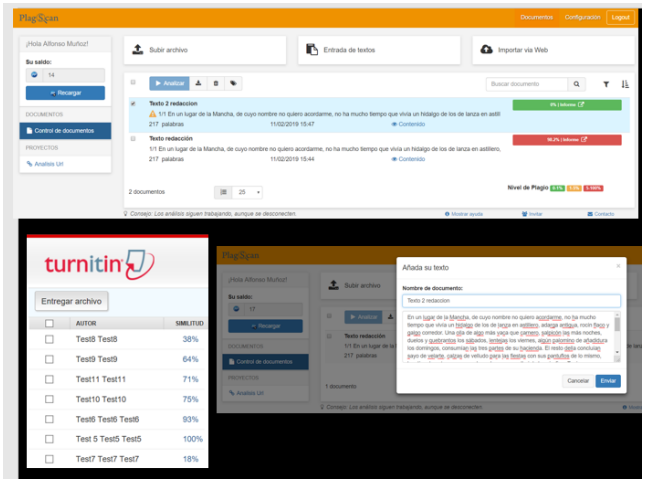


Figura 13: Evasión de mecanismos de propiedad intelectual basado en confusables

Confusables útiles: ą ę ġ ģ į ķ ļ ņ ŗ ş ţ ų
Dominio legítimo: https://2019.jnic.es/
Dominio falso: https://2019.jn̨ic.es/

Figura 14: Confusables útiles para evadir protección por defecto Punycode

productos. Como comentamos anteriormente esto va más allá de dominios web fraudulentos y tiene amplio uso.

Otro caso podría ser por ejemplo la evasión de sistemas de propiedad intelectual. Por ejemplo, recientemente en España se pusieron de moda estos sistemas al detectar plagio en la publicación de la tesis doctoral del presidente del gobierno Pedro Sánchez, el software utilizado fue Turnitin y Plagscan. Usando el diccionario creado es sencillo demostrar cómo evadir estos sistemas o reducir drásticamente su funcionalidad. En estos dos ejemplos se usó el prólogo del Quijote (un texto ampliamente reconocido). En el caso de PlagScan se cambió unicamente un carácter (por ejemplo todas las aes) por uno de sus confusables y fue suficiente para evadir el sistema. En el caso de Turnitin fue necesario modificar más de un carácter, pero pudo lograrse, con una simple prueba, reducir drásticamente su precisión (Fig. 13). Los fabricantes de estos productos han sido notificados. Estos son solo algunos de los ejemplos que se pueden deducir del uso del diccionario generado.

Todo lo anterior demuestra que ha día de hoy todavía la validación de caracteres Unicode es un tema de actualidad que debe ser tratado con precaución.

V-C. Evasión de contramedidas punycode

El apartado anterior destacó la importancia de trabajar en un modelo de seguridad basada en profundidad no dependiente exclusivamente de las medidas de seguridad que implemente un navegador web. Alguna razón ya ha sido enumerada: que el usuario observe que una url no es la legítima no implica que el código de esa página web asociada no se ejecute. El otro problema recae en certificar que siempre las contramedidas existentes como punycode nos ayudarán a detectar las urls fraudulentas.

En nuestra investigación y con la herramienta y diccionario generado demostramos cómo en determinados escenarios es posible esquivar las medidas de protección por defecto (punycode) existente en los navegadores actuales (Chrome, Firefox, Opera, Safari e Internet Explorer/Edge). Para ello simplemente es necesario centrase en urls con caracteres que pertenezcan a un mismo bloque unicode. Por ejemplo, la herramienta `Deep Confusables` genera dominios similares usando los bloques Unicode de Ascii y Latin-1. Estos resultados no son expandidos y los navegadores modernos no detectarían el fraude. Por ejemplo, la sustitución de caracteres en una url `a,e,g,i,k,l,n,r,s,t,u` por ciertos confusables no serán detectados. Véase en cualquier navegador el ejemplo con la url de la conferencia JNIC (Fig. 14).

A día de hoy sería posible realizar ataques con este tipo de propuestas y el problema viene realizado con la forma que tienen los navegadores web de entender si una url es legítima. En el caso de los navegadores más avanzados en protección en esta materia se puede destacar que actualmente este ataque se salta la política IDN de Chrome y en el caso Firefox es necesario forzar via configuración que siempre se expanda cualquier cadena unicode (`about:config` y `network.IDN_show_punycode` a `true`).

VI. Conclusiones

En esta investigación se demuestra la utilidad real de utilizar procedimientos de machine learning y deep learning en ciberseguridad, en concreto, en aspectos ofensivos. Utilizar esta aproximación es interesante para modelar distintos tipos de atacantes e intentar proponer mejores contramedidas. Nuestro trabajo se ha centrado en la creación del mejor diccionario de confusables existente a día de hoy, utilizando para ello un proceso semiautomático usando deep learning y transfer learning. Este diccionario puede ser útil para auditar la validación

de datos en multitud de sistemas, servicios y productos, entre ellos: generación de dominios falsos, evasion de tecnologías de propiedad intelectual, dlp, creación de covert channels, etc. Este diccionario ha permitido detectar fallos en grandes proveedores de servicios como Telegram, Whatsapp, Signal, Skype, etc., así como la evasión de mecanismos famosos de verificación de propiedad intelectual.

Los autores ponen a disposición de la comunidad científica este diccionario para facilitar la validación frente a entradas Unicode "maliciosas" con la esperanza de que surjan propuestas de contramedidas mejoradas.

Referencias

[1] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[3] B. Hariharan, P. Arbelaez, R. B. Girshick, and J. Malik, "Simultaneous detection and segmentation," *CoRR*, vol. abs/1407.1808, 2014. [Online]. Available: http://arxiv.org/abs/1407.1808

[4] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub, "Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, pp. 68 EP –, Jan 2002, article. [Online]. Available: http://dx.doi.org/10.1038/nm0102-68

[5] R. Kaundal, A. S. Kapoor, and G. P. S. Raghava, "Machine learning techniques in disease forecasting: a case study on rice blast prediction." *BMC Bioinformatics*, vol. 7, p. 485, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/bmcbi/bmcbi7.html#KaundalKR06

[6] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, and J. Tohka, "Machine learning framework for early mri-based alzheimer's conversion prediction in mci subjects," *NeuroImage*, vol. 104, pp. 398 – 412, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811914008131

[7] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8 – 17, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2001037014000464

[8] H. Wu, H. Wang, and Y. Shi, "Can machine learn steganography? - implementing LSB substitution and matrix coding steganography with feed-forward neural networks," *CoRR*, vol. abs/1606.05294, 2016. [Online]. Available: http://arxiv.org/abs/1606.05294

[9] D. Volkhonskiy, I. Nazarov, B. Borisenko, and E. Burnaev, "Steganographic generative adversarial networks," *CoRR*, vol. abs/1703.05502, 2017. [Online]. Available: http://arxiv.org/abs/1703.05502

[10] J. Hayes and G. Danezis, "Generating Steganographic Images via Adversarial Training," *ArXiv e-prints*, Mar. 2017.

[11] D. K. Bhattacharyya and J. K. Kalita, *Network Anomaly Detection: A Machine Learning Perspective*. Chapman & Hall/CRC, 2013.

[12] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, *Network Traffic Anomaly Detection and Prevention - Concepts, Techniques, and Tools*, ser. Computer Communications and Networks. Springer, 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-65188-0

[13] F. Iglesias and T. Zseby, "Analysis of network traffic features for anomaly detection," *Machine Learning*, vol. 101, no. 1, pp. 59–84, Oct 2015. [Online]. Available: https://doi.org/10.1007/s10994-014-5473-9

[14] B. Cakir and E. Dogdu, "Malware classification using deep learning methods," in *Proceedings of the ACMSE 2018 Conference*, ser. ACMSE '18. New York, NY, USA: ACM, 2018, pp. 10:1–10:5. [Online]. Available: http://doi.acm.org/10.1145/3190645.3190692

[15] L. Liu, B.-s. Wang, B. Yu, and Q.-x. Zhong, "Automatic malware classification and new malware detection using machine learning," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 9, pp. 1336–1347, Sep 2017. [Online]. Available: https://doi.org/10.1631/FITEE.1601325

[16] N. Peiravian and X. Zhu, "Machine learning for android malware detection using permission and api calls," in *Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, ser. ICTAI '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 300–305. [Online]. Available: http://dx.doi.org/10.1109/ICTAI.2013.53

[17] A. Bansal and S. Mahapatra, "A comparative analysis of machine learning techniques for botnet detection," in *Proceedings of the 10th International Conference on Security of Information and Networks*, ser. SIN '17. New York, NY, USA: ACM, 2017, pp. 91–98. [Online]. Available: http://doi.acm.org/10.1145/3136825.3136874

[18] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," in *Proceedings of the Conference on Design, Automation & Test in Europe*, ser. DATE '17. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2017, pp. 169–174. [Online]. Available: http://dl.acm.org/citation.cfm?id=3130379.3130417

[19] G. Grieco, G. L. Grinblat, L. C. Uzal, S. Rawat, J. Feist, and L. Mounier, "Toward large-scale vulnerability discovery using machine learning," in *CODASPY*, 2016.

[20] B. Chernis and R. Verma, "Machine learning methods for software vulnerability detection," in *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, ser. IWSPA '18. New York, NY, USA: ACM, 2018, pp. 31–39. [Online]. Available: http://doi.acm.org/10.1145/3180445.3180453

[21] Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, and Y. Zhong, "Vuldeepecker: A deep learning-based system for vulnerability detection," *CoRR*, vol. abs/1801.01681, 2018. [Online]. Available: http://arxiv.org/abs/1801.01681

[22] F. Wu, J. Wang, J. Liu, and W. Wang, "Vulnerability detection with deep learning," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, Dec 2017, pp. 1298–1302.

[23] W. Brendel, J. Rauber, and M. Bethge, "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," *ArXiv e-prints*, Dec. 2017.

[24] A. Shafahi, W. Ronny Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," *ArXiv e-prints*, Apr. 2018.

[25] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe, "Adversarial Attacks and Defences Competition," *ArXiv e-prints*, Mar. 2018.

[26] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 601–618. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer

[27] C. Liao, H. Zhong, A. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," 2018.

[28] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2012.

[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[30] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[31] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," https://www.tensorflow.org/, 2015.

[33] F. Seide and A. Agarwal, "Cntk: Microsoft's open-source deep-learning toolkit," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 2135–2135. [Online]. Available: http://doi.acm.org/10.1145/2939672.2945397

[34] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[35] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” 2016.
[36] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 2015.
[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017.
[39] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” 2016.
[40] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” 2017.
[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” 2018.
[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
[43] A. M. Costello, “Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA),” RFC 3492, Mar. 2003. [Online]. Available: https://rfc-editor.org/rfc/rfc3492.txt
[44] D. J. C. Klensin, “Internationalized Domain Names in Applications (IDNA): Protocol,” RFC 5891, Aug. 2010. [Online]. Available: https://rfc-editor.org/rfc/rfc5891.txt

Evaluación de algoritmos de clasificación para la detección de ataques en red sobre conjuntos de datos reales: *UGR'16 dataset* como caso de estudio

Ignacio Diaz-Cano
Dpto. Ing. en Automática, Electrónica, Arquitectura y Redes, ESI
Universidad de Cádiz
{ignacio.diaz@uca.es}

Roberto Magán-Carrión
Dpto. de Ingeniería Informática, ESI
Universidad de Cádiz
Network Engineering & Security Group
Universidad de Granada
{roberto.magan@uca.es,rmagan@ugr.es}

***Resumen*—La evaluación y rendimiento de sistemas de detección de intrusiones normalmente se realiza mediante el empleo de conjuntos de datos previamente obtenidos dentro del mismo contexto, entorno y aplicación en donde se implantará el sistema final. En concreto, para NIDS, existe un gran número de soluciones al respecto que, sin embargo, basan su rendimiento en conjuntos de datos no adecuados, bien por que están obsoletos, o por su representatividad, o realismo, entre otros. En el presente trabajo se introduce la problemática anterior y se presenta un estudio y evaluación de sistemas NIDS basados en algoritmos tradicionales de clasificación supervisada utilizando, ahora sí, conjuntos de datos adecuados como el recientemente creado conjunto de datos de red UGR'16.**

***Index Terms*—NIDS, aprendizaje supervisado, network, dataset**

Tipo de contribución: *Investigación original*

I. Introducción

Una parte importante en el robustecimiento de sistemas y redes de comunicaciones desde el punto de vista de la seguridad es la detección de comportamientos anómalos o ataques. Para tal fin, son los sistemas de detección de intrusiones (IDS) los más utilizados.

Existen varios tipos de IDS [1], de los que destacamos principalmente, los *Network* IDS (NIDS) y los *Host* IDS (HIDS). Los primeros se centran en analizar flujos y/o trazas de tráfico provenientes de *firewalls* u otros dispositivos de red principalmente. Por otro lado, los HIDS tienen en cuenta fuentes de datos e información del propio sistema en donde se despliega, como son, ficheros *syslog*, carga de CPU, etc. Con respecto a la forma en la que realizan el proceso de detección, también se puede diferenciar entre IDS basados en firmas (S-IDS) o patrones de tráfico IDS (A-IDS). Los primeros basan su funcionamiento en la comparación de una serie de reglas predefinidas para decidir si un comportamiento observado es sospechoso o no, mientras que los segundos [2] consideran comportamientos anómalos aquellos que se desvían en cierta medida del comportamiento normal del sistema o red de comunicaciones.

La evaluación de este tipo de sistemas recae habitualmente en la utilización de conjuntos de datos obtenidos con antelación dentro del contexto y aplicación objetivos. Las características del conjunto de datos, condiciona fuertemente la elección de los métodos, algoritmos o técnicas sobre las que se basan los IDS. De forma general, y especialmente para NIDS (tipología de IDS en la que se centra el presente trabajo) [3], es común encontrar soluciones basadas en técnicas de aprendizaje automático supervisado, no supervisado o semi-supervisado.

No solo el conjunto de datos elegido condiciona el uso de una u otra técnica sino que dependiendo de, llamémoslo así, su "calidad" se pueden enmascarar los resultados de rendimiento de sistemas de detección que hacen uso de ellos. De hecho, los autores en [4] sacan a la luz determinadas deficiencias encontradas en conjuntos de datos de red existentes en la literatura, algunos de ellos de uso extendido en el contexto del trabajo como es el caso de KDDcup'99 [5] que, usados para la evaluación de NIDS [6] o [7], hacen que, al menos, se discuta la validez de los resultados obtenidos. Así, dichos autores en [4] construyen el conjunto de datos UGR'16 especialmente diseñado para la evaluación de NIDS y que mejora los anteriores en términos de duración, tamaño, representatividad del entorno, veracidad, documentación y disponibilidad, entre otros.

Con objeto de abordar la problemática anterior, en el presente trabajo se evalúan sistemas NIDS basados en técnicas de aprendizaje automático (o Machine Learning (ML) en su término en inglés) supervisado tradicionales ahora sí, sobre conjuntos de datos adecuados, como es el caso del conjunto UGR'16. Por lo tanto el trabajo presentado aquí primero, introduce la problemática existente de la elección de conjunto de datos adecuados para detección y clasificación de ataques en red y, segundo, supone una primera aproximación al problema que dará pie a discutir sobre la veracidad y validez de conclusiones obtenidas de trabajos existentes en la literatura que se basan en la utilización de *dataset* no adecuados.

La estructura de este documento es la siguiente: en la Sección II se describen una serie de trabajos relacionados y que usan otros conjuntos de datos de red. A lo largo de la Sección III se introduce y describe el *dataset* UGR'16. Los diferentes pasos y procedimientos a seguir tanto para el tratamiento del conjunto de datos y su adecuación al problema, como para el entrenamiento y evaluación de los algoritmos de clasificación utilizados, se muestran en la Sección IV. En la Sección V se muestran los resultados obtenidos y la discusión de los mismos, terminando en la Sección VI con las conclusiones extraídas y la líneas futuras por las que podría discurrir el

estudio presentado.

II. Trabajos relacionados

La aplicación de algoritmos o técnicas de ML (Machine Learning) se ha utilizado y se utiliza de manera generalizada para el tratamiento de datos en diferentes campos y aplicaciones y, como no podría ser de otra manera, también en ciberseguridad [8]. La aplicación de una u otra técnica depende en gran parte del contexto del problema y de la naturaleza y forma de los datos a tratar. De esta manera, podríamos hablar de métodos de aprendizaje supervisado, no supervisado y semi-supervisado [3]. La aplicación de los primeros, no tiene sentido sin la utilización de datos etiquetados. Con respecto a métodos no supervisados, cuya aplicación principal es la de detectar comportamientos anómalos que se desvían en cierta forma y sentido del comportamiento "normal", el etiquetado de los datos es secundario aunque necesario en la validación de los modelos. Por último, enfoques semi-supervisados aportan beneficios en la detección de eventos no conocidos (parte no supervisada) y su posterior clasificación (parte supervisada) [9].

En el contexto del aprendizaje supervisado y la detección de eventos de seguridad en red existen varios trabajos en los que los autores aplican diferentes técnicas y algoritmos sobre determinados conjuntos de datos. Por ejemplo, en [6] los autores evalúan diferentes técnicas sobre el bien conocido conjunto KDDCup'99 que muestran las deficiencias de dicho conjunto de datos y como éstas afectan al rendimiento obtenido. Los autores del trabajo [7] evalúan el conjunto de datos USNW-NB15 [10] en función del rendimiento que ofrecen diferentes algoritmos ML. Dichos autores concluyen que USNW-NB15 es un conjunto válido para la comparación y evaluación de NIDS. En la misma línea y más recientemente, los autores de [11] evalúan algoritmos similares sobre los anteriores conjuntos de datos y cómo el pre- y pos-procesamiento de estos influye en el rendimiento de dichas técnicas.

La "calidad" en ciertos términos de un conjunto de datos determina, como no podría ser de otra manera, el rendimiento de cualquier algoritmo de aprendizaje. De hecho, los autores del conjunto de datos UGR'16 [4] estudian en detalle el estado del arte en conjuntos de datos de red y concluyen la falta de estos para la comparación justa entre algoritmos y sistemas de detección de eventos de seguridad en red o NIDS.

Dicho esto, podríamos aventurarnos a decir qué conclusiones y resultados de trabajos anteriores en la evaluación y rendimiento de técnicas y algoritmos de ML en NIDS no son del todo válidos, completos o definitivos. De hecho, en el recientemente publicado trabajo [12], los autores ponen de manifiesto este problema y constatan cómo trabajos también publicados recientemente, todavía basan su conclusiones y resultados en conjuntos de datos obsoletos como el ya mencionado KDDCup'99.

Son las anteriores razones las que nos llevan a evaluar en el presente trabajo aquellas técnicas ML previamente usadas en otros conjuntos de datos aunque ahora sobre el conjunto UGR'16 que, entre otras características, lo diferencia del resto, como se abordará en la Sección III, en la representatividad del entorno en donde se recoge, su realismo, su duración, su frescura y, no menos importante, en su disponibilidad y documentación.

III. Descripción del dataset: UGR'16

UGR'16[1] se compone de 4 meses de trazas de tráfico de red *netflow* anonimizadas recogidas dentro de la infraestructura de red de un PSI (Proveedor de Servicios de Internet) español. Sus creadores, lo dividen en dos partes de duración y tamaño diferentes: CAL y TEST. Con respecto a la primera, que comprende desde marzo hasta junio de 2016, se captura tráfico procedente del propio uso del ISP, sus clientes y la interacción con y desde el exterior. Con respecto a la segunda, es aquí donde se suman los flujos de tráfico procedente de la ejecución de ataques entre máquinas y entornos controladas dentro del ISP como son *low-rate DoS*, *Scan* o *Botnet*. Adicionalmente, una inspección manual de algunos conjuntos de flujos en donde varios detectores sugerían posibles anomalías ayudaron a la identificación de eventos de seguridad o ataques en todo el conjunto, como aquellos del tipo *UDP scan*, *SSH scan* o campañas de *Spam*. Además, no solo se etiquetaron todos aquellos flujos de tráfico que se corresponden con los anteriores ataques, si no que aquellos cuyas IP se encuentran dadas de alta en conocidas listas *Blacklist* también lo fueron.

En total, UGR'16 se compone de 23 ficheros (17 para el subconjunto CAL y 6 para el de TEST) que corresponden con cada una de las semanas implicadas. En cuanto a su tamaño, cada uno de ellos ocupa alrededor de 14 GB (comprimidos). Todos ellos están disponibles tanto en formato *nfcapd* como en CSV, contemplando este último un número reducido de las características originales. Para aquellos lectores interesados, se recomienda leer el artículo [4] en donde se proporciona más información al respecto.

Un análisis preliminar del conjunto de datos y su distribución en clases se muestra de forma gráfica, en la Figura 1 y numérica, en la Tabla I. Como era de esperar, y un patrón que se repite en el contexto del problema, es la existencia de grandes diferencias entre el número de flujos que se corresponden con ataques y aquellos que no lo son (*Background*). De hecho el porcentaje de la clase minoritaria con mayor número de observaciones asciende a un $1,96\,\%$ del total de flujos del subconjunto de TEST (ataque *Spam*).

La problemática anterior es sin duda relevante sobre todo en el contexto del problema en el que se enmarca el presente trabajo: clasificación supervisada de ataques. Este aspecto junto con el elevado número de flujos a tratar condicionan el rendimiento de cualquier algoritmo de ML que, por regla general, tiende a incrementar la tasa de falsos negativos obtenida. Consecuentemente, se hace necesaria la provisión de métodos que minimicen estas desigualdades y además aborden el manejo de tal cantidad de información, sobre todo, de cara a validar soluciones en entornos productivos.

Como se observa en la Tabla I, es en el subconjunto al que los autores llaman TEST, en donde se contemplan todas y cada una de las tipologías de ataque contempladas. Por esta razón, tanto la metodología como los resultados que se obtienen en las secciones correspondientes se refieren a este subconjunto.

IV. Metodología

La metodología y procedimientos que se han seguido para la utilización y adecuación del conjunto de datos a los dife-

[1]Disponible en https://nesg.ugr.es/nesg-ugr16/

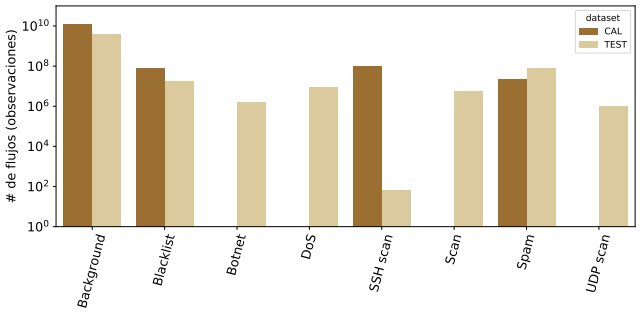


Figura 1. Distribución de clases del conjunto de datos UGR'16

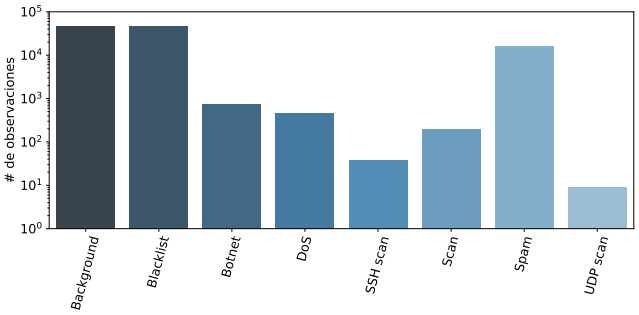


Figura 2. Distribución de clases tras la aproximación FaaC.

Tabla I
DISTRIBUCIÓN DE CLASES EN UGR'16 EN NÚMERO DE FLUJOS *netflow*.

Clase	CAL	%	TEST	%
Background	$\sim 13,000M$	$98,4$	$\sim 4,000M$	$97,14$
Blacklist	$\sim 80M$	$0,62$	$\sim 18M$	$0,46$
Botnet	0	0	$\sim 2M$	$0,04$
DoS	0	0	$\sim 9M$	$0,23$
SSH scan	$\sim 105M$	$0,81$	64	~ 0
Scan	0	0	$\sim 6M$	$0,14$
Spam	$\sim 24M$	$0,18$	$\sim 78M$	$1,96$
UDP scan	0	0	$\sim 1M$	$0,03$

Tabla II
NUEVAS VARIABLES TRAS LA APLICACIÓN DE LA APROXIMACIÓN FAAC

Descripción	Cantidad	Valores
IP origen	2	*public, private*
IP destino	2	*public, private*
Puerto origen	52	HTTP, SMTP, SNMP, ...
Puerto destino	52	HTTP, SMTP, SNMP, ...
Protocolo	5	TCP, UDP, ICMP, IGMP, Otro
Flags	6	A, S, F, R, P, U
ToS	3	0, 192, Otro
# paquetes	5	*very low, low, medium, high, very high*
# bytes	5	*very low, low, medium, high, very high*

rentes modelos y algoritmos ML así como para la evaluación de estos, se resume en la Figura 3.

Dependiendo de las acciones y procesos realizados sobre el conjunto de datos original, se obtienen conjuntos derivados sobre los que se trabajará. Estos son:

- *UGR'16-binario* (clasificación binaria). Para cada uno de los ataques se crea un nuevo conjunto donde las ocurrencias del ataque en cuestión se etiquetan de forma general como *ataque* mientras que las demás lo hace como *no ataque*.
- *UGR'16-MC* (clasificación multi-clase) A cada ataque se le asigna una clase así como al tráfico *Background*. Para solventar la problemática de la ocurrencia varios ataques en una misma observación, se elige aquel que predomina en la agregación realizada en el proceso de *parsing* que se verá más adelante.
- *UGR'16-MC-SMOTE* (clasificación multi-clase). Con objeto de mitigar el efecto que tuviera en el rendimiento de clasificación el desbalanceo existente entre clases, se aplica aquí la técnica SMOTE (Syntethic Minority Over-Sampling Technique)[8] sobre el conjunto UGR'16-MC.

A continuación se describen cada una de las etapas de las que consta la metodología de trabajo propuesta.

IV-A. Parsing and feature engineering

La información proveniente de tráfico de red, normalmente en términos de flujos de comunicaciones o trazas de *logs*, se puede encontrar estructurada o no siendo muy heterogénea y de distinta tipología. Normalmente, dicha información no se puede utilizar directamente como fuente de datos para algoritmos y/o sistemas de detección o clasificación en general. Así, es necesario el pre-procesado de dicha información para adecuarla a las necesidades primero, del problema abordado y segundo, a los métodos de clasificación y detección de anomalías tanto en forma como en tiempo. Si bien existen algunas aproximaciones como los modelos *word2vec*, muy utilizados en procesado de lenguaje natural o algunas más simplistas, como *one-hot-encoding*. En el presente trabajo se utilizará la aproximación FaaC (Feature-as-a-Counter) que introducen los autores por primera vez en [13]. FaaC, entre otras funcionalidades, aborda algunas de las problemáticas existentes en el tratamiento de grandes cantidades de datos de diferentes fuentes de información. Principalmente, transforma y fusiona diferentes fuentes de información y sus variables en otras nuevas que son contadores de las anteriores dentro de un determinado periodo de tiempo. Por ejemplo, dependiendo del problema abordado, podría ser interesante obtener el número de veces que se accede al puerto 22 durante un cierto intervalo de tiempo concluyendo, quizá, que un número alto de accesos a dicho puerto se podría deber a un ataque de fuerza bruta SSH.

En concreto, y para nuestro trabajo, ampliamos el conjunto de variables de 11 variables que nos ofrece el conjunto UGR'16 de las trazas *Netflow* a las 132 variables que se muestran en la Tabla II.

Esta transformación se traduce en una reducción en el número de observaciones (cada una agrega la información correspondiente a un minuto del conjunto original) tal y como se observa en la nueva distribución de clases de la Figura 2, donde aparece aplicada la aproximación Facc, viendo cómo se reduce, debido a esa agrupación por minuto, el número de trazas, respecto a las que aparecen en forma de *netflow* en la Figura 1. En lo que respecta a las etiquetas de los ataques, esta agregación se traduce en nuevos contadores a su vez, que determinan el número de veces que se etiquetó una determinada clase en un periodo considerado. Nótese que ahora se han de abordar situaciones en las que más de un ataque aparece en la misma observación, casuística que en el conjunto original no se contemplaba.

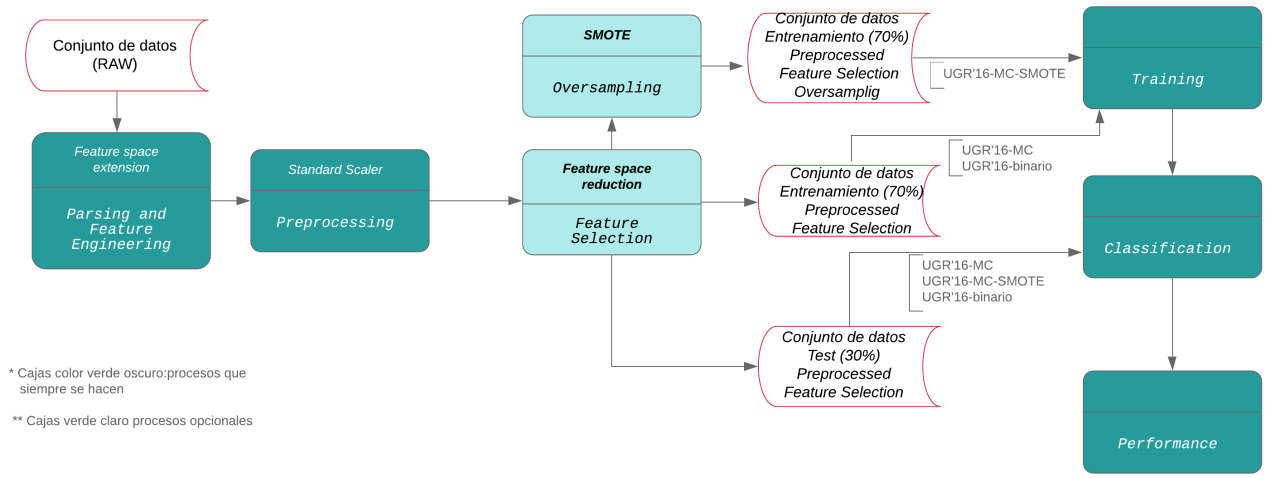


Figura 3. *Pipeline* para la adecuación a del conjunto de datos y la evaluación de los algoritmos de ML propuestos.

Según los autores de la metodología MSNM (Multivariate Statistical Network Monitoring) [14] para la detección de anomalías con PCA (Principal Component Analysis) no supervisada, FaaC es una aproximación adecuada para la problemática abordada en este trabajo y que solventa problemáticas actuales relativas al procesamiento y gestión de grandes cantidades de información: variedad, volumen, veracidad, procesamiento en tiempo real, etc. Veremos aquí como afecta este enfoque en su uso junto con métodos de clasificación supervisada tradicionales [8].

IV-B. Pre-processing

Con objeto de equiparar el rango de valores del nuevo conjunto de variables y evitar así resultados no deseados debido a la disparidad existente en sus valores de ellas, se normaliza el conjunto de datos.

IV-C. Feature Selection

Para evitar situaciones de *overfitting* y eliminar aquellas variables que no aportan información relevante en la decisión final, se delega esta decisión a la votación mayoritaria obtenida tras el empleo de tres conocidos métodos de selección de variables: *Lasso, Recursive Feature Elimination (REF) y Select From Model* [15][8].

IV-D. Oversampling

Una problemática común existente en el contexto del problema que nos ocupa, es el desbalanceo existente entre clases. En la Tabla III se observa este hecho, donde aparece el número de observaciones clasificadas por clases, que aparecen en el conjunto de datos de entrenamiento, observándose lo que venimos comentando sobre la desproporción, lógica por otra parte de un caso real, en cuanto al número de tráfico normal y el anómalo. Con objeto de analizar y solventar (en su caso) el impacto que tiene sobre la decisión final de los clasificadores dicho desbalanceo de clases, se emplea el enfoque SMOTE (Syntethic Minority Over-Sampling Technique) muy utilizado en la literatura [8], que incrementa el número de observaciones de las clases minoritarias de forma sintética al de la mayoritaria, en este caso, *Background*.

Tabla III
DESBALANCEO DE CLASES EN EL CONJUNTO DE ENTRENAMIENTO UGR'16-MC.

Clase	Nº de observaciones
Background	16,860
DoS	259
Botnet	406
Scan	99
UDPscan	7
SSHscan	22
Spam	8,820

IV-E. Training

Una fase fundamental y que determina el rendimiento de algoritmos y técnicas de clasificación y detección en general, es el entrenamiento de estos.

Para ello es usual utilizar una gran parte del conjunto de datos, dejando el resto para evaluar su rendimiento y capacidad de generalización ante observaciones que no ha contemplado durante la fase de entrenamiento (conjunto de *test*). En nuestro caso, y ya que el número de observaciones no es muy elevado, se dedica el 70 % de todo el conjunto a entrenamiento, siendo el resto para el conjunto de test.

Otro aspecto importante a tener en cuenta es la selección de hiper-parámetros de los modelos ML seleccionados que, como no podía ser de otra forma, depende del propio algoritmo ML utilizado. Se han utilizado aquí métodos tradicionales de clasificación supervisada como son: SVM[16], Decision Tree [17], Random Forest [18], SVM Lineal [19] y Logistic Regression [20].

Son varios los procedimientos existentes para obtener el conjunto de hiper-parámetros que mejor rendimiento ofrezca para un determinado algoritmo de clasificación y un determinado conjunto de datos. Una aproximación es la prueba exhaustiva de conjuntos de hiper-parámetros pre-seleccionados que, por ejemplo, se realiza con el método *Grid Search* [11] [8]. El conjunto de hiper-parámetros optimizado para cada uno de los algoritmos propuestos, aplicando una validación cruzada con $k = 5$ se observa a continuación:

- *Support Vector Machine (SVM)*. Se tomó como mejor valor para el *kernel*, *Radial Basis Function kernel*. Además se tomó C con el valor de 10.
- *Decision Tree*. Se establece una profundidad máxima de 30.
- *Radom Forest*. Se establece una profundidad máxima de 50 para cada uno de los 100 árboles contemplados.
- *Linear SVM*. El método de selección tomó en este caso un valor de C igual a 10.
- *Logistic Regression*. Se establece regularización $l1$, C=1 y algoritmo de optimización L-BFGS.

IV-F. Testing

Antes mencionado, de cara a evaluar la capacidad de clasificación del método utilizado ante la presencia de información no contemplada en el proceso de entrenamiento, se destina el 30 % del conjunto de datos a este fin. Es durante esta fase donde se validará la adecuación o no de los métodos propuestos de cara a ser explotados en un entorno real.

Para ello, se propone la utilización de una serie de métricas de rendimiento que se exponen a continuación.

IV-G. Métricas de rendimiento

Para la evaluación del rendimiento de soluciones de detección NIDS se han venido utilizando varias métricas [3]. Algunas de ellas, que han sido utilizadas en el presente trabajo son:

- *Recall*. Gracias a ésta podemos evaluar la capacidad del método para clasificar ataques correctamente. Es también conocido como la tasa de verdaderos positivos (o True Positive Rate (TPR)). Se define como,

$$Recall = \frac{VP}{VP + FN} \tag{1}$$

  y relaciona el número de Verdaderos Positivos (VP) y Falsos Negativos (FN).
- *Precision*. A diferencia de la anterior, evalúa la capacidad del clasificador para no cometer errores al clasificar tráfico normal como ataques. Se define como,

$$Precision = \frac{VP}{VP + FP} \tag{2}$$

  y relaciona el número de VP con el de Falsos Positivos (FP).
- *F1-score*. Representa la media armónica para los dos anteriores valores. Un valor de *F1-score* alto, cercano a 1, se traduce en un buen rendimiento del sistemas de clasificación. Se define como,

$$F1 - score = \frac{2 \times P \times R}{P + R} \tag{3}$$

  done P es la *Precision* y R es *Recall*.

V. Resultados y discusión

Para cada uno de los algoritmos seleccionados en la Sección IV-E se evaluará su rendimiento siguiendo la metodología expuesta en la Sección IV. Todos los algoritmos y utilidades para el tratamiento de datos se obtienen del paquete *scikit-learn* de *Python* (versión 0.20.1) [15].

Los resultados de *F1-score* obtenidos para cada clasificador y conjunto de datos derivados de la metodología mencionada en la Sección IV, se muestran de forma gráfica en las Figuras 4, 5 y 6. Así mismo, de forma detallada en las Tablas V y IV se muestran el rendimiento obtenido en términos de *F1-score*, *Recall* y *Precision*.

En general, se observa un rendimiento muy alto en la clasificación de tráfico *Background*, más acuciada aún en el conjunto de datos UGR16-binario. Esto se debe principalmente al desbalanceo de clases existente que, para los conjuntos de datos binarios, es todavía más notable. También de forma general, se observa un comportamiento similar para casi todos los clasificadores cuando se aplica selección de variables y *oversampling* (UGR'16-MC-SMOTE) en comparación con los conjuntos UGR'16-MC UGR'16-binario. Esto pone en evidencia que no todas las variables del conjunto son relevantes para la clasificación y que balancear correctamente las clases, no solo en detección de ataques en red si no en cualquier problema de clasificación supervisada indice directamente sobre el rendimiento ofrecido.

También de forma general, ataques con menos observaciones, son más difíciles de detectar. Este es el caso de los ataques *UDPscan* y *SSHscan* incluso tras el balanceo de clases.

Dentro de las métricas, se puede observar cómo con la anomalía *Botnet* (basada en tráfico real), la puntuación muy baja con respecto a *Precision*, tanto en el experimento binario, como en el Multiclase. Sin embargo, en el conjunto de *SMOTE* la puntuación llega a ser muy pareja a la de *Precision*, lo que nos puede llevar a pensar que los clasificadores tienen problemas de forma sistemática para detectar esta anomalía, ya que puede ser confundida en ocasiones con tráfico normal, por su naturaleza. Por otro lado, nos inclinamos a pensar que el incremento de observaciones favorece la capacidad del clasificador para encontrar las muestras anómalas (*Recall*).

Las mediciones muestran que no hemos sido capaz de detectar de de ninguna manera el tráfico SSHscan, pensando que la razón pudiera ser porque el número de observaciones de esta clase es demasiado bajo, y necesita un entrenamiento más amplio para que el clasificador pueda aprender la detección de esta anomalía.

Clasificadores basados en árboles, como *Random Forest*, junto con métodos de clasificación no lineales presentan un mejor rendimiento en general. Este comportamiento mejora notablemente con el conjunto UGR'16-MC-SMOTE para casi

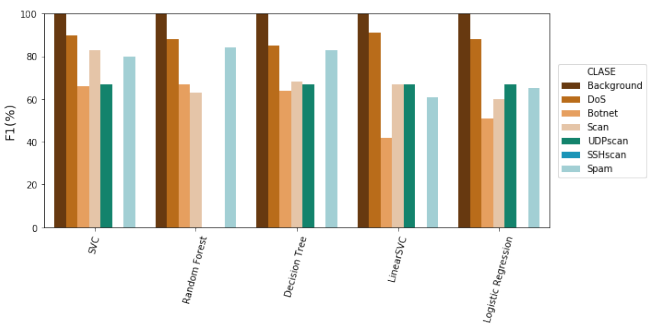


Figura 4. F1-score para cada clase y clasificador en UGR16-Binario

Tabla IV
RENDIMIENTO DE CLASIFICACIÓN PARA LOS ATAQUES *UDPscan*, *SSHscan*, *Spam*, Y TRÁFICO *Background*

Modelo	Dataset	Background			UDPscan			SSHscan			Spam		
		Pr.	Recall	F1	Precision	Rec.	F1	Pr.	Recall	F1	Precision	Rec.	F1
SVC	UGR16-binario	100	100	100	100	50	67	0	0	0	86	76	80
	UGR16-MC	87	93	90	100	50	67	0	0	0	85	76	80
	UGR16-MC-SMOTE	92	91	92	100	50	67	0	0	0	83	87	85
Random Forest	UGR16-binario	100	100	100	0	0	0	0	0	0	90	78	84
	UGR16-MC	88	95	92	100	50	67	0	0	0	88	78	83
	UGR16-MC-SMOTE	91	93	92	100	50	67	0	0	0	87	84	86
Decision Tree	UGR16-binario	100	100	100	100	50	67	0	0	0	75	75	75
	UGR16-MC	87	87	87	100	50	67	0	0	0	75	76	76
	UGR16-MC-SMOTE	87	83	85	20	50	29	0	0	0	71	76	73
LinearSVC	UGR16-binario	100	100	100	100	50	67	0	0	0	82	49	61
	UGR16-MC	76	94	84	0	0	0	0	0	0	80	47	59
	UGR16-MC-SMOTE	86	72	79	20	50	29	0	0	0	66	67	67
Logistic Regression	UGR16-binario	100	100	100	100	50	67	0	0	0	82	54	65
	UGR16-MC	79	93	86	0	0	0	0	0	0	80	56	66
	UGR16-MC-SMOTE	87	75	75	9	50	15	0	0	0	70	69	70

Tabla V
RENDIMIENTO DE CLASSIFICACIÓN PARA LOS ATAQUES *DoS*, *Botnet* Y *Scan*

Modelo	Dataset	DoS			Botnet			Scan		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SVC	UGR16-binario	97	85	90	94	51	66	100	70	90
	UGR16-MC	97	83	89	94	53	67	91	67	77
	UGR16-MC-SMOTE	99	80	89	95	85	89	93	89	91
Random Forest	UGR16-binario	99	79	88	98	50	67	100	46	63
	UGR16-MC	98	83	90	99	51	67	100	42	59
	UGR16-MC-SMOTE	99	89	94	98	83	90	93	91	90
Decision Tree	UGR16-binario	90	81	85	64	64	64	69	67	68
	UGR16-MC	86	77	81	69	58	63	40	53	46
	UGR16-MC-SMOTE	88	91	89	80	81	80	89	89	89
LinearSVC	UGR16-binario	96	87	91	75	29	42	78	59	67
	UGR16-MC	97	85	91	75	39	52	76	42	54
	UGR16-MC-SMOTE	71	91	80	46	88	60	50	93	55
Logistic Regression	UGR16-binario	94	83	88	84	37	51	81	48	60
	UGR16-MC	95	85	90	85	42	57	73	42	54
	UGR16-MC-SMOTE	64	93	76	44	88	59	67	93	78

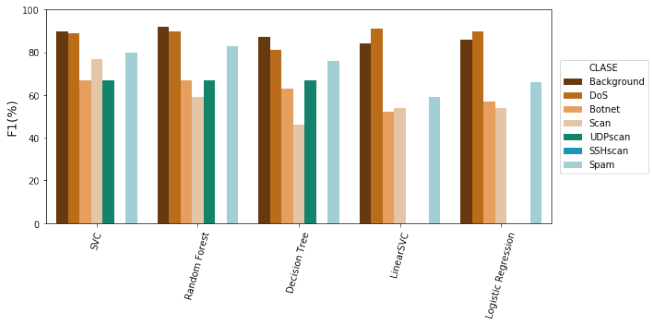


Figura 5. F1-score para cada clase y clasificador en UGR16-MC

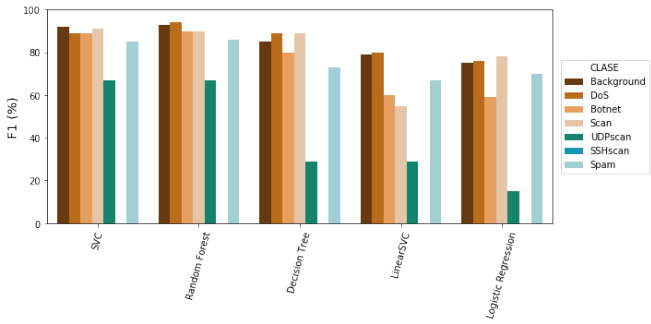


Figura 6. F1-score para cada clase y clasificador en UGR16-MC-SMOTE

todos los ataques. Esto es debido a la propia naturaleza no lineal del conjunto de datos *netflow* así como, en el caso de *Random Forest* su filosofía de trabajo en forma de *esemble* de varios sub-árboles.

VI. CONCLUSIONES Y TRABAJO FUTURO

El presente trabajo pretende ser una primera aproximación a una problemática común en en la evaluación de sistemas NIDS. Esta es, la utilización de conjuntos de datos no adecuados y que por tanto se traduce en resultados y conclusiones que al menos se podrían poner en entredicho. De esta forma, se ha escogido el recientemente creado conjunto de datos UGR'16 que solventa deficiencias encontradas en otros ampliamente utilizados y aceptados por la comunidad investigadora.

Junto con este conjunto de datos se han evaluado una serie de algoritmos de clasificación supervisada y se ha propuesto una metodología de trabajo que revela la importancia del correcto tratamiento del conjunto de datos previo al entrenamiento de los clasificadores. En concreto, las fases de *oversampling* y *feature selection*.

Después de los resultados obtenidos, hemos comprobado que la aproximación Faac, empleada en la Sección IV-A, dentro del enfoque de este experimento, clasificación supervisada, no ha afectado igualmente de forma positiva al mismo, al igual que en los problemas de aprendizaje no supervisado, donde sí se conocen resultados satisfactorios empleando esta aproximación[14].

Las métricas *Recall* y *Precision* deberán ser tenidas en cuenta con mayor interés en el caso que el contexto de nuestro sistema necesite una tasa de Falsos Negativos baja, es decir, un valor alto de *Recall*. Así, En las tablas IV y V se puede observar como el clasificador *Random Forest*, aplicado sobre el conjunto UGR16-MC-SMOTE ofrece unos resultados muy efectivos frente a la comparación con otros clasificadores en el resto de conjuntos del experimento. Sin embargo, un valor alto de *Precision* lo encontramos en el conjunto UGR16-binario, también en el clasificador *Random Forest*.

Como trabajo futuro se plantea la adición y evaluación de nuevas técnicas y algoritmos *deep learning* no solo en la clasificación en sí, si no como métodos alternativos para la solución FaaC de *feature enginering*. Además, y un paso determinante para validar le hecho de que conjuntos de datos no adecuados ofrecen resultados no concluyentes o no válidos en los sistemas que los utilizan, se prevé comparar los resultados obtenidos en el presente trabajo con algunos del estado del arte que utilizan conjunto de datos ampliamente aceptados por la comunidad investigadora.

Referencias

[1] Roberto Di Pietro and Luigi V. Mancini, Eds., *Intrusion Detection Systems*, ser. Advances in Information Security. Springer US, 2008, vol. 38.

[2] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, feb 2009.

[3] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[4] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs," *Computers & Security*, vol. 73, pp. 411–424, Mar. 2018.

[5] University od California, "KDD Cup 1999 Data," 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[6] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6.

[7] N. Moustafa and J. Slay, "The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set," *Inf. Sec. J.: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, Apr. 2016.

[8] D. Freeman and C. Chio, *Machine Learning and Security*. O'Reilly Media, Febrero 2018.

[9] J. Camacho, G. Maciá-Fernández, N. M. Fuentes-García, and E. Saccenti, "Semi-supervised Multivariate Statistical Network Monitoring for Learning Security Threats," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2019.

[10] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6.

[11] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, Oct. 2018, pp. 1–8.

[12] K. Siddique, Z. Akhtar, F. Aslam Khan, and Y. Kim, "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research," *Computer*, vol. 52, no. 2, pp. 41–51, feb 2019.

[13] J. Camacho, G. Maciá-Fernández, J. Díaz-Verdejo, and P. García-Teodoro, "Tackling the Big Data 4 vs for anomaly detection," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 500–505.

[14] J. Camacho, P. García-Teodoro, and G. Maciá-Fernández, "Traffic Monitoring and Diagnosis with Multivariate Statistical Network Monitoring: A Case Study," in *2017 IEEE Security and Privacy Workshops (SPW)*, May 2017, pp. 241–246.

[15] A. G. F. Pedregosa, G. Varoquaux, "Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot," Tech. Rep., 2011.

[16] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 1999, pp. 61–74.

[17] D. of Statistics, "Classification and Regression Trees," Carnegie Mellon University, Tech. Rep., 2009. [Online]. Available: https://www.stat.cmu.edu/{~}cshalizi/350/lectures/22/lecture-22.pdf

[18] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, oct 2001.

[19] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability Estimates for Multiclass Classification by Pairwise Coupling," NTU CSIE, Tech. Rep., 2004.

[20] C. Bishop, "Pattern recognition machine learning." Springer, New York, Inc, Information Science and Statistics, Berlin, 2006.

HIDS by signature for embedded devices in IoT networks

Bruno V. Dutra*, João F. de Alencastro†, Francisco L. de Caldas Filho‡, Lucas M. C. e Martins§, Rafael T. de Sousa Jr.¶ and Robson de O. Albuquerque‖
National Science and Technology Institute on Cyber Security, Electrical Engineering Department,
University of Brasília (UnB), P.O. Box 4466, Brasília–DF, Brazil, CEP 70910-900
Email: { bruno.dutra*, joao.alencastro†, francisco.lopes‡, lucas.martins§, rafael.desousa¶, robson‖} @redes.unb.br

***Abstract*—Cybersecurity in the Internet of Things (IoT) has become a major concern due to the huge number of vulnerable devices and the difficulty for the IoT middleware to completely block the interactions of these devices with other entities outside the IoT realm. Hence, one possible way to protect IoT is to enhance smart devices with intrusion detection capabilities considering their limited resources. With such assumptions in mind, this paper describes a host-based intrusion detection system (HIDS) by signature for IoT smart devices. In this HIDS the attack signatures remain in a central controller on the cloud, which is periodically consulted by the hosts/devices. The proposed system takes actions defined by the administrator to prevent vulnerable IoT devices to be compromised and thus to join botnets. In addition, the proposed system is also able to notify the IoT middleware about potential failure indicators.**

***Index Terms*—Internet of Things (IoT), IoT vulnerabilities, smart devices, host-based intrusion detection system (HIDS), intrusion detection rules, Mirai botnet.**

I. Introduction

Since the introduction of the term, in 1999 by Kevin Ashton [1], the Internet of Things (IoT) has been largely seen with an enormous potential for growth and development of new technologies. The IoT has the purpose of expanding the limits of the traditional computing, which uses desktops and conventional computers, to a new environment where common objects are interconnected to exchange data and to be remotely controlled. As described in [2], this new computational revolution arises from the connection between objects to create smart environments.

Nowadays IoT is experiencing an impressive growth and some projections estimate that by 2020 the number of devices connected to IoT instances will grow exponentially to 50 billion [3].

This growth and the popularization of IoT instances in the society has brought new cybersecurity threats, particularly due to the configuration of botnets based on IoT devices. Popular annual cybersecurity reports, such as [4] in 2018, point out the increasing scope and intensity of Distributed Denial of Service (DDoS) attacks coming from these botnets. These attacks are taking advantage of vulnerabilities in devices that are developed and deployed with no or little concern to basic security measures [5], as exemplified by the vulnerability of default passwords, that exposes a huge number of devices to botnets, such as the Mirai botnet [6], which uses this kind of devices to perform DDoS attacks.

An attack performed by botnets basically proceeds in two main steps. The first step consists in the attackers taking control of devices: an attacker will execute a search for vulnerable devices, which are often IoT devices with basic security vulnerabilities, then the found vulnerabilities are exploited to establish control over the device. The second step consists in manipulating the devices: the attacker orchestrates the botnet controlled devices to perform in concert and to achieve a specific objective [6]. Usually the manipulation of devices is disseminated and coordinated under the command of a single malicious agent.

Given the high risk and impacts related to this kind of vulnerability in IoT devices, this paper presents a Host-Based Intrusion Detection System (HIDS) that uses a signature approach for detecting vulnerabilities and attacks against devices in an IoT network instance. The proposal includes an operational IDS agent for each device and an HIDS controller that holds rules for signature-based intrusion detection, thus constituting a distributed security measure for the IoT instance, as described in a related previous work [7]. This IoT security measure is designed to count on the IoT middleware support to be implemented in the form of an application collaborating with device agents by means of fully distributed operations.

The proposed HIDS takes into consideration the heterogeneity of IoT devices regarding processing power. For instance, sensors often have very limited processing power, which constitute a challenge for their protection [8]. Other devices my operate under a UNIX-like operating system in a Raspberry Pi platform, thus presenting higher processing power. The security measure described in the present paper is designed to adapt to these different devices.

Besides this introduction, this paper is organized as follows: in Section II, we present a literature review about security in IoT and, in Section III, we present related works. In Section IV, we present the proposed HIDS for IoT instances and, in Section V, we highlight the proposed HIDS life cycle. In Section VI, we present the testing methodology and the results. Finally, in Section VII, we present general conclusions and suggestions for future work.

II. State of the Art

Since this paper focuses on security issues in the IoT domain, this section presents and discusses academic and in-

dustry views of some concepts that are useful for understating the paper content.

A. IoT Architecture

IoT solutions are being projected for and used in several domains like agriculture, transportation, logistics, industry, smart grid, home automation, surveillance, health care and personal assistance [2]. Notwithstanding that each one of these domains has its own characteristics, there are two common factors among them: the heterogeneity of technologies and the large number of entities in the solution. Since IoT is supposed to be a ubiquitous solution, the used devices need to be cheaper, smaller and more abundant than conventional ones.

IoT instances can be viewed as a layered architecture composed by devices, network connectivity, middleware and applications. The devices layer has all sort of devices acting as sensors and/or actuators, and including smart devices. The network connectivity layer embraces the network infrastructure to make possible the devices to connect to the middleware. The middleware layer corresponds to a software or a set of software that supports devices activities, for instance, by providing storage and processing capabilities. The applications layer is composed by services and applications that interact with data and actions provided by the devices.

B. UIoT Middleware

The UIoT Middleware [9] [10] was proposed to control and notify the current state of generic devices and evolved to be a cloud-based IoT middleware that is capable to store large amounts of data and process it for the connected IoT devices. It comprises a set of components such as the cloud-based UIoT Gateway, that handles all requests the devices makes to the middleware, the Data Interface Management System (DIMS), which is the API interface for all data handled by the middleware, and the User Interface Management System (UIMS), that is the user-friendly interface for the middleware data and operations.

UIoT has an authentication mechanism that demands an explicit registration before a new device is able to interact with other devices or the middleware itself. As described in [11], in this self-registration model the device must ask its registration to the middleware, using a REST API. To successfully accomplish the self-registration process, the device must provide its identification data and specify its services. When the middleware validates the given registration data, the device is able to send data to and to consume data from the middleware.

As also described in [11], the UIoT Gateway acts as a semantic gateway, translating communications from the devices to the middleware. This allows UIoT to handle IoT devices that are resource-constrained in terms of power, memory, processing and/or networking so that, for instance, they are not able to communicate using the TCP/IP protocol suite. This feature also allows UIoT to integrate certain IoT devices that are only capable of communicating with hardware and software from the same manufacturer [12].

C. Host-Based IDS

Host based Intrusion Detection Systems are classical IDS types designed to run only within a target host where the HIDS performs tasks of monitoring and analyzing files, memory, input/output and processes, but do not observe aggregated network traffic since the HIDS has access just to traffic addressed to the host in which it runs. The basic function of the HIDS is then to trace and verify information relative to logs, registers, events, file systems, permissions, among others elements that are considered as tagged objects [13] interesting for intrusion detection matters. Thus any file, device or process can be tagged, so that it can be distinguished easily and verified frequently.

Another approach [14] considers that intrusions can be detected by trapping different behavior in processes, system calls, and other events that are considered as anomalies from a normal behaviour. For instance, if an e-mail application client starts to open and write files, create sockets and listen to different ports that are not supposed to pertain to this application, then it is possible that it was compromised.

D. Botnets

Since there are several ways to invade networking computing systems, this fact allows the organization of one of the most successful distributed attack malware and method called botnet or zombie net. A botnet is a set of networking computing devices that are compromised and then controlled by the attacker and explored without their owners knowledge. Once the devices are invaded, malicious scripts and codes can be executed in their operational context, accomplishing multiple routines designed to perform attacks to other designated targets. For instance, in a basic DDoS attack the botnet controller can orchestrate the bots to send continuous TCP, UDP or ICMP requests to a target server in order to overwhelm the listener and stop its correct functioning. Botnets can also be used to collect sensitive data and personal identities or to distract organizations from real attacks. These possibilities can be combined in a typical life cycle of a botnet as shown in Figure 1.

E. Botnets: Vulnerabilities serving as Entry Points

1) Vulnerabilities in conventional systems: Traditional operating systems have a list of known vulnerabilities that are the first to be explored by an attacker, including:

- Known vulnerabilities of default TCP ports, such as 135, 139 or 593;
- Backdoors left behind by a Trojan previously installed;
- Default passwords;
- Configuration files in unsecured locations;
- Database related vulnerabilities;
- Default routes and directories.

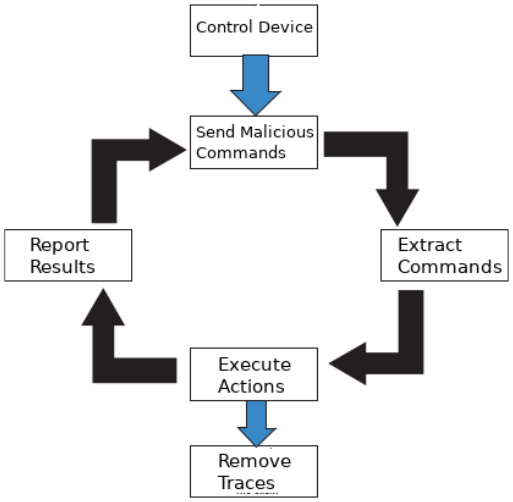


Figure 1. Life cycle of a botnet [15].

2) Vulnerabilities in IoT systems: According to [16] there are known vulnerabilities common to IoT environments, including:

- Authentication bypass and/or inappropriate authorization;
- Denial of Service attacks in IoT;
- Eavesdropping in IoT;
- Node capture in the IoT;
- Physical security of sensors and actuators.

F. HIDS: Prevention Methods

An HIDS works with information collected inside a computational device, allowing the HIDS to perform routine activities to determine which processes and users might be involved in some kind of attack. As discussed in [17] an HIDS is used to check and maintain protected a host system and its network activities, regardless the system has been attacked or not.

Still according to the [17], an HIDS utilize the audit data, incoming traffic, logs produced by the applications, among other data, to detect malicious activities, to prevent intruder‘s activities and to trace the attacks. Most of the existing audit mechanisms are implemented in the host operating system.

G. HIDS: Audit

The gathering and analysis of audit-trails can reveal substantial information about events that might have occurred, such as:

1) Improper changes in the system's configuration files;
2) Connection attempts that are inconsistent or reiterated (such as in brute-force guessing attacks);
3) Creation of illegal application processes, or the removal of the legitimate ones;
4) Sudden scarcity of resources, e.g. of memory and processing time;
5) Abnormal increase of disk usage;
6) Irregular attempts to address connections.

H. HIDS for IoT

A set of HIDS has a fundamental role in the discovery of vulnerabilities in conventional networks, because each HIDS can alert the administrator about possible attacks, helping to understand the nature of the vulnerabilities. There are numerous implementations of HIDS applications for traditional network devices, e.g., OSSEC, Tripwire, AIDE and Prelude Hybrid IDS.

However, according to [18], IoT requires robust IDS that can detect new attacks and that simultaneously imposes a small overhead to the running environment. A set of IoT HIDS must present a form of distributed behavior regarding both the sharing of rules that are created for the network and the reporting to the IoT controller regarding occurrences of detected attacks.

Thus, each computational system that hosts an HIDS in an IoT instance will be part of a “web of trust”, consisting in all host devices that have a distributed HIDS agent. Each HIDS agent in the host operating system will have to refer to a central controller that will provide the distribution of detection rules and coordinate incident control. Each instance of this distributed set of HIDS will have some common features of a conventional HIDS, such as algorithms to audit all the items mentioned in Subsection II-G, and once the detection rules are configured, they will be able to perform actions on events detected in the audit process.

Finally, for hosts with higher processing capacity the HIDS must be specifically configured to detect if the device is being controlled and/or if it is part of a botnet. In this case the audit can focus on specific vulnerabilities of IoT networks, as the HIDS agent will be able to perform response actions on the device, such as closing connections, generating reports and setting actions.

III. Related Work

There are numerous studies on intrusion detection systems in resource constrained devices. However, most are concerned with studying how traditional network intrusion detection systems operate on an embedded device such as a Raspberry Pi.

One interesting work is [19], which investigates how a traditional IDS operates on a Raspberry Pi thus observing the use of CPU and RAM in a resource-constrained device. In this sense, the analysis of an existing intrusion detection tool in a Raspberry Pi is similar to the one proposed in the present paper, since in both cases the central issue is the requirement of greater management resources.

Another relevant study is [20], that attempts to analyze which traditional IDS running on a Raspberry Pi has the best management of resource usage considering the packet capture rate for vulnerability analysis.

The paper [6] reports the growth of botnet networks related to the increasing number of IoT devices that have security holes and are included in botnets such as the cited Mirai, but this work does not address security measures to mitigate such attacks.

Differently from the mentioned works, paper [21] proposes an analysis of intrusion detection methods for IoT in a general context, considering a wide variety of devices used in IoT.

The majority of IDS implementation in IoT is based on adapting existing detection tools to devices usually used in the IoT context and managing the necessary resources. The proposal in the present, however, is based on the creation of a host-based intrusion detection system specific for the IoT context and with a set of rules intended for common IoT vulnerabilities.

IV. Description of the Proposal

As mentioned before, this paper presents a signature-based HIDS for smart devices that will try to connect to the IoT network. As shown in Figure 2, this proposed HIDS comprises well-defined entities that aim to detect attacks and vulnerabilities in smart devices connected to a given IoT instance. The following entities are described hereafter:

1) HIDS Agent for smart devices;
2) Communication API between the HIDS Controller and the HIDS Agent;
3) HIDS Controller;
4) Detection Rules;
5) Event Reports.

A. HIDS Agent

The HIDS Agent is an application that runs on the IoT smart device and performs the tests determined by the rules registered in the local database. The output of these tests is compared with the expected output and, if the output is different from the expected one, then the action foreseen in the rule will be executed and an event report will be generated and sent to the HIDS Controller. To test the rules in the local database, the HIDS Agent uses the functions described below:

1) Threat Scan: This function of the HIDS application reads each of the rules present in the local database and performs a case test for each of them. The case test consists of verifying if the output obtained by executing the rule code is equal to the standard output expected that is set by the HIDS Controller during the registration of the rule. If the output generated by executing the rule code is different from the one in the output field, the action foreseen in the rule will be executed and an event report will be generated and sent to the HIDS Controller by means of the communication API.

2) Enable Self Analysis: Enables the threat scan to be done automatically and periodically.

3) Disable Self Analysis: Disables the threat scan.

4) Update Rules: This function makes a request to update rules to the HIDS Controller through the communication API. Thus, the request is sent from the HIDS Agent to the HIDS Controller which validates the request and responds with the most updated set of rules in the format shown in Figure 3. Once the HIDS Agent instance receives the updated rules, the HIDS application will check whether the rules have been updated, added, or removed from the previous rule set, and will persist the resulting data in the local rule database.

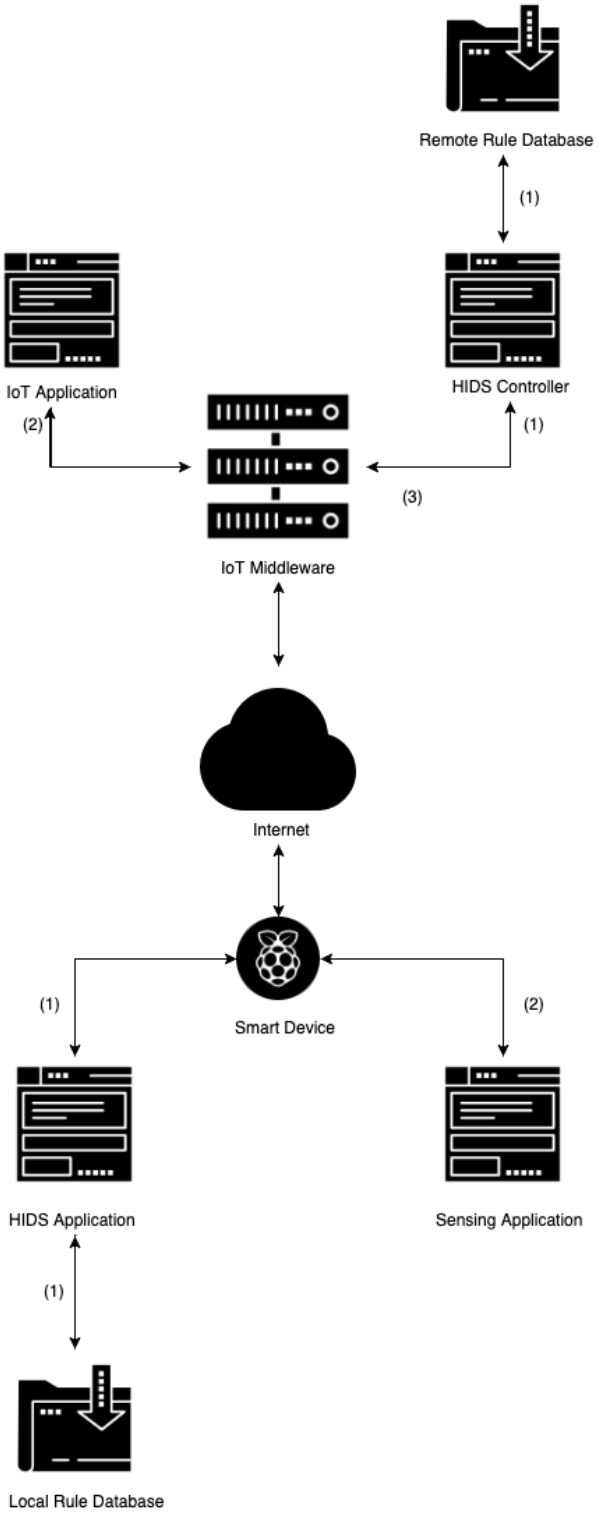


Figure 2. Logical architecture of the proposed IoT HIDS.

5) Set the time between rule updates: Rule updating can be initiated using the command line interface (CLI), but the HIDS application previews that rule update requests are made automatically and periodically. This function allows the administrator to set the time between update requests. If it is not explicitly configured, the default time associated with the instance will be 15 minutes.

6) Show Rules: Shows the existing rules in the local rules database.

7) Show settings: Shows the settings of the HIDS application, such as the period for updating rules, configurations of the communication API, etc.

Table I
LIST OF AVAILABLE COMMANDS

Name	Description	CLI Command
scan_threats	Checks for threats according to rules specified in the rules database.	python manage scan
enable_auto_scan	Enables scanning periodically for threat detection.	python manage enable_auto_scan
disable_auto_scan	Disables scanning periodically for threat detection.	python manage disable_auto_scan
update	Updates periodically the database of threats rules.	python manage update
set_update_rules_period	Changes the time for updating HIDS rules.	python manage set_update_time
show_rules	Shows all available rules.	python manage show_all_rules
show_info	Displays HIDS configuration information.	python manage info

B. Communication API

This component provides communication services for the HIDS Agent in the smart device to interact with the HIDS Controller accessible through the IoT midleware, according to the architecture illustrated in Figure 2. Communications between the mentioned entities occur via the HTTP protocol with the POST method in a standard JSON file, as shown in Figure 3.

```
[
    {
        "name": "os_name",
        "created_at": "2019-01-21 19:04:29.085463",
        "premise": "Linux",
        "action": "print(\"System is not suported.\")\nraise SystemExit",
        "id": 1,
        "test_case": "import platform\noutput =  platform.system()"
    },
    {
        "name": "os_version",
        "created_at": "2019-01-21 19:05:52.516036",
        "premise": "Linux-4.15.0-20-generic-x86_64-with-LinuxMint-19-tara",
        "action": "print(\"System is not suported.\")\nraise SystemExit",
        "id": 2,
        "test_case": "import platform\noutput =  platform.platform()"
    }
]
```

Figure 3. A JSON formatted message to update rules in the HIDS Agent local database.

```
{
    "instance_id"   : "554752",
    "instance_ip"   : "172.16.5.55",
    "instance_mac"  : "ff:ff:ff:ff:ff:ff",
    "rule_id"       : "2",
    "rule_name"     : "dns_attack_rule",
    "output_rule"   : "dns_attack",
    "occur_time"    : "2018-10-20 19:11:41.750209"
}
```

Figure 4. A JSON formatted message to send event reports to the HIDS Controller.

C. HIDS Controller

The HIDS Controller manages a set of HIDS Agents, logging and updating detection rules, defining rules premises, analyzing event reports, and providing the data visualization for the IoT administrator. The HIDS Controller is the entity that communicates directly with the IoT Gateway to define customized assumptions and report the findings. The registration of detection rules in the HIDS controller can be done directly to the application running on the server (Figure 5) or indirectly via the web API. Some of the functions of the HIDS Controller are described in the following subsections.

1) Registration of new rules: Registers the rules in a remote database. This function is responsible for sending the updated rules to all HIDS Agents in smart devices associated with the HIDS Controller.

2) Definition of Assumptions: Defines the default expected output of the rules that is used in the tests performed in the HIDS Agents to verify threats and vulnerabilities.

3) Event Report Analysis: Analyzes the event reports that arrive at the HIDS Controller coming from the HIDS Agents associated with it.

4) Threat Treatment: Based on the event report, performs some action to counter the reported vulnerabilities.

D. Rules

A rule is an information block that assists the HIDS Agent in taking some previously defined actions, an abstraction that has the following 6 main fields:

- Name: Rule name;
- Id: Uniquely identifies a rule;
- Date: Date of creation;
- Assumption: Value taken as true for a given condition or characteristic of the system;
- Test case: Code that finds the characteristic or condition to be tested and returns it for verification;
- Action: In case the value found in the test case is different from the value found in the assumption, the action defined in this field must be taken.

The registered rules are designed to counter certain anomalies commonly found in IoT systems. For the sake of simplifying the application, these rules are classified in different contexts described below:

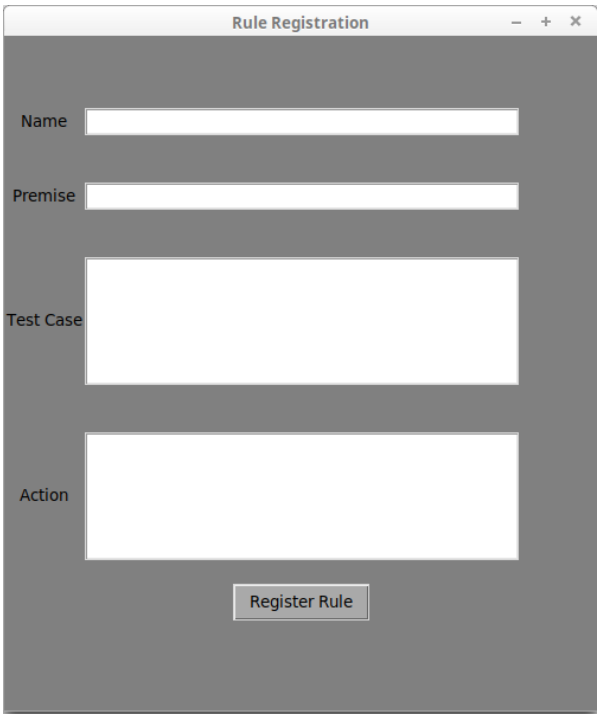


Figure 5. Form for registering rules in the HIDS Controller.

1) Network;
2) Resources;
3) Known vulnerabilities.

1) Network: HIDS Agents communicate with the IoT Gateway through a traditional network infrastructure that is considered as potentially presenting a number of well-known vulnerabilities and attacks. Such vulnerabilities can serve as entry points for a malicious agent to take control of the smart device that hosts the HIDS Agent. Thus, the HIDS rules were developed to detect vulnerabilities and attacks related to this network context, including for instance rules regarding the following items:

1) Port scan detection;
2) DDoS attacks detection;
3) DNS attacks detection.

2) Resources: The use of resources such as processing, memory and disk space is of great importance to smart devices. Thus, these resources can be the target of attacks that aim to disable or reduce these resources for legitimate applications. To detect such attacks, rules for this context have been created, including:

1) Processes with excessive memory use;
2) Processes with excessive processing usage;
3) Processes with excessive usage of HD memory;
4) High temperature.

3) Known vulnerabilities: This context provides for the creation of rules for known vulnerabilities specific to the IoT environment, as well as regarding good practices for configuring smarts devices. Rules implemented in this context include:

1) Default passwords;
2) Standard open ports;
3) Configuration files in standard and unprotected directories.

E. Event Report

An Event Report is a set of information that is sent by an HIDS Agent to the HIDS Controller. This set of information is sent at the moment a vulnerability is detected by the rules run in the HIDS Agent. As illustrated in Figure 4 this information set contains the following item that are discussed hereafter:

1) instance_id: Identifier of the HIDS Agent;
2) instance_ip: IP address of the HIDS Agent;
3) instance_mac: MAC address of the HIDS Agent;
4) rule_id: Identifier of the rule that detected the vulnerability;
5) output_rule: Output found by the rule;
6) occur_time: Timestamp of the moment the vulnerability was found.

V. Life Cycle of an HIDS Agent

For the detection and treatment of vulnerabilities/attacks it is necessary that the HIDS Agent application operate according to some well defined steps as shown in Figure 6. The following sequence of steps transparently define the behavior of the application on the smart device that wants to send data to the HIDS Controller cloud application that will handle the received data:

1) Register the HIDS Agent on the HIDS Controller;
2) Download the HIDS source code;
3) Request updated rules to the HIDS Controller;
4) Validate the HIDS Agent request and respond with the updated rules;
5) Update the local rules database;
6) Run case tests for each local database rule;
7) Perform the actions provided in the rules;
8) Send Event Report;
9) Handle received event reports.

A. Register the HIDS Agent on the HIDS Controller

Once the smart device is to enter the IoT network to send its monitored data to IoT applications in the cloud, a self-registration process is required, as described in Section II. Once this process is completed, the smart device will be registered in the HIDS Controller as an associated smart device.

B. Download HIDS source code

When the process of self-registration of the HIDS Agent in the HIDS Controller is finished, the smart device downloads the HIDS Agent application source code and starts the installation process. From this point on, the smart device runs an instance of the HIDS and can be accepted as a valid device in the IoT network instance.

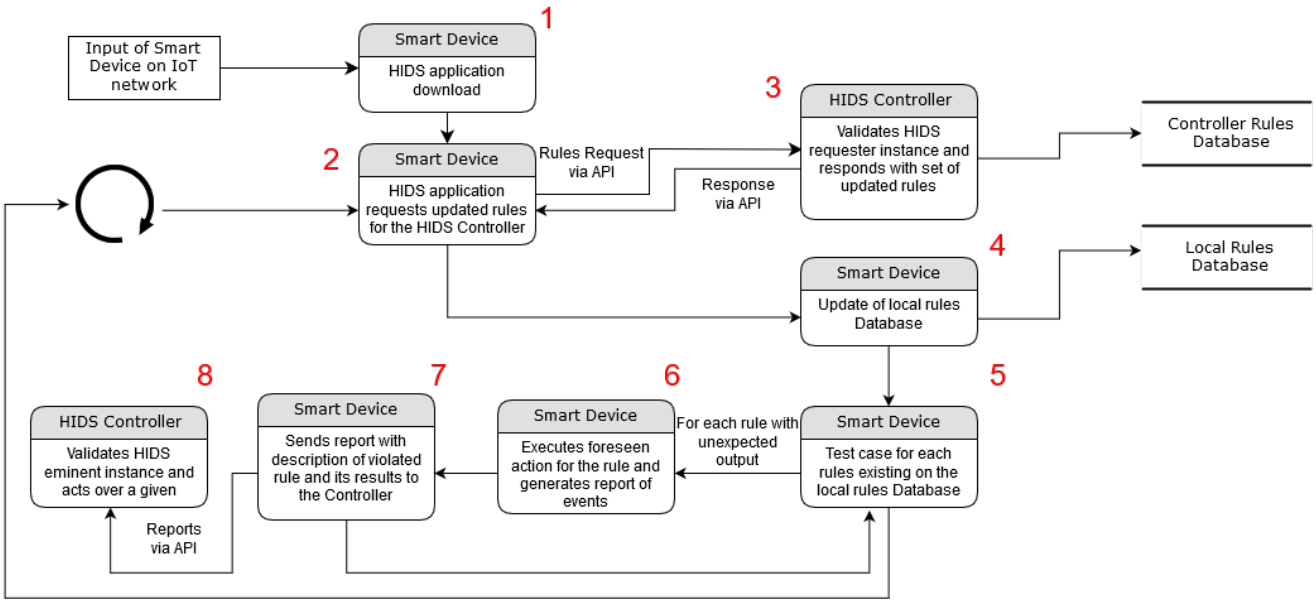


Figure 6. Application life-cycle from the standpoint of the HIDS Agent on a smart device.

C. Request updated rules to the HIDS Controller

In this step, the HIDS Agent uses the communication API to send to the HIDS Controller a request to update detection rules.

D. Validate the HIDS Agent request and respond with the updated rules

The HIDS Controller validates the requesting HIDS Agent based on its identification that was previously registered during the self-registration process. Thus, the instance is validated and then the HIDS Controller responds with the set of updated rules.

E. Update the local rules database

Once the updated rules have been received in the HIDS Agent, the local rules database is updated.

F. Run case tests for each local database rule

9 Once the local rules database is updated, each rule is tested to determine if the output found is different from the output expected for the rule.

G. Perform the actions provided in the rules

For each rule that fails on the test runs, the HIDS Agent will execute the action provided in the rule.

H. Send Event Report

For each rule that fails the test runs, an event report is generated and sent to the associated HIDS Controller.

I. Handle event report received

The HIDS Controller will perform actions to counter the vulnerabilities reported by the HIDS Agent.

VI. Results and Discussion

The validation process for this paper proposal includes the configuration of a controlled laboratory environment to verify the HIDS prototype regarding both the detection of expected attacks and the generation of the event report by the HIDS Agent for the HIDS Controller. Also, the process includes verifying if the HIDS Controller, once notified, will perform an action to counter the detected attacks. Thus, in this controlled environment we submit the prototype to simulated attacks aimed at measuring the number of detected attacks and vulnerabilities discovered, resulting in the rate of detection of threats in the IoT network. As the controlled validation environment shows results that serve just for functional validation of the prototype, the same prototype was submitted to operations in a closed non-controlled experimental IoT, as an initial validation approach before test in an open IoT instance connected to the Internet.

A. Simulated Attacks and Successful Detection

For each rule described in Section IV-D, 150 clients are configured with the proposed HIDS Agent and then all of them were tested regarding a vulnerability specific of the tested context. The following scenarios were executed in the simulated environment as illustrated in Figure 7:

a) Raspberry Pi configured with default user and password: Each of the 150 simulated Raspberry Pi was configured with standard user and password, each of them received an instance of the proposed HIDS, and each of these instances was configured with a standard user verification rule. Once the HIDS Agent instances were started, as expected all of the Raspberry Pi detected this vulnerability and an event

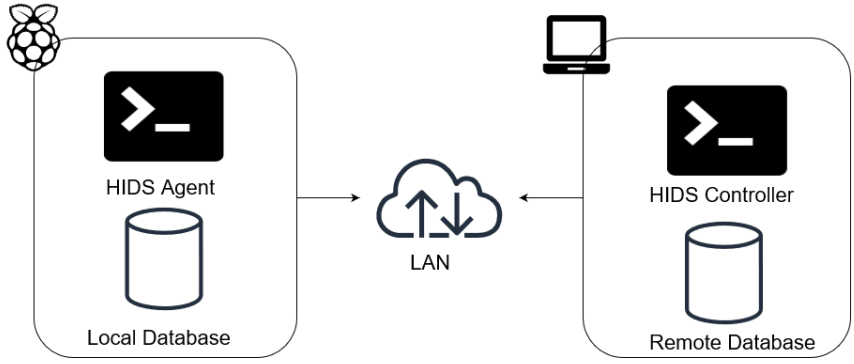


Figure 7. Scenario where the tests were run.

report was sent to the associated HIDS Controller as shown in Table II.

b) Raspberry Pi with open SSH port: Each of the 150 simulated Raspberry Pi was configured with an open SSH port, each of them received an instance of the proposed HIDS, and each of these instances was configured with a rule to detect that the SSH port is open. Once the HIDS instances were started, as expected all of the Raspberry Pi detected this vulnerability and an event report was sent to the associated HIDS Controller as shown in Table II.

c) Raspberry Pi with a malicious program consuming more than 80% of available RAM: For each of the 150 simulated Raspberry Pi devices, processes were created that intentionally consumed more than 80% of the available RAM. Once the HIDS instances were started, as expected all of the Raspberry Pi detected this vulnerability and an event report was sent to the associated HIDS Controller as shown in Table II.

It is important to note that such a high detection rate in the exposed scenarios is explained by the fact that all vulnerabilities tested had previously registered signatures in the HIDS Agent instance, in addition to being devices tested in a controlled environment.

Table II
VALIDATION TESTS OF RULES FOR ATTACKS/VULNERABILITIES, LISTING EXECUTED (EXE.) AND DETECTED (DET.) ONES

Rule	Description	Exe.	Det.	Rate
Default Credentials	An HIDS Agent was initialized on a device with standard credentials.	150	150	100%
SSH Port Open by Default	An HIDS Agent was initialized on a device with SSH port opened.	150	150	100%
Anomalous Process Behavior	An HIDS Agent was initialized on a device that is running a resource-intensive process.	150	150	100%

B. False Positives

As in the controlled validation environment the rules present correct functionality, but the results are very limited to allow a general validation regarding false positive and false negative conclusions by the HIDS Agent, the same tests of Table II were performed in an experimental though closed IoT environment, i.e., a place that allowed attacks when the devices are executing real IoT activities. In this situation, the results for false positives are listed in Table III.

Table III
FALSE POSITIVES RESULTS FOR THE RULES IN A CLOSED IOT ENVIRONMENT, LISTING EXECUTED (EXE.) AND DETECTED (DET.) ATTACKS

Rule	Description	Exe.	Det.	Rate
Default Credentials	An HIDS Agent was initialized on a device with valid credentials.	150	0	0%
SSH Port Open by Default	An HIDS Agent was initialized on a device with SSH port closed.	150	0	0%
Anomalous Process Behavior	An HIDS Agent was initialized on a device that is running standard processes.	150	20	13.3%

C. Designing New Detection Rules for the Proposed HIDS

New rules similar to those described in this paper can be designed specifically to meet the requirements of applications running on the smart device. In a conventional IDS, there is a certain difficulty in creating custom rules such as these. On the other hand, in the proposed HIDS Agent, anyone with a basic knowledge in the Python programming language can create a custom rule and add it to the remote rule table stored by the HIDS Controller to update similar Agent instances associated with the Controller.

VII. CONCLUSION AND FUTURE WORK

The proposed HIDS for IoT devices, developed as a prototype operating in a Raspberry Pi with resource usage management, was submitted to an initial validation process concerning 3 classes of detection rules, i.e., for network events, computing resources, known IoT vulnerabilities. The developed prototype in the controlled validation environment was able to detect all the vulnerabilities it was supposed to detect using the designed rules.

In a lessen controlled validation environment, false positive results were observed related to processes running normal IoT activities. In this case, the false positive anomaly detection rate

was 13.3%, which is high percentage but considered common and somewhat expected in these cases. This detection rule has greater difficulty in deciding whether an action is anomalous or if it represents a case of a pattern previously defined by analyzing the network over a period of time. Given that it is necessary to reduce this false positive rate, a planed evolution of the proposed HIDS comprise using methods such as neural network, deep learning, KNN, among others, in future works to minimize false positives.

Differently from IDS proposals usually implemented in the context of IoT, the proposed HIDS described in this paper can be customized for vulnerabilities specif to some IoT instance or application.

Also, the proposed HIDS is suitable for resource constrained IoT devices because it has been developed and prototyped for devices with restricted resources that are common in IoT networks. This is an interesting characteristic since the proposed distributed intrusion detection system can handle most of the known vulnerabilities in IoT devices and, at the same time, introduces HIDS Agents that run directly on the smart device associated with the IoT network, including the heterogeneous range of IoT devices with limited processing capacity such as simple sensors and actuators.

Another interesting future work consists in developing new scenarios in which the HIDS Agent runs on an independent specialized device to detect vulnerabilities and subsequently replicate the rules to devices with hardware limitations. This work involves a significant research regarding functional and semantic inter-operation with those devices.

Intrusion detection systems need systematic permanent approaches to update and validate their rules and internal algorithms, particularly considering new attacks in real environments. As described in the validation process for this paper proposal, the proposed distributed intrusion detection system must systematically be submitted to real IoT networks events, as requirement to be considered a realistic solution.

Acknowledgements

The authors gratefully acknowledge the support from the Brazilian Research Councils CNPq (Grant 465741/2014-2 INCT on Cybersecurity), CAPES (Grant 23038.007604/2014-69 FORTE), FAP-DF (Grants 0193.001366/2016 UIoT, 0193.001365/2016 SSDDC, and Call 01/2019), as well as the LATITUDE/UnB Laboratory (Grant 23106.099441/2016-43 SDN), the Ministry of the Economy (Grants 005/2016 DIPLA, 011/2016 SEST and 083/2016 ENAP), and the Institutional Security Office of the Presidency of the Republic of Brazil (Grant 002/2017).

References

[1] K. Ashton. That 'Internet of Things' Thing. Accessed: Apr 05, 2019. [Online]. Available: https://www.rfidjournal.com/articles/view?4986

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] A. L. Albertin and R. M. Albertin, "A internet das coisas irá muito além as coisas," *GV-executivo*, vol. 16, no. 2, pp. 12–17, 2017.

[4] "Annual CyberSecurity Cisco," Cisco 2018, Tech. Rep., 2018. [Online]. Available: https://www.cisco.com/c/dam/m/digital/elq-cmcglobal/witb/acr2018/acr2018final.pdf

[5] J. J. Costa Gondim, R. De Oliveira Albuquerque, A. Clayton Alves Nascimento, L. J. García Villalba, and T.-H. Kim, "A methodological approach for assessing amplified reflection distributed denial of service on the internet of things," *Sensors*, vol. 16, no. 11, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/11/1855

[6] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[7] H. G. C. Ferreira and R. T. de Sousa Júnior, "Security analysis of a proposed internet of things middleware," *Cluster Computing*, vol. 20, no. 1, pp. 651–660, Mar 2017. [Online]. Available: https://doi.org/10.1007/s10586-017-0729-3

[8] T. H. Lee, C. H. Wen, L. H. Chang, H. S. Chiang, and H. M. C., "A Lightweight Intrusion Detection Scheme Based on Energy Consumption Analysis in 6LowPAN," in *Advanced Technologies, Embedded and Multimedia for Human-centric Computing. Lecture Notes in Electrical Engineering*, vol. 260, 2014.

[9] H. G. C. Ferreira, E. Dias Canedo, and R. T. de Sousa Júnior, "IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct 2013, pp. 53–60. [Online]. Available: https://doi.org/10.1109/WiMOB.2013.6673340

[10] H. G. C. Ferreira, E. D. Canedo, and R. T. de Sousa Júnior, "A ubiquitous communication architecture integrating transparent UPnP and REST APIs," *International Journal of Embedded Systems*, vol. 6, no. 2-3, pp. 188–197, 2014. [Online]. Available: https://doi.org/10.1504/IJES.2014.063816

[11] F. L. d. Caldas Filho, L. M. C. e. Martins, I. P. Araújo, F. L. L. d. Mendonça, J. P. C. L. da Costa, and R. T. de Sousa Júnior, "Design and Evaluation of a Semantic Gateway Prototype for IoT Networks," in *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, ser. UCC '17 Companion. Austin, Texas, USA: ACM, Dec 2017, pp. 195–201.

[12] C. F. C. Ribeiro, F. L. d. Caldas, L. M. C. e Martins, C. J. B. Abbas, and R. T. de Sousa Júnior, "Protocolos de Redundância de Gateway Aplicados em Redes IoT," in *Anais do XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2018)*, Campina Grande, PB, Brazil, sep 2018, pp. 1065–1069.

[13] S. R. Snapp, J. Brentano, G. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, and K. N. Levitt, "DIDS (distributed intrusion detection system)-motivation, architecture, and an early prototype," 2017.

[14] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 255–264.

[15] C. A. Schiller, J. Binkley, D. Harley, G. Evron, T. Bradley, C. Willems, and M. Cross, *Botnet: the killer web app*. Syngress Publishing, 2007.

[16] A. T. Ebraheim Alsaadi, "Internet of Things: Features, Challenges, and Vulnerabilities," *International Journal of Advanced Computer Science and Information Technology (IJACSIT)*, vol. 4, no. 1, pp. 1–13, 2015.

[17] K. Letou, D. Devi, and Y. Jayanta, "Host-based Intrusion Detection and Prevention System (HIDPS)," *International Journal of Computer Applications*, vol. 69, pp. 30–31, 05 2013.

[18] M. F. Elrawy, A. I. Awad, and H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey," *Journal of Cloud Computing*, vol. 7, no. 1, p. 21, Dec 2018.

[19] A. Sforzin, F. G. Mármol, M. Conti, and J. Bohli, "RPiDS: Raspberry Pi IDS — A Fruitful Intrusion Detection System for IoT," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, July 2016, pp. 440–448.

[20] A. Aspernäs and T. Simonsson, "IDS on Raspberry Pi: A performance evaluation," Bachelor's Thesis, Linnaeus University, 2015.

[21] O. Anthony, J. Odeyabinya, and S. Emmanuel, "Intrusion detection in internet of things (iot)," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 1, pp. 504–509, Feb 2018.

Metodología para la detección de Botnets en la nube mediante técnicas de optimización por medio Grid-Search

David Gonzalez-Cuautle[†], Gabriel Sanchez-Perez[†],
Aldo Hernandez-Suarez[†] y Ana Sandoval-Orozco[‡]∗
[†]Instituto Politécnico Nacional, Sección de Estudios de Posgrado e Investigación
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Culhuacán
Av. Santa Anna 1000, 04260 Ciudad de México, México
Emails: {dgonzalezc1701, ahernandezs1325}@alumno.ipn.mx, {gasanchezp, ltoscano}@ipn.mx

[‡]Departament of Electrical Engineering Faculty of Technology University of Brasilia (UnB)
Campus Universitario Darcy Ribeiro Brasilia CEP 70910-900, Brazil
Email: asandoval@redes.ubn.br

***Resumen*—En los recientes años las botnets se han convertido en una de las amenazas más serias para todo aquello que se encuentra en la nube debido a la gran implementación de servicios desplegados e información que circula en la misma. La dependencia de las infraestructuras y redes virtualizadas en la nube introduce una gran cantidad de riesgos y vulnerabilidades tales como la denegación de Servicio (DoS), correo no deseado, phishing, filtración de información y la detección apropiada de anomalías. Una solución para la detección rápida y eficaz de botnets es el análisis del flujo red por medio de aprendizaje automático para diferenciar entre tráfico de red benigno y malicioso. En este trabajo, se propone una metodología para comparar diferentes conjuntos de datos y mostrar a su vez el rendimiento de los algoritmos de aprendizaje supervisado más utilizados en el estado del arte y su optimización mediante hiper-parámetros con Grid-Search.**

***Index Terms*—Botnets, Nube, Denegación del Servicio, Phishing, Aprendizaje Automático, Aprendizaje Supervisado, Optimización, Hiper-parámetros, Grid-Search**

Tipo de contribución: *Investigación Original*

I. Introducción

La nube ofrece muchas características beńeficas tanto para particulares como para comunidades tales como la transparencia y la elasticidad del servicio, pero todo ello introduce una serie de vulnerabilidades que son resultado subyacente de su naturaleza al ser un sistema virtualizado y directamente involucrado con la Internet.

Hoy en día muchos ataques pueden ser fácilmente ocultados dentro de un tráfico masivo HTTP, a pesar de contar con dispositivos físicos y lógicos en grandes infraestructuras, muchas de ellas solo toman en cuenta la identificación de botnets por medio de soluciones basadas en firmas o comportamientos heurísticos que requieren de actualizaciones constantes de sus repositorios o bases de datos correspondientemente; lo cual es un contratiempo en proporción a lo rápido que se desarrollan y propagan las nuevas amenazas dentro la nube.

Una botnet es una red de computadoras comprometidas en Internet, la cual corre un programa malicioso llamado Bot o Agente [1]. Esta red de computadoras son controladas vía remota por una Botmaster, es decir, la encargada de controlar toda la infraestructura a través de un servidor de Comando y Control (C&C).

El impacto de las redes de botnets es tan significante en el sector público o privado debido a la variedad de ataques que se realizan a diferentes infraestructuras a través de la nube. Los ataques de botnets bien identificados incluyen Denegación de Servicio Distribuida (DDoS, por sus siglas en inlglés), correo no deseado, suplantación de identidad, robo de identidad y filtraciones de información [2].

Existen dos formas de clasificar a las arquitecturas de C&C, centralizada o descentralizada.

- **Centralizada**: los bots se comunican con uno o un pequeño número de servidores de C&C que, a su vez, están controlados por la Botmaster.
- **Descentralizada**: los bots son recibidos por al menos un dispositivo y se extienden a otros dispositivos por medio de una red punto a punto.

En su defecto, la mayoría de los botnets dependen en gran medida de un procedimiento de comunicación entre sus huéspedes y su servidor de C&C donde es almacenada toda la información proveniente de los protocolos de Transferencia de Hipertexto (HTTP, por sus siglas en inglés) o IRC (Internet Relay Chat).

Las botnets más modernas generalmente imitan el tráfico de red generado por las aplicaciones normales para evadir la detección de agentes de seguridad tales como Honeynets, sistemas de detección de intrusos en red (NIDS), sistemas de prevención de intrusos (IPS) y sistema de detección de intrusos (IDS). Por esta razón, la detección de una botnet tiene a convertirse en un tema de investigación importante.

Una botnet tiene un ciclo de vida que consta de tres etapas:

- **Infección**: Las botmasters infectan otras computadoras a través de phishing o ingeniería social, etc. Cada dispositivo comprometido es un nuevo integrante de esta red maliciosa, es decir un bot. Generalmente, las víctimas desconocen que una descarga contenga malware desde un servidor binario ya sea cuando abren un correo electrónico o navegan por algunos sitios web poco fiables.

- **Comunicación con el C&C**: Las botmasters usualmente usan su servidor de C&C para actualizar el código de malware y propagarse hacia otros infraestructuras dentro de la nube. Los bots también se conectan al servidor de C&C periódicamente para informar sus estados mediante un proceso de escaneo eficazmente coordinado y en gran escala.
- **Ataque**:Los bots controlados por la botmaster lanzan diversas actividades maliciosas de acuerdo con las instrucciones recibidas o buscan otras computadoras víctimas para seguir expandiéndose y así lograr un estado de resiliencia más eficiente, empleando esquemas donde se evite su detección a través de un ISP.

Para mitigar la amenaza a la seguridad que representan las botnets, una solución alterna para superar tales limitaciones es entrenar los algoritmos de Aprendizaje Automático (AA) y generar modelos robustos para identificar con precisión los flujos de red de las botnets con los conjuntos de datos proporcionados por [3]- [5].

Un elemento fundamental para que una botnet funcione y logre llevar a cabo su principal tarea, es el procedimiento de escaneo para la detección de huéspedes que genera un volumen excesivo de flujos red en comparación con los de una infraestructura normal de la nube, es por ello que al analizarlos con AA y caracterizando cada una de las entradas y cada una de las salidas de los mismos, se podrá identificar de manera proactiva la propagación temprana de una botnet.

Con ayuda del Aprendizaje Supervisado (AS), se pueden construir muestras etiquetadas para resolver problemas de clasificación relacionados con la detección de botnets. Es decir, con las muestras de los conjunto de datos ya distinguidas entre tráfico benigno o malicioso, puede discernirse por medio del comportamiento generado en ambos casos si en la infraestructura de la nube existe o no una botnet.

Las muestras se someten a pasos de pre-procesamiento en función de las características intrínsecas, como los valores categóricos, la dimensión de las características y el equilibrio de clases; luego se introducen en el algoritmo de AS elegido para capacitarse y obtener un modelo de clasificador. En consecuencia, el modelo clasifica los rastros entrantes de la red como benignos o maliciosos, con cierto grado de certeza.

En la literatura se han propuesto muchos métodos de detección durante la última década, sin embargo, las metodologías propuestas no están definidas apropiadamente y por tanto hacen que analizar todos los flujos de red que generen conexiones excesivas y anormales sea una tarea abrumadora.

Teniendo en cuenta lo que los autores sugirieron en [6], no existe un algoritmo que ofrezca el mejor rendimiento en todos los casos, en particular para la detección de botnets. En este trabajo, se propone un *portfolio de algoritmos* para evaluar adecuadamente varios algoritmos de AS y encontrar el más adecuado evaluando su desempeño con muestras de la distintas botnets. Debido a la naturaleza de los algoritmos de AS, los parámetros adicionales, también denominados hipermarámetros, son necesarios para mejorar el rendimiento de los modelos de clasificador resultantes. Este manuscrito presenta una técnica de sintonización exhaustiva conocida como Grid-Search (GS) [7].

El objetivo principal es buscar en profundidad los mejores hiper-parámetros que se aprenden de los datos de entrenamiento y aquellos que se optimizan por separado. Como resultado, se elige el algoritmo óptimo para detectar efectivamente el tráfico malicioso de las botnets dentro de la nube.

El resto del artículo está organizado de la siguiente manera. Sección II proporciona una visión general de los trabajos relacionados con la detección botnets y los algoritmos empleados. La Sección III describe la metodología propuesta. La Sección IV detalla los resultados experimentales obtenidos del *Portfolio de algoritmos* que emplean técnicas de Grid Search. Finalmente, La Sección V concluye este trabajo.

II. Trabajos relacionados

En [8] es demostrada la importancia que juega el protocolo HTTP para la propagación de ataques de Denegación de Servicios Distribuidas (DDoS) ya que su mayoría los servicios de la nube son a través de éste, lo cual trae consigo las vulnerabilidades y errores inherentes del protocolo así que se adapta el conjunto de datos *CIDDS-001 (Coburg Intrusion Detection Dataset)* donde el tráfico entrante está basado en el tiempo para estimar la entropía de un conjunto de características en el encabezado de flujo de red. Este conjunto de datos contiene tráfico benigno como de ataques, el cual permite realizar una evaluación comparativa de los sistemas de detección de intrusos de red en la nube. El sistema de detección propuesto para este artículo consiste en tres pasos generales: la estimación de la entropía, el pre-procesamiento y la clasificación del tráfico. Los experimentos mostraron que al aplicar el algoritmo de árboles aleatorios, alcanzó una detección de ataques del 97 % precisión, seguido de Naive-Bayes, vecinos más cercanos, árboles de decisión y perceptrón de multicapa con un 94 %, 86 %, 73 % y 28 % respectivamente.

La evolución de las botnets de un arquitectura centralizada a una descentralizada y su detección por medio de una red definida por software (SDN) son planteadas en [9]. La metodología en este trabajo da un enfoque relacionado con el AA en comparación con las técnicas tradicionales en la administración de una red. Emplear un red definida por software le permite detectar y categorizar tráfico de redes punto a punto (P2P). Un flujo benigno es generalmente descrito como paquetes con la misma fuente y destino en un periodo de tiempo en específico. Un punto crucial es la detección de patrones de tráfico anormales dentro de una red, donde resalta el tamaño del flujo, su duración y el tamaño medio del paquete. Aquí detallan qué características del protocolo TCP/UDP son utilizadas para aplicar los algoritmos de AA, tales como el conteo de paquetes, volumen total, duración del flujo, total de bytes para los cabezales e intervalo de llegadas de los paquetes. Un módulo auxiliar de clasificación de flujo benigno o malicioso es creado y probado con aplicaciones que benignas como eMule, uTorrent y Skype mientras que del lado de las maliciosas se utilizaron las botnets Zeus y Storm. Los algoritmos de AS empleados arrojaron que la máquina de soporte vectorial tuvo un 97.55 % de precisión, mientras que vecinos más cercanos obtuvo un 91.66 % en la detección de flujo malicioso.

Un estudio de la taxonomía de las botnets [10] mostró que sus bots necesitan sincronizarse con su botmaster para recibir comandos o mostrar la información que han obtenido hasta

ese momento, esta sincronización es programada y genera patrones anormales para un usuario estándar en la nube. En este artículo de acuerdo al protocolo de comunicación utilizado las botnets pueden clasificarse en tres categorías botnets basadas en IRC (Internet Relay Chat, por sus siglas en inglés), Mensajes Instantáneos (IM) y las web. Incluso es usada una métrica llamada Tasa de Repetición de Huésped Total (TRH), es decir, es un fenómeno de contacto repetido a largo plazo y muy similar al patrón de acceso similar de los clientes de un servidor web, sin embargo, la diferencia radica en que la carga útil del paquete es muy pequeña y con instrucciones muy simples. Con lo anterior, se remarca que el tráfico más empleado hoy en día es el proveniente del protocolo HTTP; esto hace que muchas redes de corta fuegos y proxies permitir a sus huéspedes tener acceso a Internet. La evaluación de este tipo de botnets se hace con seis distintas, incluyendo sus variantes. Arboles de decisión, perceptrón multicapa son los algoritmos de AS empleados y dando como resultado precisiones del 99 % para la botnet Zeus, Spyeye 98 %, Weasel 82 %, SJTU 26.3 % y finalmente QingPu con un 13.6 % en su detección.

En [11] , para que la nube cuente con resiliencia es necesario observar y analizar el comportamiento de la red donde se planea establecer, además de tomar medidas correctivas en caso de cualquier anomalía detectada. Además, la detección de malware es de vital importancia para hacer una observación detallada de las propiedades especificas de un sistema, es decir, el análisis de tráfico de red debe de hacerse cuidadosamente en cada nodo individual para identificar las características más representativas que pueden incorporarse dentro de un esquema de detección unificado que cubra tanto al sistema como a la red. La botnet Kelihos fue el malware utilizado para probar los conceptos ya mencionados.

Para cada uno de los nodos en la red existe un hiper-visor dedicado a la Administración de la resiliencia en la Nube (CRM, por sus siglas en inglés) compuesto por:

- Motor de análisis de red
- Motor de análisis de sistema
- Sistema de resiliencia del motor
- Motor de coordinación y organización

Las características como el puerto origen, el puerto destino, número de bytes, número de paquetes, media, desviación estándar de paquete, llegada del paquete, duración del flujo y suma de los primeros paquetes son las usadas en este trabajo, demostrando la detección efectiva de flujos de red anómalos en la nube.

Una fuente importante de actividad maliciosa a través de una red son generadas por las botnets [18] . Aquí se demuestra que de las botmasters dependen en gran medida de un procedimiento de escaneo para detectar huéspedes vulnerables y establecerse a través de un servidor de comando y control (C&C). La propuesta en este trabajo permite que la observación de los flujos anormales generados por una botnet bajo el esquema propuesto pueda ayudar de manera eficiente a los expertos en seguridad de redes en realizar un perfilado adecuado y una identificación oportuna y temprana de la amenaza. Zeus y Mariposa son las empeladas para realizar las pruebas.

La dos herramientas principales utilizadas en el trabajo son Wireshark [1] y NMAP [2] cuya obtención de características del tráfico empieza por la fase de inspección profunda de paquetes (DPI, por su siglas en inglés) en los rastreos de paquetes capturados y detección de cualquier firma de carga útil pasando por 6 filtros personalizados de Wireshark para la distinción de ambas botnets, resultando bajo la métrica condicional derivada de la distribución de 8 características de flujo seleccionadas. Éstas estadísticas en bruto por flujo que forman la base del trabajo son las siguientes:

- Puerto Origen
- Puerto Destino
- Conteo de paquetes
- Conteo de bytes
- Duración del flujo
- Media del tamaño del paquete
- Media del tiempo de llegada de cada paquete
- Tamaños de los primeros 10 paquetes para cada flujo unidireccional

La metodología para este trabajo, reside en el cálculo de la entropía condicional entre los vectores de características ya mencionadas para cada flujo de exploración unidireccional consecutivo. Demonstraron que se exhibe un comportamiento mucho mayor consistencia en sus actividades de escaneo entre los bytes y paquetes con respecto a los puertos IP de origen y destino por cada flujo, ya que existen patrones claros con respecto a la distribución de volumen saliente y entrante (es decir, conteos de bytes y paquetes) de sus flujos de escaneo asociados en IP específica puertos origen y destino.

Realizar el monitoreo de tráfico pasivo basado en la tecnología honeypot para luego analizar los registros de ataque de red y determinar los intrusos, todo lo anterior permite la construcción de un clasificador de tráfico que determinará si es malicioso y benigno del conjunto de datos recopilados es abordado en [19] .

Las características estadísticas del tráfico de red son de suma importancia para las clasificaciones basadas en AA, como la desviación estándar mínima, media y máxima de los paquetes. El análisis de seguridad en este trabajo es definido como:

- Encontrar, descubrir y utilizar técnicas para analizar datos de seguridad.
- Las herramientas de recopilación de datos de seguridad continúan mejorando y la calidad de los datos aumenta exponencialmente.
- Necesidad de aplicar herramientas y técnicas para analizar los datos de seguridad.

Los 5 objetivos principales de esta investigación son:

- Mecanismo automatizado de captura y registro de datos en red.
- Procesamiento de datos y extracción de características.
- Desarrollo del motor de pre-procesamiento de datos para extraer características de datos relevantes y de ataque.
- Análisis de datos y clasificación basada en las herramientas “R” y weka.
- Medición del desempeño y discusiones de resultados.

[1] Wireshark Traffic Analyzer: https://www.wireshark.org/
[2] Nmap Security Scanner: https://nmap.org/

Es planteada una metodología de 7 pasos basada en el protocolo TCP para la extracción y clasificación del tráfico, descrita en la siguiente lista:

- **Captura de datos (TCPDUMP):**Por medio de la herramienta TCPDUMP es obtenida la información de los registros de red en tiempo real en achivos PCAP.
- **Fusión de datos (Mergecap):** Los archivos PCAP son fusionados en un solo archivo para ser pre-procesado de una manera más eficiente.
- **Extracción automatizada de características (Tshark):** Con la herramienta Tshark todas las estadísticas de las conversaciones son obtenidas.
- **Formato de la información:** Las conversaciones obtenidas son guardadas en archivos con los formatos 'ARFF' y 'CSV'.
- **Pre-procesamiento de la información:** Es empleada la herramienta WEKA para la selección de características ya que algunas no aportan relevancia y por tanto se puede optimizar al cojunto de datos.
- **Clasificador de tráfico:** Después del pre-procesamiento de los datos, un solo archivo se ha generado para la construcción del modelo de clasificación. Este archivo contiene los datos que serán utilizados para entrenar y probar el clasificador.
- **Evaluación del desempeño:** Para probar el desempeño del clasificador, las métricas de Precisión, tasa de error y matriz de confusión son empleadas.

El algoritmo clasificador utilizado en este trabajo es Näive-Bayes arrojando resultados de un 66.11 % de precisión en la distinción de instancias.

III. Metodología Propuesta

En esta sección es descrita la propuesta de la metodología, junto con el flujo del trabajo que se muestra en la Fig. 1.

Existen dos fases importantes para que esta metodología sea eficiente: El **entrenamientos** y la **prueba**; con ambas entradas se podrá hacer una **predicción eficiente** de los datos.

La fase de entrenamiento se desglosa de la siguiente manera:

- **Conjunto de datos de entrenamiento:** Para poder clasificar el tráfico benigno del malicioso, es necesario contar una base de un conjunto de entrenamiento bien definido.
- **Extracción y selección de características:** La importancia de este paso es ver las características que aporten la suficiente relevancia para la clasificación de botnets, ya que de ello dependerá la existencia de un sobre-ajuste o bajo-ajuste de los datos al momento de entrenar los datos.
- **Portfolio de algoritmos:** Los algoritmos de AS más utilizados en el estado de la técnica son empleados para generar un modelo predictivo lo suficientemente robusto.
- **Grid-Search:** Permite al portfolio de algoritmos ajustar sus parámetros y optimizarlos para encontrar la combinación ideal que proporcione la mayor precisión.
- **Modelos clasificadores:** Tras entrenar los datos (detectar los patrones en los datos) y siendo optimizados por Grid-Search, se crea un modelo que servirá para hacer las predicciones con base a los algoritmos del portfolio.

Figura 1. Flujo de trabajo de la metodología propuesta.

Mientras que en la fase de prueba, los pasos son los siguientes:

- **Conjunto de datos de prueba:** Para poder probar los modelos clasificadores es necesario compararlos con datos de la misma estructura y así definir su eficiencia de predicción.
- **Extracción y selección de características:** Las características al igual que en conjunto de entrenamiento deben de ser las que aporten la suficiente relevancia para la clasificación de botnets.
- **Modelos clasificadores:** Para generar una predicción, el modelo clasificador requiere de dos entradas, el conjunto de entrenamiento (con los patrones ya detectados de los datos) y contra qué compararlos (conjunto de datos de prueba).

Detalladamente, los conjuntos de datos son proporcionados por la Universidad de Coburg en Alemania [3] y el Instituto Canadiense para la Ciber-Seguridad [4].

A continuación, se realiza una tarea de extracción y selección de características utilizadas en [8] y [10]. Mediante métodos de reducción dimensional [12] . Por lo tanto, el portfolio está diseñado para entrenar diferentes algoritmos de AS y sintonizar diversos hiper-parámetros sobre la marcha, utilizando técnicas de Grid-Search. Finalmente en el paso de prueba, el rendimiento de Los modelos de clasificación resultantes se miden en términos de Precisión, Sensitividad y Exhaustividad.

III-A. Descripción de los conjuntos de datos

Los conjuntos de datos para desarrollar la metodología son nombrados como *CIDDS-001* y *ISCX-Bot-2014*.

- *Coburg Intrusion Detection Data Sets (CIDDS-001):* En la Tabla I se muestra a detalle las características del conjunto de datos empleadas para este trabajo. Al realizar el pre-procesamiento de datos de los archivos CSV, 105530 muestras de flujo malicioso fueron obtenidas y 49554 de flujo benigno.
- *ISCX-Bot-2014:* Dentro de este conjunto de datos las características utilizadas son las mostradas en la Tabla II donde se que incluyeron 16 tipos de botnets en la Tabla

[3]CIDDS data sets: https://www.hs-coburg.de/fileadmin/hscoburg/WISENT-CIDDS-001.zip/

[4]Canadian Institute for Cybersecurity: https://iscxdownloads.cs.unb.ca/iscxdownloads/ISCX-Bot-2014/\#ISCX-Bot-2014/

III se describe a detalle las botnets utilizadas. El tamaño del archivo de captura de red (PCAP) es de 8,5 GB, de los cuales el 44,97 % corresponde a flujos maliciosos.

Tabla I
CARACTERÍSTICAS UTILIZADAS EN EL CONJUNTO DE DATOS *CIDDS - Coburg Intrusion Detection Data Sets (CIDDS-001)*

	Característica	Descripción
1	Duration _ip	Duración del paquete
2	Src Pt_port	Puerto de salida
3	Dst Pt_ip	Puerto destino
4	Packets_port	Número de paquetes enviados
5	Bytes_packets	Número de bytes enviados

Tabla II
CARACTERÍSTICAS UTILIZADAS EN EL CONJUNTO DE DATOS *ISCX-Bot-2014*

	Característica	Descripción
1	Src_ip	Dirección IP origen
2	Src_port	Puerto Origen
3	Dst_ip	Dirección IP destino
4	Dst_port	Puerto destino
5	Out_packets	Número de paquetes de salida
6	Out_bytes	Número de byte de salida
7	Income_packets	Número de paquetes de entrada
8	Income_bytes	Número de bytes de entrada
9	Total_packets	Número total de paquetes transmitidos
10	Total_bytes	Número total de bytes transmitidos
11	Duration	Duración del flujo

Tabla III
DISTRIBUCIÓN DE LAS DIFERENTES BOTNETS UTILIZADAS EN EL CONJUNTO DE DATOS *ISCX-Bot-2014*

	Nombre Botnet	Tipo	Porción de flujo
1	Neris	IRC	25967 (5.67 %)
2	Rbot	IRC	83 (0.018 %)
3	Menti	IRC	2878(0.62 %)
4	Sogou	HTTP	89 (0.019 %)
5	Murlo	IRC	4881 (1.06 %)
6	Virut	HTTP	58576 (12.80 %)
7	NSIS	P2P	757 (0.165 %)
8	Zeus	P2P	502 (0.109 %)
9	SMTP Spam	P2P	21633 (4.72 %)
10	UDP Storm	P2P	44062 (9.63 %)
11	Tbot	IRC	1296 (0.283 %)
12	Zero Access	P2P	1011 (0.221 %)
13	Weasel	P2P	42313 (9.25 %)
14	Smoke Bot	P2P	78 (0.017 %)
15	Zeus Control (C&C)	P2P	31 (0.006 %)
16	ISCX IRC bot	P2P	1816 (0.387 %)

A cada conjunto de muestras se le denomina como X. Y por cada muestra de red en ambos casos, se le es asignada la etiqueta $y \in \{0, 1\}$ que corresponden a flujo benigno o malicioso, respectivamente.

En la realidad, los conjunto de datos provenientes de capturas de red son totalmente desbalanceados, esto provoca modelos de clasificación insatisfactorios donde la clase predominante generaría un sobre ajuste en los resultados finales. Para resolver este problema de es empleado una librería de Python llamada SMOTE [5].

SMOTE es un método de sobremuestreo. Lo que hace es crear muestras sintéticas (no duplicadas) de la clase minoritaria. De ahí que la clase minoritaria sea igual a la clase mayoritaria. SMOTE hace esto seleccionando registros similares y modificando ese registro una columna a la vez por una cantidad aleatoria dentro de la diferencia con los registros vecinos.

III-B. Extracción y selección de características

Es crucial en el AA elegir un conjunto de características apropiado, que mejore significativamente el rendimiento de la clasificación y a su vez el costo computacional sea menor. Es por ello que el análisis de componentes principales (PCA) ha mostrado resultados eficientes en las tareas de extracción y selección de características en conjuntos de datos con altos niveles dimensionales.

PCA es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos hallando las causas de variabilidad del conjunto y ordenándolos por importancia.

El primer componente principal Y^* se define como la combinación lineal de las variables originales que tiene *varianza máxima*. Los valores en este primer componente se representarán como en la Ecuación 1 :

$$Y^* = O_{a1}, \tag{1}$$

donde O es la matriz de observaciones que tiene media cero y por lo tanto también Y^*.

$$\frac{1}{n} Y^{*\prime} Y^* = \frac{1}{n} a\prime_1 O\prime O_{a1} = a\prime_1 S_{a1}, \tag{2}$$

donde S es la matriz de varianzas y covarianzas de las observaciones. E imponiendo la restricción $a\prime_1 a_1 = 1$ y mediante el multiplicador de Lagrange:

$$M = a\prime_1 S_{a1} - \lambda(a\prime_1 a_1 - 1), \tag{3}$$

maximizar la expresión supone derivar respecto a a_1 e igual a cero.

$$\frac{\partial M}{\partial a_1} = 2Sa_1 - 2\lambda a_1 = 0, \tag{4}$$

resultando ser $Sa_1 = \lambda a_1$ donde a_1 es un eigen-vector de la matriz S y λ su correspondiente eigen-valor.

Con las Ecuaciones 1-4 es posible crear un nuevo subespacio y por tanto obtener la mayor variabilidad de los datos con una cantidad menor de dimensiones.

III-C. Diseño de portfolio de algoritmos

El portfolio de algoritmos son conjuntos predefinidos diseñados para evaluar el mejor algoritmo en una instancia dada en la optimización de un problema.

La solución a dicho problema de selección de los algoritmos se realiza de la siguiente manera: *debe existir un conjunto de algoritmos A con un problema de clasificación p. El procedimiento de selección S tiene que determinar qué*

[5]SMOTE: https://imbalanced-learn.readthedocs.io/en/stable /generated/imblearn.over_sampling.SMOTE.html#imblearn-over-sampling-smote

algoritmo es el mejor dado un conjunto de características C en el mismo entorno.

La propuesta es la siguiente:

- Separar los pasos de aprendizaje de los momentos de ejecución.
- Maximizar las características importantes e hiper-parámetros para cada algoritmo.
- Ejecutar los diferentes algoritmos de forma aislada en el mismo ambiente.
- Ejecutar los algoritmos en paralelo dentro del mismo ambiente

Para cumplir con las propuestas descritas, es necesario que los algoritmos más utilizados en el estado del arte relacionados con la detección de botnets sean incluidos en *el portfolio*: Regresión logística (LR) [13], Árboles de Decisión (DT) [14], Bosques Aleatorios (RF) [15], Näıve Bayes (NB) [16] y Análisis Discriminante Lineal (LDA) [17] .

III-D. Grid Search

Grid Search o la búsqueda de hiper-parámetros es una tarea del AA con la cual un algoritmo $a \in A$ es parametrizado por un conjunto de hiper-parámetros λ para encontrar un subconjunto mejorado λ^* que pueda optimizar el modelo resultante M minimizando la función de pérdida $L(X, M)$. La idea principal es formalizada en la Ecuación 5 .

$$\begin{aligned} \lambda^* &= \underset{\lambda}{\operatorname{argmin}}\, L(X_P; a(X_T; \lambda)) \\ &= \underset{\lambda}{\operatorname{argmin}}\, F(\lambda; a, X_T, X_P, L), \end{aligned} \tag{5}$$

donde L denota la función de pérdida; $X_T \in X$ es el subconjunto de entrenamiento; $X_P \in X$ es el subconjunto de prueba, a es el algoritmos elegido; λ es el conjunto de hiper-parámetros que deben optimizarse y F es una función objetivo destinada a probar el rendimiento del nuevo conjunto de hiper-parámetros λ^*. En la tabla IV son descritos el conjunto de algoritmos A, sus hiper-parámetros inherentes λ y su rango correspondiente de valores.

Todos los algoritmos la optimización de hiper-parámetros fueron programados implementando Sklearn una libería de Python especial para el AA.

IV. Resultados

En ambos conjuntos de datos X se dividió en particiones de entrenamiento (X_{Train}) y de prueba (X_{Test}) mediante muestreos aleatorios.

Para X_{Train} y X_{Test} fue aplicado PCA en los conjuntos de datos *CIDDS-001* y *ISCX-Bot-2014* dando como resultado un subespacio de 2 dimensiones en ambos casos, además con estos se explicaba el 90 % y 82 % de la variabilidad de los datos, respectivamente.

Todo algoritmo se ejecutó de manera independiente en un entorno de programación basado en Python.

Gracias al método del sobremuestreo (SMOTE), la clase minoritaria pudo estar al nivel de la mayoritaría y así evitar el sobre ajuste al momento de probar cualquier conjunto de datos de X_{Test}. En las Tablas V y VI son mostrados los resultados antes de ser efectuadas las técnicas búsqueda de hiper-parámetros.

Tabla IV
Cartera de algoritmos y sus correspondientes hiper-parámetros

a	λ	**Valores /Rangos**
LR	Inversión de la fuerza de regularización del término C	{0.0001,0.01}
	Número de núcleos de CPU utilizados al paralelizar	all pc.
	Norma utilizada en la función de penalización.	ℓ_1, ℓ_2
	Algoritmo a utilizar en el problema de optimización	Linear LBFGS*, SAG†, SAGA‡
DT	La profundidad máxima del árbol	{1,9}
	No. de características a considerar para la mejor división	{1,50}
	Estrategia utilizada para elegir la división en cada nodo	Sqrt, Log_2 Best, Random
RF	Usar muestras de bootstrap cuando construya árboles	True, False
	La func. para medir la calidad de una división	Entropy,GINI§
	La profundidad máxima del árbol.	{1,10}
	No. de características a considerar para la mejor división	{1,10}
	Min. no. de muestras requeridas para dividir un nodo interno	{1,10}
	Min. no. de muestras requeridas para estar en un nodo de hoja	{1,10}
	Min. no. de muestras requeridas para dividir un nodo interno	{1,10}
	No. de árboles en el bosque	{1,4}
NB	Prior probabilities of the classes	{1,10}
LDA	Solver to use	Svd, Lsqr_2 Eigen,
	Shrinkage	{0,1}

*de memoria limitada BFGS
† Gradiente Promedio Estocástico
‡ Método de gradiente incremental rápido
§ Índice GINI

Tabla V
Resultados de rendimiento utilizando valores predeterminados de cada Algoritmo para el conjunto de datos CIDDS-001

a	**Precisión**	**Sensitividad**	**Exhaustividad**
LR	0.8825	0.8878	0.8855
DT	0.9987	0.9985	0.9986
RF	0.9989	0.9989	0.9989
NB	0.3995	0.4971	0.4054
LDA	0.8823	0.9169	0.8944

Mediante la validación cruzada de 10 pliegues con sub-conjuntos aleatorios de X_{Test} cada algoritmo $a \in A$ se probó mediante la iteración con su correspondiente conjunto de hiper-parámetros para encontrar un λ óptimo.

El rendimiento de cada algoritmo se basó en términos de Precisión, Sensitividad y Exhasutividad.

Teniendo en cuenta que las muestras de botnet son la clase objetiva para la clasificación; las muestras clasificadas correc-

Tabla VI
RESULTADOS DE RENDIMIENTO UTILIZANDO VALORES PREDETERMINADOS DE CADA ALGORITMO PARA EL CONJUNTO DE DATOS ISCX-BOT-2014

a	Precisión	Sensitividad	Exhaustividad
LR	0.8439	0.8764	0.8545
DT	0.9857	0.9850	0.9854
RF	0.9891	0.9902	0.9897
NB	0.7757	0.5035	0.4119
LDA	0.7314	0.6409	0.6501

Tabla VII
RESULTADOS DE RENDIMIENTO DESPUÉS DE APLICAR GRID-SEARCH PARA CADA ALGORITMO PARA EL CONJUNTO DE DATOS CIDDS-001

a	Precisión	Sensitividad	Exhaustividad
LR	0.8712	0.9131	0.8831
DT	**0.9885**	**0.9928**	**0.9906**
RF	**0.9870**	**0.9922**	**0.9895**
NB	0.8640	0.9105	0.8741
LDA	0.8648	0.8968	0.8762

tamente se conocen como verdaderos positivos (TP), mientras que las mal clasificadas se conocen como falsos positivos (FP). A su vez, las muestras clasificadas con precisión como benignas se reconocen como verdaderos negativos (TN), por el contrario, las muestras incorrectas clasificadas como benignas se denotan con falsos negativos (FN). En las Ecuaciones 6, 7 y 8 se describen brevemente las métricas de rendimiento utilizadas para la clasificación de botnet:

- Precision: es la relación entre las muestras de botnet predecidas correctamente y el total de muestras de botnet:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

- Sensitividad: es la proporción de muestras de botnets predecidas correctamente para todas las muestras clasificadas como maliciosas:

$$Sensitividad = \frac{TP}{TP + FN} \tag{7}$$

- Exhaustividad: es el promedio ponderado de $Precision$ y $Sensitividad$:

$$Exhaustividad = 2 * \frac{P \times R}{P + R} \tag{8}$$

Tabla VIII
RESULTADOS DE RENDIMIENTO DESPUÉS DE APLICAR GRID-SEARCH PARA CADA ALGORITMO PARA EL CONJUNTO DE DATOS ISCX-BOT-20141

a	Precisión	Sensitividad	Exhaustividad
LR	0.8519	0.8796	0.8553
DT	**0.9662**	**0.9630**	**0.9646**
RF	**0.9661**	**0.9630**	**0.9645**
NB	0.8132	0.6032	0.5110
LDA	0.7415	0.6410	0.6504

V. CONCLUSIONES

En este trabajo se propone una metodología para la detección inmediata de botnets en la nube, ya que muchos ataques como la Denegación de Servicios Distribuida (DDoS), el robo de información, phishing o filtraciones son cada vez más sofisticados y emulan comportamientos normales de flujo de red.

Durante tales comportamientos anómalos, se generan grandes cantidades de tráfico de red. Por lo tanto la detección de estos ataques se ha vuelto muy difícil y se requieren nuevas técnicas. Este motivación, hace que se proponga una metodología para una solución más efectiva basada en AA mediante AS, diferenciando entre el tráfico benigno del malicioso y así mejorar significativamente la detección de este tipo de malware.

En el estado del arte son presentadas muchas soluciones basadas en algoritmos de AS, sin embargo, no son explicados detalladamente los pasos adecuados para realizar el entrenamiento de los mismos y tampoco las técnicas que permitan reducir la función pérdida al buscar la mejor combinación de hiper-parámetros en la etapa de aprendizaje, esta búsqueda de hiper-parámetros es también llamada Grid-Search.

La mayoría de los modelos de AS cuentan con varios parámetros para ajustar su comportamiento, por lo tanto una alternativa para reducir el sobre ajuste de los datos es optimizar éstos parámetros por medio de un proceso conocido como Grid-Search, con ello se puede encontrar la combinación ideal que proporcione la mayor precisión. Es decir, se trata de una búsqueda exhaustiva por el paradigma de fuerza bruta en el que se especifica una lista de valores para diferentes parámetros y la computadora evalúa el rendimiento del modelo para cada combinación de éstos parámetros para obtener el conjunto óptimo que brinde el mayor rendimiento para cada algoritmo de AS.

Los hiper-parámetros mostraron una optimización significativa de desempeño en términos de precisión, exhaustividad y sensitividad dando los mejores resultados en cada conjunto de datos como se muestra a continuación:

- **Conjunto de Datos CIDDS-001:** Bosques Aleatorios dio 98.70 % de precisión, 99.22 % Sensitividad y Exhaustividad 99.06 %. Arboles de decisión 98.85 % de precisión, 99.28 % Sensitividad y Exhaustividad 98.95 %.
- **Conjunto de Datos ISCX-Bot-20141:** Bosques Aleatorios 96.61 % de precisión, 96.30 % Sensitividad y Exhaustividad 96.45 %. Arboles de decisión 96.62 % de precisión, 96.30 % Sensitividad y Exhaustividad 96.46 %.

A su vez se ha demostrado que con el método SMOTE se evitó un sobre ajuste en los resultados como se ve en la Tablas V y VI, creando muestras sintéticas no duplicadas de la clase minoritaria en comparación de los resultados con los valores predeterminados de cada algoritmo.

AGRADECIMIENTOS

Los autores le agradecen al Consejo Nacional de Ciencia y Tecnología (CONACYT), al Instituto Politécnico Nacional (IPN) y al Grupo de Análisis, Seguridad y Sistemas (GASS) de la Universidad Complutense de Madrid por su apoyo

en la elaboración de este trabajo. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326.

REFERENCIAS

[1] Abdulhammed, R., Faezipour, M., Abuzneid, A., & AbuMallouh, A. (2019). Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic. IEEE sensors letters, 3(1), 1-4.

[2] C. G. J. Putman, Abhishta and L. J. M. Nieuwenhuis, "Business Model of a Botnet,"2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, 2018, pp. 441-445.

[3] Beigi, Elaheh Biglar, et al. "Towards effective feature selection in machine learning-based botnet detection approaches.Çommunications and Network Security (CNS), 2014 IEEE Conference on. IEEE, 2014.

[4] Ring, M., Wunderlich, S., Gruedl, D., Landes, D., Hotho, A.: "Flow-based benchmark data sets for intrusion detection.Ïn: Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS), pp. 361-369. ACPI (2017).

[5] Ring, M., Wunderlich, S., Gruedl, D., Landes, D., Hotho, A.: Çreation of Flow-Based Data Sets for Intrusion Detection". In: Journal of Information Warfare (JIW), Vol. 16, Issue 4, pp. 40-53, 2017.

[6] Kotthoff, L., Gent, I. P., & Miguel, I. (2011, July). A preliminary evaluation of machine learning in algorithm selection for search problems. In Fourth Annual Symposium on Combinatorial Search.

[7] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. .ª practical guide to support vector classification."(2003): 1-16.

[8] Idhammad, M., Afdel, K., & Belouch, M. (2018). Detection system of HTTP DDoS attacks in a cloud environment based on information theoretic entropy and random forest. Security and Communication Networks, 2018.

[9] Su, S. C., Chen, Y. R., Tsai, S. C., & Lin, Y. B. (2018). Detecting p2p botnet in software defined networks. Security and Communication Networks, 2018.

[10] Hsu, F. H., Ou, C. W., Hwang, Y. L., Chang, Y. C., & Lin, P. C. (2017). Detecting Web-Based Botnets Using Bot Communication Traffic Features. Security and Communication Networks, 2017.

[11] Marnerides, A. K., Watson, M. R., Shirazi, N., Mauthe, A., & Hutchison, D. (2013, December). Malware analysis in cloud computing: Network and system characteristics. In 2013 IEEE globecom workshops (GC Wkshps) (pp. 482-487). IEEE.

[12] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3):37–52, 1987

[13] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. Applied logistic regression, volume 398. John Wiley & Sons, 2013.

[14] J. Ross Quinlan. Induction of decision trees. Machine learning, 1(1):81–106, 1986.

[15] Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

[16] Irina Rish et al. An empirical study of the naive bayes classifier.

[17] Huy NH, Frenzel S, Bandt C (2014) Two-step linear discriminant analysis for classification of eeg data. In: Spiliopoulou M, Schmidt-Thieme L, Janning R (eds) Data analysis, machine learning and knowledge discovery. Springer, Berlin, pp 51–59.

[18] Marnerides, A. K., & Mauthe, A. U. (2016, February). Analysis and characterisation of botnet scan traffic. In 2016 International conference on computing, networking and communications (ICNC) (pp. 1-7). IEEE.

[19] Kumar, R., & Kaur, T. (2014). Machine Learning based Traffic Classification using Low Level Features and Statistical Analysis. International Journal of Computer Applications, 108(12).

Detectando anomalías de integridad y veracidad en fuentes de datos IIoT

Iñaki Garitano, Mikel Iturbe, Enaitz Ezpeleta y Urko Zurutuza
Mondragon Unibertsitatea
Goiru 2, 20500 Arrasate-Mondragon
{igaritano,miturbe,eezpeleta,uzurutuza}@mondragon.edu

***Resumen*—El panorama de la seguridad en entornos industriales ha cambiado completamente en las últimas décadas. Desde las configuraciones primitivas iniciales, las redes industriales han evolucionado hacia entornos masivamente interconectados, desarrollando así el paradigma de Internet Industrial de las Cosas (IIoT). En IIoT, múltiples dispositivos heterogéneos colaboran mediante la recopilación, el envío y el procesamiento de datos. Estos entornos controlados han hecho posible el desarrollo de servicios de valor agregado basándose en los datos, los cuales mejoran la operación de los procesos industriales. Así, la verificación de los datos entrantes resulta indispensable, debido a que las decisiones tomadas serán erróneas si los datos en los que se basan no son correctos. En este capítulo, presentamos un sistema de detección de anomalías IIoT (ADS), que audita la integridad y la veracidad de los datos recibidos de las conexiones entrantes. Para este fin, el ADS incluye datos de campo (magnitudes físicas basadas en datos) y metadatos de conexión (intervalo entre las conexiones entrantes y el tamaño del paquete) en el mismo modelo de detección de anomalías. El enfoque se basa en el control estadístico multivariante de procesos y se ha validado utilizando datos reales de una planta de distribución de agua.**

***Index Terms*—Industrial Internet of Things, Detección de Anomalías, Veracidad de los Datos, Integridad de los Datos**

Tipo de contribución: *Investigación original*

I. Introducción

Las comunicaciones industriales han evolucionado drásticamente a lo largo de las últimas décadas. Desde los Controladores Lógicos Programables, *Programmable Logic Controllers* (PLCs), aislados y básicos de los años 60, hasta los totalmente interconectados Internet Industrial de las Cosas, *Industrial Internet of Things* (IIoT). En este nuevo paradigma, los dispositivos industriales, sensores y servidores colaboran para proporcionar servicios de valor añadido como la recogida de datos de campo. Esta colaboración se basa generalmente en la comunicación sobre redes inseguras como puede ser Internet. Además de los riesgos asociados al uso de canales de comunicación inseguras, otros riesgos técnicos pueden afectar a las fuentes de datos, comprometiendo así la veracidad y en consecuencia, la validez de los datos. Así, es necesario asegurar las comunicaciones y proporcionar a las personas encargadas de analizar los datos generados a partir de sistemas IIoT, información sobre la veracidad de los datos.

El IIoT, como nuevo paradigma, se encuentra todavía en desarrollo en términos de mecanismos de seguridad. Sin embargo, cuando se diseñan soluciones de seguridad para aplicaciones industriales, es necesario considerar las particularidades de estos entornos, los cuales difieren de las redes IT tradicionales. Sin embargo, estas particularidades se pueden utilizar para crear mecanismos de seguridad específicos.

En las redes industriales, en comparación con las redes IT tradicionales, la mayor parte del tráfico se genera a partir de procesos automatizados. En este contexto, los Sistemas de Detección de Anomalías, *Anomaly Detection Systems* (ADSes), resultan especialmente efectivos [1]. Los ADSes son considerados como un subconjunto de los Sistemas de Detección de Intrusiones, *Intrusión Detection Systems* (IDSes). La mayoría de los IDSes desplegados son del tipo basados en firmas; este tipo de IDSes monitorizan la red en busca de trazas comunes de actividades maliciosas, conocidos como firmas. Los ADSes por el contrario, son sistemas de monitorización que se centran en buscar patrones derivados del funcionamiento normal. Los IDSes basados en firmas resultan efectivos únicamente frente a amenazas conocidas cuyas características están registradas en la base de datos de firmas. Si ocurre un ataque desconocido, los IDSes basados en firmas no serán capaces de detectarlo. Por el contrario, en el caso de de los ADSes, es posible detectar ataques desconocidos debido a que no analizan patrones conocidos sino desviaciones de la normalidad. Sin embargo esto trae consigo un mayor número de falsos positivos en comparación con los IDSes basados en firmas.

Este trabajo presenta dos contribuciones. Por una parte, se presenta un sistema donde los datos de red se enriquecen añadiendo metadatos utilizando un framework escalable Big Data, apto para entornos IIoT reales. Y segundo, un ADS basado en Control Estadístico Multivariante de Procesos, *Multivariate Statistical Process Control* (MSPC) que monitoriza conexiones entrantes IIoT y proporciona datos originales y datos de conexiones con el objetivo de detectar y diagnosticar anomalías. Estas anomalías pueden variar desde un atacante realizando un ataque *Main-in-the-Middle* (MitM), hasta sensores con un funcionamiento erróneo o problemas de comunicación. La propuesta se valida utilizando datos reales de una planta de distribución de agua.

El resto del trabajo se organiza de la siguiente forma: la Sección II presenta trabajos relacionados en el área de la seguridad IIoT. La Sección III cubre MSPC, la técnica utilizada por el ADS. La Sección IV describe la aproximación general mientras que las Secciones V y VI discuten el experimento así como los resultados obtenidos. Finalmente, la Sección VII concluye el trabajo y expone algunos apuntes finales.

II. Trabajos relacionados

Debido a la criticidad de algunos sistemas industriales y la clara manifestación de la inseguridad obtenido a través de la seguridad por oscuridad, la comunidad científica ha mostrado gran interés en el área de la seguridad industrial. Entre las

distintas propuestas, el área de la detección de anomalías es especialmente activa, tanto a nivel de red [2] como a nivel de campo [3]. En el área del IIoT, se han realizado diversos trabajos con el objetivo de proporcionar un mayor nivel de seguridad [4].

Sajid et al. [5] por ejemplo, analizan el estado del arte actual y desafíos del futuro de los Sistemas de Supervisión y Adquisición de Datos, (SCADA), basados en el Internet de las Cosas, (IoT), en el entorno de la nube. Además, publican una colección de recomendaciones y buenas prácticas para asegurar este tipo de entornos: segregación de red, monitorización y análisis de la actividad de los dispositivos, análisis de logs, monitorización de la integridad de los archivos, análisis del tráfico de red, análisis de memoria, actualización periódica, testeo de vulnerabilidades y el uso de detectores de actividades maliciosas.

En el caso de la detección de anomalías, existen diversas propuestas para (I)IoT: Rajasegarar et al. [6] presentan un algoritmo distribuido para la detección de anomalías basado el modelos de datos creados a través de un cluster hyperesférico. El sistema fue implementado y testado en un red inalámbrica real. Thanigaivelan et al. [7] presentan otro ADS distribuido el cual se basa en la monitorización de vecinos para identificar comportamientos inusuales. A nivel de dispositivo, Summerville et al. [8] se centran en la moitorización de ligera de sistemas embebidos con poca capacidad de computación. A nivel de red, Stiawan et al. [9] presentan una solución de monitorización para la detección de anomalías. La propuesta es capaz de detectar y diagnosticar la causa de la anomalía en tiempo real.

La confianza en los nodos en el campo de las IIoT sigue siendo un reto abierto. Mientras que los nodos sean o sigan accesibles a atacantes puede resultar difícil evaluar la cantidad de confianza que los usuarios tienen sobre los nodos potencialmente comprometidos. Debido a ello, la confianza en entornos IoT sigue siendo un área de investigación abierta [10]. Bao y Chan [11] utilizan la honestidad, cooperativismo e interés en la comunidad como parámetros de referencia para evaluar la confianza en los nodos, asumiendo que la mayoría de los dispositivos IoT están relacionados con los humanos. Mahalle et al. [12] proponen un sistema de control de acceso basado en la confianza, centrado en entornos IoT dinámicos y descentralizados.

El presente trabajo pretende complementar los trabajos previamente presentados mediante un ADS centralizado que responde a los siguientes y previamente no resueltos objetivos: 1) detectar anomalías de red monitorizando patrones de conexión entrantes mediante el análisis de los datos adquiridos por el servidor, 2) detectar anomalías en los datos monitorizando los datos obtenidos y 3) reducir las necesidades de computación de los dispositivos IIoT realizando todo el procesamiento en la parte centralizada del servidor.

III. Control estadístico multivariante de procesos

Esta sección presenta la técnica principal utilizado en nuestro ADS, el Control Estadístico Multivariante de Procesos, *Multivariate Statistical Process Control* (MSPC). MSPC [13] es una metodología de monitorización de procesos que se basa en el uso de gráficas de control multivariante para detectar

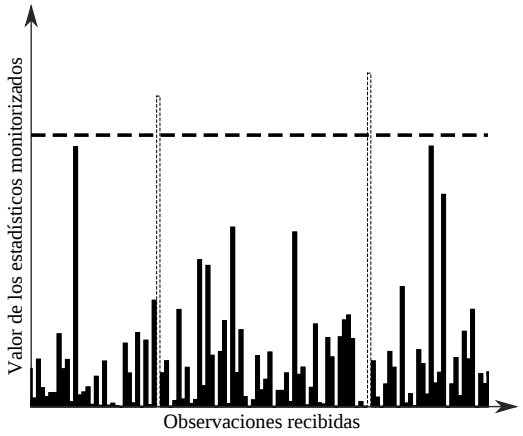


Figura 1. Ejemplo de un gráfico de control

cambios inusuales en los procesos monitorizados. Es una extensión de la técnica de Control Estadístico de Procesos, *Statistical Process Control* (SPC). Esta técnica ya ha sido propuesta como una solución viable para la detección de anomalías en los entornos IT [14] y para la detección y diagnóstico de anomalías de campo en los sistemas de control de procesos [15].

Stoumbos et al. [16] definen SPC como "un conjunto de métodos estadísticos utilizados extensivamente para monitorizar y mejorar la calidad y productividad de procesos de fabricación y servicios de operación. SPC conlleva la implementación de gráficos de control, los cuales se utilizan para detectar cambios en el proceso que pueden afectar la calidad de la salida".

La Figura 1 muestra un ejemplo de un gráfico de control con un 99 % de límite de control de nivel de confianza. En condiciones normales del proceso, el 99 % de todos los puntos caerán por debajo del límite de control. En ese caso, consideramos que el proceso se encuentra en un estado de *control estadístico*. Es importante no confundir este término con otras expresiones similares, como bucle de control o control de retroalimentación automática, ya que se refieren a diferentes conceptos. El control estadístico se refiere al estado del proceso donde solo están presentes las causas comunes de variación [13].

Es probable que la existencia de series de observación consistentes sobre el límite de control establecido se atribuya a una nueva causa especial. En el caso de un proceso físico, esta fuente de variación puede atribuirse a ataques o perturbaciones del proceso, es decir, una anomalía.

La naturaleza univariante de SPC significa que solo una única variable se monitoriza y visualiza en un gráfico de control. Sin embargo, los procesos industriales y los entornos de IIoT son multivariantes por naturaleza, ya que muchas variables de proceso se observan en una planta (p.ej., temperaturas, presiones, volúmenes o distancias). Como la monitorización de todas las variables con SPC sería poco práctico, solo se monitorizan algunas de ellas, generalmente las relacionadas con la calidad del producto (p.ej., la pureza de los productos químicos producidos).

Sin embargo, la monitorización de algunas variables relacionadas con la calidad es poco práctico. El enfoque no

considera la información que proporcionan otras variables de proceso. Por ejemplo, el diagnóstico de un evento anómalo es complicado, ya que se basa en el conocimiento experto y en una inspección de cada momento de las variables de proceso [17].

MSPC pretende resolver estos problemas al proporcionar herramientas para monitorizar todas las variables medidas de una manera eficiente. En ese sentido, MSPC no solo monitoriza la evolución de la magnitud de la variable sino también la evolución de la relación que tiene con otras variables. Para este fin, una de las técnicas principales que utiliza MSPC es el Análisis de Componentes Principales, *Principal Component Analysis* (PCA).

III-A. MSPC basado en PCA

Consideremos los datos históricos del proceso como un conjunto de datos bidimensionales de $\mathbf{X} = N \times M$, donde las variables M se miden durante N observaciones. PCA transforma el espacio M-dimensional de la variable original en un nuevo subespacio donde la varianza es máxima. Convierte las variables originales en un nuevo conjunto de variables no correlacionadas (generalmente menos en número), llamadas Componentes Principales, **Principal Components** (PCs) o Variables Latentes.

Para unos PCs, $\mathbf{X}$ y A, centrados en media y autoescalados[1], PCA sigue la siguiente expresión:

$$\mathbf{X} = \mathbf{T_A}\mathbf{P_A^t} + \mathbf{E_A} \tag{1}$$

donde $\mathbf{T_A}$ es la matriz de puntuación $N \times A$, es decir, las observaciones originales representadas de acuerdo con el nuevo subespacio; $\mathbf{P_A^t}$ es la matriz de carga $M \times A$, que representa la combinación lineal de las variables originales que forman cada una de las PC; finalmente, $\mathbf{E_A}$ es la matriz de residuos de $N \times M$.

En la MSPC basada en PCA, tanto las puntuaciones como los residuales se monitorizan, cada uno en un cuadro de control separado [18]. Por un lado, para comprender las puntuaciones, se supervisa la estadística D o la T^2 de Hotelling [19]. Por otro lado, en el caso de los residuos, la estadística elegida es la estadística Q o SPE [20].

Para una observación de n, ambas estadísticas se calculan de la siguiente manera:

$$D_n = \sum_{a=1}^{A} \left(\frac{t_{an} - \mu_{\mathbf{t}_a}}{\sigma_{\mathbf{t}_a}} \right)^2 ;\; Q_n = \sum_{a=1}^{A} (e_{nm})^2 \tag{2}$$

donde t_{an} es la puntuación de la observación en el PC a, $\mu\ mathbft_a$ y $\sigma_{\mathbf{t}_a}$ representa la media y la desviación estándar de las puntuaciones de la PC a en los datos de entrenamiento, respectivamente, y e_{nm} representa el valor residual correspondiente a la variable m.

Los estadísticos D y Q se calculan para cada una de las observaciones en los datos de entrenamiento sin anomalías, y se establecen límites de control para cada uno de los dos gráficos. Los datos de entrenamiento se inspeccionaron previamente a través del análisis de datos exploratorio con el objetivo de eliminar valores atípicos que podrían cambiar los valores de D y Q. Más tarde, estos estadísticos se calculan para los datos entrantes y se trazan en el gráfico de control. Cuando se produce un cambio inesperado en una (o más) de las variables M originales, uno (o ambos) de estos estadísticos irán fuera de los límites de control. Por lo tanto, un escenario de monitorización de M dimensiones se convierte en uno bidimensional.

Un evento se considera anómalo cuando tres observaciones consecutivas superan el límite de control del nivel de confianza del 99 % en cualquiera de las estadísticas monitorizadas [21]. Dejar algunas observaciones fuera de los límites (1 % de las observaciones con un límite de control establecido en el nivel de confianza del 99 %) mejora el rendimiento de los gráficos de control en la fase de monitorización [21], [14].

Una vez que se ha detectado una anomalía, el diagnóstico se lleva a cabo mediante gráficos de contribución [17]. Estas gráficas muestran la contribución de las variables originales a un evento anómalo. Los detalles del cálculo y análisis de las parcelas de contribución se pueden encontrar en el trabajo de Alcala y Qin [22].

En este trabajo, usamos gráficos oMEDA [23] para diagnosticar las causas de anomalías al relacionar eventos anómalos con las variables originales. En esencia, los gráficos oMEDA son diagramas de barras donde los valores más altos o más bajos en un conjunto de variables reflejan su contribución a un grupo de observaciones. Por lo tanto, cuando se calculan en un grupo de observaciones dentro de un evento anómalo, las variables más relevantes relacionadas con ese evento en particular serán las que tengan las barras más altas y más bajas. Aunque similar, una de las principales diferencias de las gráficas de oMEDA con las gráficas de contribución tradicionales es que las gráficas de oMEDA son capaces de comparar diferentes conjuntos de observaciones, mientras que las gráficas tradicionales solo pueden computar una sola serie de ellas. En ese sentido, las gráficas oMEDA pueden considerarse una extensión de las gráficas de contribución. En este caso, para calcular oMEDA, primero definimos una variable *dummy*, $\mathbf{d}$, un vector de longitud N, en el que las observaciones anómalas que se deben calcular se marcan con 1, dejando el resto como 0.

Para un conjunto de observaciones marcadas en $\mathbf{d}$, oMEDA se calcula de la siguiente manera:

$$d^2_{A,(i)} = \frac{1}{N} \cdot \left(2 \cdot \sum_{(i)}^{d} - \sum_{A,(i)}^{d} \right) \cdot \left| \sum_{A,(i)}^{d} \right| \tag{3}$$

donde $\sum_{(i)}^{A}$ y $\sum_{A,(i)}^{A}$ representa la suma ponderada de elementos para la variable i en $\mathbf{X}$ y su proyección $\mathbf{X}_A$ de acuerdo con los pesos en $\mathbf{d}$, respectivamente. Los valores absolutos mayores de d^2 indicarán una mayor contribución de esa variable en la causa de la observación anómala.

IV. Propuesta sugerida

A lo largo de esta sección se describe el ADS para IIoT basado en MSPC. Los entornos IIoT son fundamentalmente entornos multivariantes, donde constantemente se monitorizan las distintas cualidades físicas. Generalmente, la monitorización se realiza utilizando recolectores locales tales como sensores, los cuales coleccionan y después envían los datos de campo recogidos a un dispositivos que ejerce de puerta

[1]Normalizado a cero la media y la varianza de la unidad

de enlace para después ser procesados en la nube. Los dispositivos que ejercen de puerta de enlace son dispositivos hardware que hacen posible enviar información a la nube. En la nube, se recoge información desde distintas puertas de enlace IIoT y después se procesa.

La propuesta sugerida a lo largo de esta sección realiza la detección de anomalías en cuatro distintas fases tal y como muestra la Figura 2:

Enriquecimiento de datos Cuando un dato adquirido por el dispositivo IIoT de campo llega a la nube, se evalúan algunas métricas estadísticas de nivel de red, como el tamaño del paquete y el intervalo de tiempo entre el último paquete recibido. Estos nuevos datos derivados se añaden al conjunto de datos, creando así un conjunto de datos ciber-físico híbrido: valores físicos recogidos por los dispositivos de campo junto con datos de red procesados. Los datos de red se pueden utilizar posteriormente para la detección de anomalías a nivel de red, como las latencias inusuales.

Creación del modelo para la detección de anomalías Una vez enriquecido el conjunto de datos, el ADS lo utiliza para construir el modelo MSPC. Con el objetivo de descartar valores atípicos, se realiza un análisis exploratorio para limpiar el conjunto de datos. Después, se calculan los límites estadísticos D y Q.

Detección de Anomalías Una vez que el modelo ha sido creado y los límites han sido establecidos, para cada lectura se calculan D y Q, y se compara el resultado obtenido con los límites establecidos. Si se registran consecutivamente tres lecturas que superan los límites, el evento se establece como anómalo.

Diagnóstico de anomalías Una vez que se ha detectado una anomalía, se computa el vector oMEDA sobre la primera lectura fuera de límites para analizar la contribución de cada variable al evento anómalo. En base a la gráfica oMEDA, el operador puede verificar si ha ocurrido algo inusual a nivel de planta (lectura anómala) o a nivel de red (conexión de red inestable). Así, las fuentes inestables o anómalas se pueden etiquetar como no confiables para futuros análisis sobre los datos recogidos.

IIoT es un paradigma escalable, donde se pueden añadir tanto nuevos sensores como nuevas funcionalidades, y el cual requiere una arquitectura escalable para soportar una creciente complejidad de los datos. En este sentido, el ADS se ha desarrollado utilizando herramientas Big Data como Apache Kafka para la fase de enriquecimiento de los datos como Apache Spark [24] para llevar a cabo las distintas fases de detección de anomalías diagnóstico.

V. Entorno de experimentación

Esta sección describe los recursos de hardware y software, la topología de la red y la configuración experimental, considerando las condiciones de anomalía y normalidad junto con los experimentos propuestos.

V-A. Arquitectura

El objetivo principal es emular una configuración industrial común donde diferentes sensores instalados localmente envían información a servidores de nube privada / pública siguiendo un patrón periódico. Los sensores pueden estar conectados a Internet o no y, por lo tanto, la información puede enviarse a través de Internet o por medios de comunicación privados.

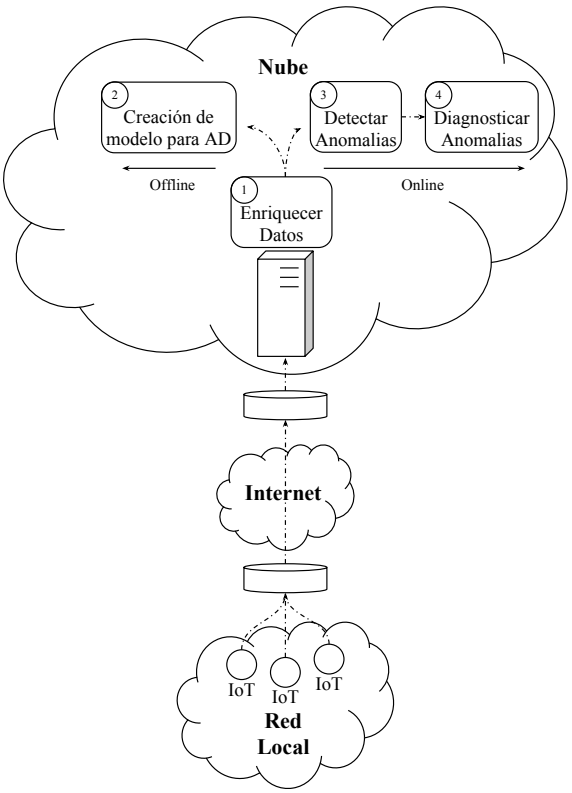


Figura 2. Fases del ADS desarrollado

Tres nodos, cada uno de ellos emulando un dispositivo IoT o un servidor y dos redes de comunicación diferentes componen el equipo de hardware necesario para esta configuración. Mientras que uno de los servidores captura y procesa toda la información recopilada en una interfaz específica, el otro servidor configura el tráfico en las condiciones deseadas y controladas. El dispositivo emulador de IoT recopila y envía algunas variables de proceso que reflejan el estado del proceso. La recopilación y el reenvío de datos se realizan de forma regular, siguiendo una frecuencia preestablecida.

El lado del software de la configuración experimental se compone de cuatro herramientas de software: 1) Un script de Python que recopila y reenvía la información de forma regular, 2) una herramienta de modelado de tráfico de red, *Traffic Control* (TC) [25], que entre otras cosas permite agregar un retraso preestablecido o descartar paquetes, 3) una versión modificada de Apache Kafka y 4) una instancia de Apache Hadoop.

El script de Python utilizado para emular el proceso de recopilación y reenvío de datos está disponible públicamente [2] para realizar más pruebas y reconstruir los resultados mostrados en este trabajo. El script básicamente obtiene un archivo CSV, una dirección IP de destino y un valor de frecuencia de envío de paquetes como entrada, y como resultado envía los valores de las variables de cada fila a la dirección IP de

[2]https://bitbucket.org/danzsecurity/dataforwarder

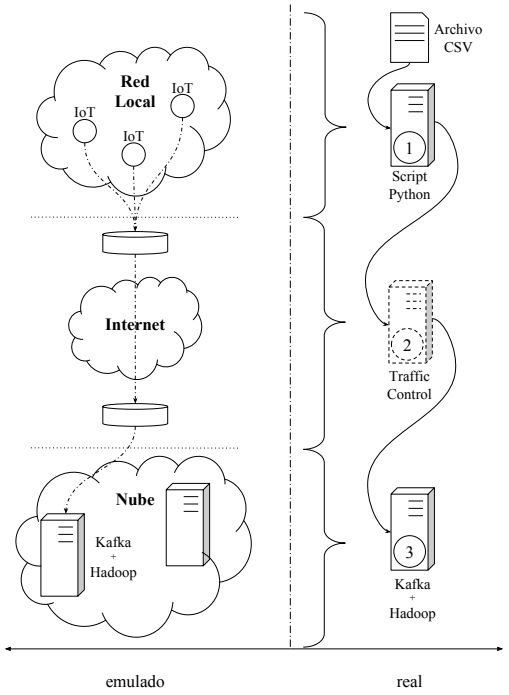


Figura 3. Topología del experimento

destino en un período determinado. El formato de archivo y el protocolo utilizado para enviar datos son JSON y HTTP respectivamente. El script de Python se instala en un host que ejecuta la distribución GNU/Linux Debian.

La herramienta de configuración de paquetes de red TC permite descartar paquetes de forma aleatoria o en función de otros parámetros. Además, la herramienta también puede introducir retrasos específicos o aleatorios. Dentro de esta configuración, TC se instala en un servidor separado con dos interfaces de red, ejecutando la distribución GNU/Linux Debian y con la función de reenvío de IP activada.

Finalmente, tanto la versión modificada de Apache Kafka como Apache Hadoop se instalan en el mismo servidor. De nuevo, ejecutando la distribución GNU/Linux Debian. La versión modificada de Apache Kafka, el Proxy Kafka REST, originalmente desarrollado por Confluent [26], se modificó para evaluar automáticamente algunas métricas relacionadas con los paquetes capturados. Esas métricas incluyen el intervalo de tiempo entre dos paquetes consecutivos y el tamaño de cada paquete. Junto con los paquetes recibidos, las métricas evaluadas se envían a Apache Hadoop para fines de almacenamiento y procesamiento. La versión modificada de Apache Kafka también está disponible públicamente en[3].

Todas estas herramientas de software permiten la emulación de un IoT real para el caso de uso/escenario de reenvío de datos a la nube. El reenvío de datos no solo imita los retrasos reales y la pérdida de paquetes, sino que también permite ataques MitM modificando los valores de los datos. Por lo tanto, se pueden emular anomalías o ataques tanto de tiempo como de modificación de valor.

La Figura 3 muestra tanto la topología emulada como la real. Como se muestra, la topología emulada se compone de tres redes diferentes: 1) una red local, 2) Internet y 3) una red en la nube. La red local es donde se ubican los diferentes dispositivos de IoT. Estos dispositivos miden básicamente el proceso y las variables ambientales, como la temperatura, siguiendo una periodicidad preestablecida; después, envían todas las medidas a un servidor en la nube. Por otro lado, la red de Internet podría ser una sola red pública o privada o una combinación de ambas; como en las redes reales, los paquetes pueden ser retrasados o eliminados de forma aleatoria debido a una falla de la red. Finalmente, está la red en la nube, que puede ser una infraestructura de nube pública, privada o híbrida y administrada por un proveedor de servicios en la nube, empresa externa o internamente. La red en la nube aloja un servidor dedicado a la adquisición y almacenamiento de todos los datos enviados por el dispositivo IoT. Además, evalúa las métricas necesarias y las almacena junto con los datos adquiridos.

La red real está compuesta por tres servidores conectados directamente a través de dos redes diferentes. Dos de cada tres servidores, el primero y el tercero, tienen una sola interfaz de red, mientras que el segundo servidor tiene dos interfaces. El último funciona como un puente transparente, reenviando paquetes de una interfaz a la otra y retrasando o eliminando paquetes.

Durante el experimento, se crearon dos conjuntos de datos diferentes: 1) un conjunto de datos de normalidad y 2) un conjunto de datos alterado manualmente o de anomalía. La Figura 4 muestra las configuraciones y los servidores donde los valores se modificaron y los paquetes de red se retrasaron o se eliminaron. Ambas configuraciones obtuvieron el mismo archivo CSV como entrada; sin embargo, la salida se almacenó en dos archivos diferentes.

A lo largo de los experimentos, el primer servidor lee una fila del archivo CSV en un período preestablecido. Luego, algunos valores se modifican según el tipo de conjunto de datos que estamos creando y se envían al segundo servidor. El mismo enfoque se aplica en el segundo servidor. En condiciones normales, ningún paquete se retrasa ni se descarta. Sin embargo, bajo condiciones alteradas manualmente, algunos de ellos se retrasan o eliminan al azar. Finalmente, el tercer servidor, evalúa un conjunto de métricas y las almacena, junto con los datos adquiridos como un conjunto de datos para su posterior análisis.

Como resultado, la configuración experimental proporciona dos conjuntos de datos diferentes con la misma entrada, uno de ellos, el conjunto de datos normal, creado en condiciones normales y el otro, el conjunto de datos de anomalías, con la modificación de algunos valores y el retraso o eliminación de paquetes.

V-B. Conjunto de datos

El conjunto de datos utilizado es un conjunto de datos real que proviene de una planta de distribución de agua en el norte de España. Allí, varias variables, que se muestran en la Tabla I, son monitorizadas para asegurar la calidad del agua.

Enriquecimos el conjunto de datos con las siguientes variables, en función de los datos de red recibidos: Δt (tiempo transcurrido desde que se recibió la última lectura, en ms) y el tamaño del paquete de red en KB. Por lo tanto, el conjunto

[3]https://bitbucket.org/danzsecurity/modifiedkafkarest

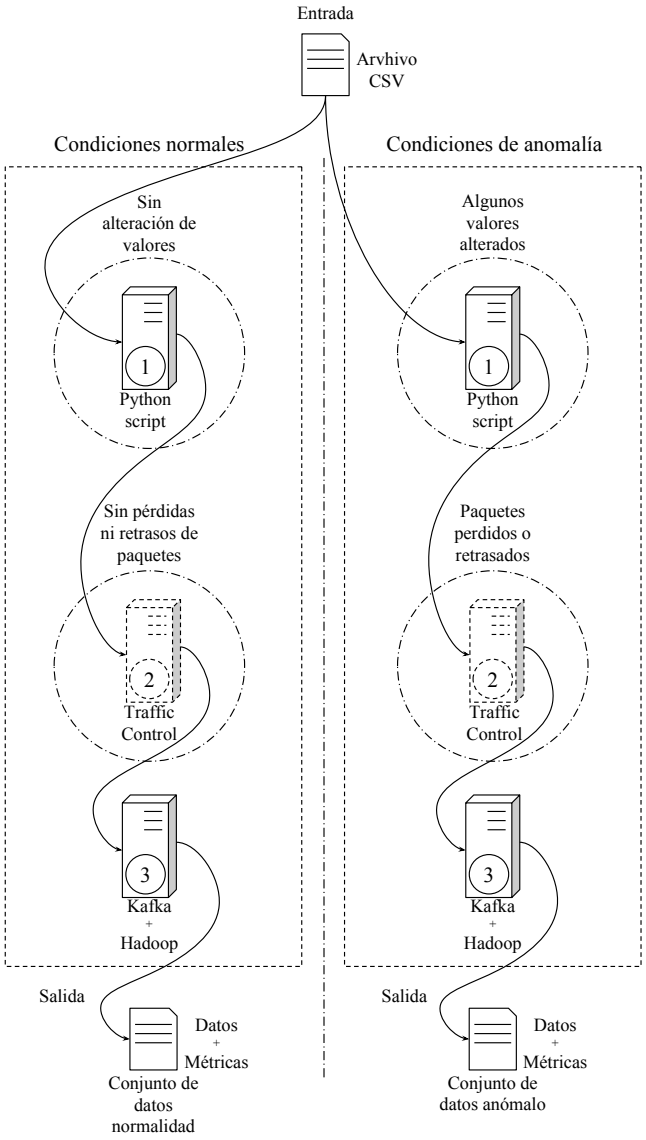


Figura 4. Condiciones normales vs anomalías

Tabla I
VARIABLES ORIGINALMENTE PRESENTES EN EL CONJUNTO DE DATOS

Variable	Unidad
Acidez	pH
Temperatura	°C
Conductividad	$\mu S/cm$
Oxigeno disuelto	mg/l
Reducción de Tensión	mV
Materia orgánica	numero de ocurrencias/m
Turbidez	NTU
Niveles de amonio	mgN/l

de datos de validación final consta de 10 variables, con un total de 22000 lecturas.

V-C. Experimentos

Para validar nuestra propuesta, hemos diseñado un conjunto de experimentos sobre el conjunto de datos explicado anteriormente. Estos experimentos se muestran en la Tabla II. Todas las variaciones del ataque se realizaron en la parte superior del conjunto de datos, donde el nodo central modifica el tráfico antes de transmitirlo a la nube.

VI. RESULTADOS

Esta sección describe los resultados obtenido al aplicar la propuesta presentada sobre el entorno de experimentación descrita en la Sección V. Además, esta sección expone las gráficas oMEDA de las anomalías detectadas. Los cuatro escenarios han sido identificados como anómalos, y los gráficos oMEDA han sido calculados sobre la primera observación fuera de límites.

VI-A. Escenario 1: Man-in-the-Middle para modificar el tamaño del paquete

La Figura 5 muestra el gráfico oMEDA para el escenario en el que el atacante dobla el tamaño de paquete. Como se puede observar, la gráfica oMEDA muestra que la variable relacionada con el tamaño de paquete es el principal contri-

Tabla II
ESCENARIOS DE VALIDACIÓN

Escenario	Descripción
Escenario 1	Un atacante realiza un ataque Man-in-the-Middle y modifica el tamaño del paquete
Escenario 2	Un atacante realiza un ataque Man-in-the-Middle que elimina la mitad de los paquetes, que no llegan a la nube
Escenario 3	Un atacante realiza un ataque Man-in-the-Middle y modifica la lectura de pH y temperatura. En la nube se recibe la siguiente lectura: $pH_{wat} = 9$ y $T_{wat} = 23$, ambos más altos que la media.
Escenario 4	Un atacante realiza un ataque Man-in-the-Middle, elimina la mitad de los paquetes y, al mismo tiempo, inyecta el valor $pH = 5$, más bajo de lo normal

buyente a la anomalía debido a que tiene un tamaño mayor que el que debiera.

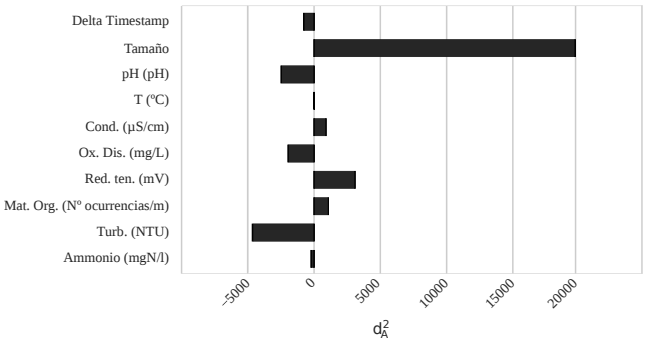


Figura 5. oMEDA para el diagnóstico de anomalías del Escenario 1

VI-B. Escenario 2: Man-in-the-Middle para eliminar uno de cada dos paquetes

En el escenario 2, el atacante elimina uno de cada dos paquetes. Tal y como se muestra en la Figura 6, la variable que más contribuye a la anomalía es el tiempo entre paquetes.

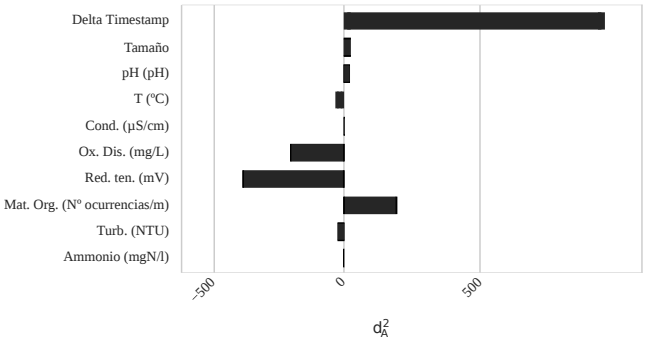


Figura 6. oMEDA para el diagnóstico de anomalías del Escenario 2

VI-C. Escenario 3: Man-in-the-Middle para modificar los valores del pH y la temperatura

En el tercer escenario, el atacante no elimina ni altera el tamaño de ningún paquete. En este caso, realiza un ataque de integridad y establece los valores de acidez y temperatura del agua a valores arbitrarios. Como muestra la gráfica oMEDA correspondiente, Figura 7, es apreciable como el nivel de pH es mayor que la tasa usual, al igual que la temperatura. Esto se debe a que la temperatura del agua varia a lo largo del año, mientras que los niveles de pH se mantienen constantes, así los cambios más pequeños en el pH ejercen grandes variaciones en las gráficas oMEDA.

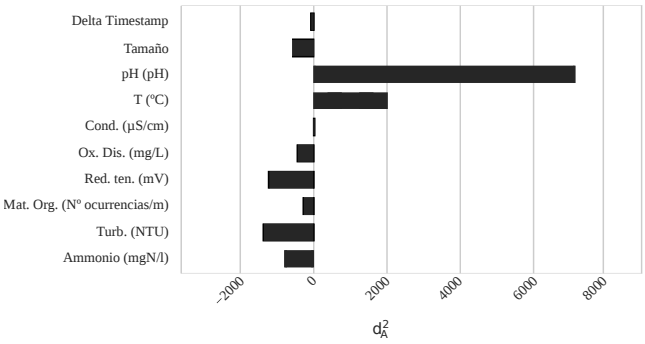


Figura 7. oMEDA para el diagnóstico de anomalías del Escenario 3

VI-D. Escenario 4: Man-in-the-Middle para eliminar uno de cada dos paquetes y modificar el valor del pH

En el último escenario, como combinación de los escenarios 2 y 3, el atacante elimina la mitad de los paquetes mientras inyecta un valor de pH menos que el habitual. Como muestra la Figura 8, podemos observar el incremento del intervalo entre paquete y como la disminución de los valores de pH.

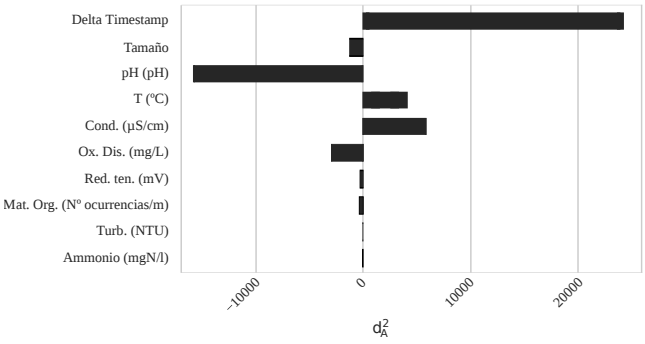


Figura 8. oMEDA para el diagnóstico de anomalías del Escenario 4

De esta forma, el ADS es capaz de detectar anomalías y de establecer sus causas. De la misma forma, mientras que los datos de entrada se han etiquetado como anómalos, el operador puede identificarlos como “poco veraces”, mostrando así al analizador de los datos que los datos no son fiables.

VII. CONCLUSIONES

Este trabajo presenta un sistema de detección de anomalías para entornos IIoT. Dicho sistema proporciona a los operadores información útil acerca de la causas de la anomalías además de si las fuentes de información IIoT son fiables o no; esto último dependiendo de la tasa de fallos y la causa de las anomalías. Nuestra propuesta se basa en el Control

Estadístico Multivariante de Procesos, *Multivariate Statistical Process Control* (MSPC), que permite detectar y diagnosticar anomalías basándose en lo datos. El sistema de detección de anomalías primero enriquece los datos disponibles con metadatos de red, para luego construir un modelo de detección de anomalías y proceder a la detección de anomalías online. Una vez detectadas las anomalías, se utilizan las gráficas oMEDA para diagnosticar la causa de las mismas.

Esta propuesta a sido implementa a través de herramientas Big Data y ha sido validada utilizando un conjunto de datos real obtenida de una distribuidora de agua. Los resultados muestran que es posible detectar y diagnosticar anomalías de distinta naturaleza, incluso incluyendo datos no presentes en el conjunto de datos original.

Para desarrollar aun más la solución, las variables relacionadas con la red se pueden diseñar para escenarios específicos yendo más allá que el tamaño e intervalo entre paquetes. Para un mayor desarrollo de la solución, la construcción de variables relacionadas con la red puede diseñarse para escenarios específicos y pueden ir mucho más allá del tamaño y el intervalo entre paquetes. La elección o construcción de características relevantes para la detección de anomalías es un campo de investigación en sí mismo, y enfoques como este pueden beneficiarse enormemente de los resultados en este área.

Además, la cuantificación del resultado de la detección y el diagnóstico de anomalías en una escala continua pueden ayudar a determinar un verdadero "valor de confianza" para cada una de las fuentes, que informará a los operadores si una fuente de datos produce más anomalías (y, por lo tanto, es menos confiable), o por el contrario, es una fuente que produce pocas anomalías.

VIII. Agradecimientos

Iñaki Garitano está parcialmente financiado por INCIBE bajo la beca "INCIBEC-2015-02495" correspondiente a "Ayudas para la Excelencia de los Equipos de Investigacion avanzada en ciberseguridad". Este trabajo ha sido desarrollado por el grupo Sistemas Inteligentes para Sistemas Industriales financiado parcialmente por el Departamento de Educación del Gobierno Vasco. Este trabajo ha sido financiado parcialmente por el Departamento de Desarrollo Económico e Infraestructuras del Gobierno Vasco a través del proyecto "CYBERPREST", "KK-2018/00076".

Referencias

[1] M. Cheminod, L. Durante, and A. Valenzano, "Review of Security Issues in Industrial Networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 277–293, 2013.

[2] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[3] D. I. Urbina, J. Giraldo, A. A. Cardenas, J. Valente, M. Faisal, N. O. Tippenhauer, J. Ruths, R. Candell, and H. Sandberg, "Survey and new directions for physics-based attack detection in control systems. NIST GCR 16-010," National Institute of Standards and Technology, Tech. Rep., Nov 2016.

[4] A. R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[5] A. Sajid, H. Abbas, and K. Saleem, "Cloud-assisted iot-based scada systems security: A review of the state of the art and future challenges," *IEEE Access*, vol. 4, pp. 1375–1384, 2016.

[6] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Hyperspherical cluster based distributed anomaly detection in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 74, no. 1, pp. 1833 – 1847, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731513002013

[7] N. K. Thanigaivelan, E. Nigussie, R. K. Kanth, S. Virtanen, and J. Isoaho, "Distributed internal anomaly detection system for internet-of-things," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan 2016, pp. 319–320.

[8] D. H. Summerville, K. M. Zach, and Y. Chen, "Ultra-lightweight deep packet anomaly detection for internet of things devices," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, Dec 2015, pp. 1–8.

[9] D. Stiawan, M. Y. Idris, R. F. Malik, S. Nurmaini, and R. Budiarto, "Anomaly detection and monitoring in internet of things communication," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, Oct 2016, pp. 1–4.

[10] S. Sicari, A. Rizzardi, L. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146 – 164, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614003971

[11] F. Bao and I.-R. Chen, "Dynamic trust management for internet of things applications," in *Proceedings of the 2012 International Workshop on Self-aware Internet of Things*, ser. Self-IoT '12. New York, NY, USA: ACM, 2012, pp. 1–6. [Online]. Available: http://doi.acm.org/10.1145/2378023.2378025

[12] P. N. Mahalle, P. A. Thakre, N. R. Prasad, and R. Prasad, "A fuzzy approach to trust based access control in internet of things," in *Wireless VITAE 2013*, June 2013, pp. 1–5.

[13] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, no. 3, pp. 403–414, 1995.

[14] J. Camacho, A. Pérez Villegas, P. García Teodoro, and G. Maciá Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118–137, 2016.

[15] M. Iturbe, J. Camacho, I. Garitano, U. Zurutuza, and R. Uribeetxeberria, "On the feasibility of distinguishing between process disturbances and intrusions in process control systems using multivariate statistical process control," in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*. Toulouse, France: IEEE, Jun. 2016, pp. 155–160.

[16] Z. G. Stoumbos, M. R. Reynolds Jr, T. P. Ryan, and W. H. Woodall, "The state of statistical process control as we proceed into the 21st century," *Journal of the American Statistical Association*, vol. 95, no. 451, pp. 992–998, 2000.

[17] T. Kourti, "Process analysis and abnormal situation detection: from theory to practice," *Control Systems, IEEE*, vol. 22, no. 5, pp. 10–25, 2002.

[18] J. Camacho, A. Pérez Villegas, R. A. Rodríguez Gómez, and E. Jiménez Mañas, "Multivariate Exploratory Data Analysis (MEDA) Toolbox for Matlab," *Chemometrics and Intelligent Laboratory Systems*, vol. 143, pp. 49–57, 2015.

[19] H. Hotelling, "Multivariate quality control," *Techniques of Statistical Analysis*, 1947.

[20] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[21] H.-J. Ramaker, E. N. Van Sprang, J. A. Westerhuis, S. P. Gurden, A. K. Smilde, and F. H. Van Der Meulen, "Performance assessment and improvement of control charts for statistical batch process monitoring," *Statistica Neerlandica*, vol. 60, no. 3, pp. 339–360, 2006.

[22] C. F. Alcala and S. J. Qin, "Analysis and generalization of fault diagnosis methods for process monitoring," *Journal of Process Control*, vol. 21, no. 3, pp. 322–330, 2011.

[23] J. Camacho, "Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models," *Journal of Chemometrics*, vol. 25, no. 11, pp. 592–600, 2011.

[24] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[25] The Linux Foundation, "Iproute2," https://wiki.linuxfoundation.org/networking/iproute2, accessed: 2018-08-18.

[26] Confluent, "Confluent rest proxy," https://docs.confluent.io/current/kafka-rest/docs, accessed: 2018-08-18.

Metodología supervisada para la obtención de trazas limpias del servicio HTTP

Jesús Díaz-Verdejo
Universidad de Granada
ETSITT - CITIC
jedv@ugr.es

Rafael Estepa
Universidad de Sevilla
ETS Ingenieros. 41092 Sevilla
rafaestepa@us.es

Antonio Estepa
Universidad de Sevilla
ETS Ingenieros 41092 Sevilla
aestepa@us.es

Germán Madinabeita
Universidad de Sevilla
ETS Ingenieros 41092 Sevilla
german@trajano.us.es

Resumen- **Disponer de datos adecuados para el entrenamiento, evaluación y validación de sistemas de detección de intrusos basados en anomalías representa un problema de índole práctica relevante. Las características requeridas para los datos plantean una serie de retos contrapuestos entre los que destaca la necesidad de disponer de un volumen significativo de datos reales que no contenga instancias de ataques. Esto implica un proceso de limpieza y supervisión que puede resultar muy costoso si se realiza manualmente. En este trabajo planteamos una metodología para automatizar en lo posible la adquisición y acondicionamiento de trazas del servicio HTTP para la detección de ataques basada en URI. Esta metodología se aplica con buenos resultados sobre una traza real como caso de estudio.**

***Index Terms*- detección de intrusos, captura de datos, bases de datos de entrenamiento**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Los sistemas de detección de intrusiones (IDS, del inglés Intrusion Detection Systems) [1] constituyen una de las herramientas fundamentales de los sistemas de monitorización de la seguridad (NSM, del inglés *Network Monitoring Systems*) [2], necesarios para la gestión de la seguridad en los sistemas en red. Entre los IDS podemos diferenciar dos aproximaciones básicas con características diferenciadoras respecto de la tipología de ataques que pueden detectar: la basada en firmas (SIDS, *Signature-based IDS*) y la basada en anomalías (AIDS, *Anomaly-based* IDS). Sin entrar en excesivo detalle, por no ser el objeto de este trabajo, podemos afirmar que los SIDS se especializan en la detección de ataques conocidos, mientras que los basados en anomalías tienen como objetivo fundamental la detección de ataques novedosos (0-day). Para ello, los AIDS utilizan modelos que representan la actividad del sistema (habitualmente la actividad normal) a partir de los que deciden la existencia de incidentes o ataques mediante diferentes técnicas. En el contexto de la detección basada en red, se requiere de una base de datos de tráfico para el entrenamiento de los modelos de normalidad (o anormalidad). Por otra parte, el análisis de la efectividad de las diversas soluciones propuestas en cada escenario requiere también de datos adecuados para evaluar y validar las técnicas usadas. Por tanto, el desarrollo de sistemas AIDS requiere de conjuntos de datos con una triple finalidad [3]: entrenar, evaluar y validar los modelos y técnicas.

La adquisición o generación de estos conjuntos de datos dista de ser un problema trivial [4]. Entre los principales retos podemos destacar que los datos deben ser representativos del sistema a modelar, lo que implica que deben ser obtenidos en condiciones reales y en un volumen suficiente. Sin embargo, dado que se utilizarán para entrenar el sistema, el tráfico capturado debe estar libre de ataques o éstos deben estar identificados (etiquetado). Estos requisitos plantean un serio problema relacionado con el proceso de limpieza y acondicionamiento de los datos capturados (*sanitization* [5] en la literatura en inglés). Por una parte, hay que eliminar las posibles instancias de ataque que hayan sido incorporadas en el conjunto de datos durante la captura y, por otra parte, hay que eliminar los posibles artefactos existentes en la misma y que pueden aparecer por diversas causas. En ambos casos es necesaria una supervisión manual de los datos que puede resultar excesivamente costosa e incluso inviable si el tamaño de la base de datos es relevante.

En primera aproximación, para eliminar los ataques conocidos puede usarse un sistema SIDS [3], lo que no garantiza la inexistencia de ataques en los datos resultantes. Por tanto, incluso en este caso se requiere de una supervisión posterior.

Por otra parte, en el caso de la detección de intrusiones en el servicio HTTP a partir del análisis de los URI, que es el escenario abordado en este trabajo, se plantea, adicionalmente, un problema práctico relacionado con la adquisición de los datos y la aplicación de los sistemas SIDS. Para poder aplicar los sistemas SIDS disponibles, p.e. Snort [6] o Suricata [7], se requiere de la captura del tráfico directamente de la red (en formato pcap o similar). Sin embargo, resulta mucho más eficiente, simple y escalable la obtención de los URI a partir de las trazas del servicio, que se encuentran en formato texto e incluyen los URI junto con información adicional. Se requiere, por tanto, de alguna herramienta que permita detectar ataques en estos archivos usando las firmas disponibles para los SIDS.

En el presente trabajo planteamos una metodología que puede utilizarse para minimizar la carga de trabajo asociada a la preparación de un conjunto de datos para su uso en el desarrollo de sistemas AIDS en el contexto del servicio

HTTP. Para ello se considerarán como punto de partida las trazas generadas por el servidor web correspondiente y se usarán características del servicio para facilitar la inspección manual de los datos. Por otra parte, como parte de esta metodología se ha desarrollado una herramienta que posibilita la detección y eliminación de ataques mediante el uso de las firmas disponibles en el formato utilizado por Snort.

La metodología propuesta se ha aplicado a la extracción de una base de datos a partir de las trazas del servidor web de la biblioteca de la Universidad de Sevilla. El elevado volumen de tráfico (197 días con más de 45 millones de peticiones web) imposibilita una supervisión manual del tráfico. Los resultados obtenidos evidencian la insuficiencia de la aplicación de un SIDS para la limpieza de los datos, habiéndose podido detectar ataques y artefactos adicionales mediante supervisión manual con una carga de trabajo reducida.

El resto del artículo se estructura como sigue. En el Apartado II se presenta una breve descripción del estado del arte en relación a la adquisición de datos, las metodologías y características que éstos deben reunir. En el Apartado III se presenta y justifica la metodología propuesta, que incluye el uso de un SIDS que opera directamente sobre las trazas y que ha sido desarrollado al efecto. Esta herramienta se describe en el Apartado IV. En el Apartado V se describe el caso de estudio y la aplicación de la metodología para la extracción de una base de datos de peticiones URI. Finalmente, en el Apartado VI se presentan las conclusiones y posibles líneas de mejora de la propuesta.

II. Estado del arte

La relevancia de las bases de datos en el desarrollo de los detectores de intrusiones basados en anomalías queda reflejada en numerosas publicaciones, en las que podemos identificar dos problemáticas principales. Por un lado, desde los inicios del campo de investigación se constató la necesidad de disponer de una base de datos de un tamaño suficiente y convenientemente etiquetada para poder entrenar y evaluar los sistemas, y que constituyese un referente para la comparación de los sistemas. DARPA'98 [8] surge así como un intento de establecer una base de datos común para la comunidad investigadora. Casi inmediatamente surgieron críticas a esta base de datos argumentando múltiples limitaciones [9] [10] y que apuntaban a la necesidad de establecer metodologías y técnicas para la adquisición de datos. En este sentido, podemos destacar varias propuestas, p.e. [3] [11] [12], en las que se establecen requisitos, características y procedimientos a seguir para la obtención de las bases de datos útiles. Entre las propiedades señaladas como más relevantes están la representatividad de los datos y su adecuado etiquetado [10] [11], esto es, la identificación del carácter normal o anómalo de los datos. Esto lleva a la consideración de que es necesario también establecer procedimientos para el etiquetado de los datos cuando éstos provienen de capturas de tráfico real. En este sentido, y dado que se requiere de volúmenes de tráfico significativos, el etiquetado manual de los datos resulta inviable. Para abordar este problema se pueden encontrar dos aproximaciones complementarias en la literatura. En la primera, se asume una muy reducida proporción de ataques en el tráfico real capturado y, por tanto se considera que el impacto de los ataques en el entrenamiento es despreciable o, alternativamente, que la presencia de ataques es irrelevante para el proceso de estimación, p.e. [13]. Esta primera aproximación puede ser adecuada en algunos modelados estadísticos o no supervisados, por lo que en la mayoría de los casos es habitual utilizar SIDS para detectar el mayor número posible de ataques [3], asumiendo la inexistencia de ataques en el conjunto de datos resultante. Complementariamente, se inyectan artificialmente instancias de ataques de forma controlada, lo que posibilita su etiquetado [11].

Otro aspecto menos abordado en la literatura es la necesidad de establecer un particionado de la base de datos [3] para su uso en cada una de las tres fases del desarrollo de un IDS: entrenamiento, evaluación y validación. Hemos de reseñar que, a pesar de la aparente trivialidad de esta necesidad, son numerosos los trabajos en los que se entrenan y evalúan diferentes alternativas, o se ajustan y comparan los modelos, a partir de la evaluación sin validación posterior, lo que es metodológicamente incorrecto y lleva al sobreajuste de los modelos. También son abundantes los trabajos en los que se entrena y evalúa con el mismo conjunto de datos.

Una cuestión adicional está relacionada con la especificidad de los datos y modelos en relación a los sistemas a proteger, lo que obliga a la adquisición de datos en el escenario concreto a modelar [1]. Es más, dada la naturaleza dinámica de los sistemas, cualquier captura de datos dejará de ser representativa del sistema transcurrido un cierto periodo de tiempo [14]. Este problema, denominado *data shift* en la literatura en inglés, establece requisitos en relación a los datos a usar para el test y validación de los sistemas [3], que deben incorporar eventos novedosos respecto de los utilizados para el entrenamiento.

III. Metodología supervisada

La metodología propuesta se plantea en un escenario de adquisición de datos a partir de las trazas de peticiones al servicio web. Su objetivo es la obtención de un conjunto de peticiones URI que no contenga ataques (o donde éstos estén identificados) con una intervención mínima por parte del supervisor del procedimiento. Para ello se propone el análisis de las trazas en 7 fases (Fig. 1) de las que únicamente 2 son supervisadas. Las fases son:

- Preprocesado/acondicionado: los archivos de traza son procesados para extraer la información de interés (p.e. URI, marca temporal, método), normalizando los diferentes campos y generando un registro por muestra (petición). En esta fase es de especial relevancia la aplicación de técnicas de normalización de URI.
- Filtrado: Selección de los registros que se considerarán en las fases posteriores de acuerdo a los criterios que se estimen oportunos (p.ej. método(s), ventanas temporales).
- Detección de ataques: Uso de un SIDS para la detección de ataques conocidos. Como resultado, se obtendrá un conjunto de registros considerados como ataques y un conjunto de registros "limpios" que serán procesados en las fases posteriores.
- Segmentación: Los URI de registros limpios se segmentan en sus diferentes campos, según los RFC correspondientes [15]. En esta fase se generarán los

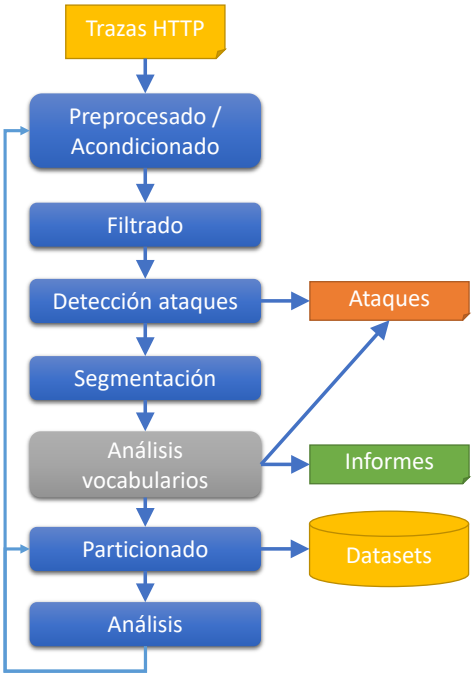


Fig. 1. Metodología propuesta.

vocabularios de *path*, atributo y valor, compuestos por las diferentes palabras que aparecen en cada uno de estos campos, incluyendo sus frecuencias de aparición.

- Análisis de vocabularios: Se analizan los archivos de vocabulario con la ayuda de diferentes histogramas para detectar anomalías en los mismos. Esta fase requiere de supervisión manual, generándose un informe con los datos de entrada y las observaciones
- Particionado: A partir de diversos criterios como periodo temporal, submuestreo o volúmen de datos, se establecen particiones disjuntas para su uso en entrenamiento, test y validación. En esta fase se puede considerar el uso posterior de técnicas como leave-k-out para la configuración de los diferentes experimentos.
- Análisis: Se extraen y comparan los vocabularios de las particiones para determinar la existencia de diferencias entre los mismos. En caso de no cumplirse los requisitos establecidos (p.e. porcentaje mínimo de palabras diferentes), se repetirá la fase anterior variando los criterios de particionado.

Hemos de señalar que como resultado del análisis de vocabularios es posible que se genere un listado adicional de registros de ataque o anómalos, que serán incorporados al conjunto correspondiente y descartados en las fases siguientes. También es posible que, en caso de no cumplirse los criterios establecidos para la base de datos resultante, se repita todo el procedimiento modificando algún criterio como la normalización o el filtrado.

A. Herramienta de detección de ataques

Los SIDS más habituales (p.ej. Snort, Suricata) operan sobre capturas en formato pcap, no estando disponibles en la actualidad herramientas que apliquen las firmas correspondientes sobre trazas en formato texto. Por tanto, para la detección de ataques a partir de las trazas de HTTP

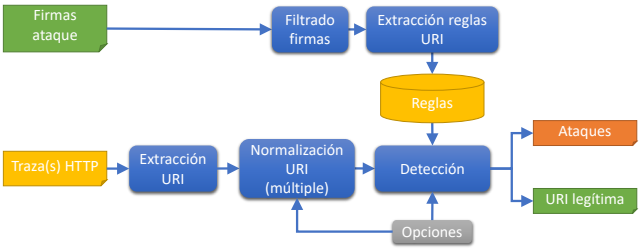


Fig. 2. Diagrama de bloques (simplificado) de *Inspectorlog*.

usando firmas de Snort ha sido necesario desarrollar una herramienta específica denominada *InspectorLog*.

InspectorLog toma como entradas un conjunto de firmas de ataque en el formato establecido para Snort y un archivo de trazas HTTP, generando a su salida dos listados: uno conteniendo los registros que han activado alguna regla, junto con detalles sobre las reglas activadas, y otro con los registros que no activan ninguna de las reglas, es decir, legítimos.

Las firmas se leen durante el proceso de inicialización de la herramienta, siendo filtradas para descartar las que no afectan al servicio HTTP (bloque filtrado de firmas en la Fig. 2). Posteriormente, estas reglas son analizadas para extraer las cadenas y/o expresiones regulares correspondientes que afecten únicamente a los URI. Por tanto, básicamente, la operación de *InspectorLog* consistirá en la detección de coincidencias entre todas las cadenas y expresiones regulares con cada uno de los URI analizados.

Dada la diversidad de formatos de las trazas según su origen y las posibles diferencias en la representación de los datos, se incluyen dos módulos previos a la detección. El primer módulo tiene como objeto la extracción de la URI a partir del archivo de traza, independientemente de su formato. En la versión actual se consideran dos de los formatos más comunes para las trazas utilizadas por Apache y listados de URI con algunas cabeceras. El segundo módulo está relacionado con la normalización de los URI, especialmente en relación al denominado *percent encoding* [16]. A pesar de establecerse un formato estándar para la representación de los caracteres en el estándar, son muchos los sistemas que no tienen en cuenta esta recomendación y, adicionalmente, algunos ataques utilizan esta representación como método de ocultación. Es más, durante el análisis de las reglas estándar de Snort hemos detectado incoherencias en relación al uso del *percent encoding*. Para solucionar estos problemas hemos establecido un procesamiento opcional que, en caso de detectar el uso de *percent encoding* en un URI, aplica todas las reglas, procede a decodificar los caracteres y vuelve a aplicar todas las reglas, repitiéndose el proceso mientras existan caracteres codificados. Una opción similar se contempla para el caso de mayúsculas/minúsculas en las reglas y en la codificación de los caracteres. En este caso, al activar la opción correspondiente se convierten todos los caracteres a minúscula para el procesamiento.

Los campos de las reglas considerados para el análisis son: *content, pcre, dsize, urilen* y *nocase*. De cada regla se almacenan también los campos *msg, reference, classtype* y *sid* para mostrarlos a la salida, según las opciones establecidas, con cada activación de la regla en el informe de detección correspondiente (Fig. 3).

```
# inspectorlog v3.0
------------------------ Initializing Rules ---------------------
Rules directory : "/media/reglas"
-------------------------- Statistics -----------------------------
Read [7146] rules, [7141] http-related, [0] with errors
--------------------------- Analysis results ----------------------------
#Alertas y firmas generadas: /bin/inspectorlog -l access_log-20170105-raw.uri -r /media/reglas -n -e -t uri
Packet [32149]  Uri [/xmlrpc.php?rsd]  Nattacks [1] Signatures  [SERVER-WEBAPP PHP xmlrpc.php post attempt - sid: 3827]
Packet [55705]  Uri [/imce?app=ckeditor%7Csendto%40ckeditor_imceSendTo%7C&CKEditor=edit-body-und-0-
value&CKEditorFuncNum=1&langCode=es]  Nattacks [1] Signatures  [SERVER-APACHE Apache mod_proxy reverse proxy information disclosure
attempt - sid: 20528]
Packet [112197]  Uri [/educacion/noticias/bibeducacion%40us.es]  Nattacks [1] Signatures  [SERVER-APACHE Apache mod_proxy reverse proxy
information disclosure attempt - sid: 20528]
```

Fig. 3. Ejemplo de informe de detección de ataques generado por *InspectorLog* (formato extendido).

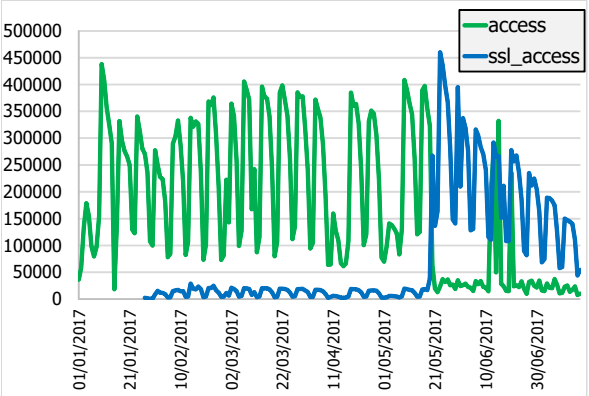


Fig. 4. Evolución temporal del número de peticiones.

B. *Herramientas de extracción del vocabulario*

Se ha desarrollado un conjunto de herramientas en forma de macros, programas en C y programas en *Python* que automatizan todas las operaciones a realizar para la aplicación de la metodología, incluyendo la selección de las reglas de interés a partir de las distribuciones oficiales. Como resultado se obtienen los diferentes listados de URI limpios, URI descartados, ataques, etc., así como los vocabularios y los histogramas de los mismos. El análisis de los vocabularios, a partir de los listados y los histogramas, es el único paso manual del proceso, para el que también se han desarrollado herramientas de apoyo para la selección de palabras candidatas y el filtrado posterior, en su caso, de los URI que incorporen elementos a descartar. Si se detectan nuevos ataques en la fase de análisis de vocabularios, éstos son incorporados a los detectados por *InspectorLog*.

IV. Caso de estudio

La metodología propuesta ha sido aplicada, entre otros, a la adquisición de trazas de varios servidores web de la Universidad de Sevilla. En particular, al servidor web del servicio de investigación (*inves* en adelante) y al de la biblioteca (*biblio*) de la universidad. A continuación, se describe la aplicación de la métodología a *biblio* y un resumen de los resultados del análisis de *inves*.

A. *Descripción de la base de datos biblio*

Las trazas suministradas por la biblioteca se organizan en varios bloques (Tabla I):

- *access_log*: Archivos correspondientes a las peticiones (método + URI) al servidor mediante http desde el 1/1/2017 hasta el 17/07/2017 organizados en un fichero por día (198 archivos).
- *ssl_access_log*: Idem mediante https desde el 27/1/2017 hasta el 17/07/2017 organizados en un fichero por día (172 archivos).
- *ssl_error*: Archivos de trazas correspondientes a mensajes de error desde el 17/1/2017 hasta el 17/07/2017 organizados en un fichero por día (185 archivos). Estos archivos no contienen URI.

La evolución temporal del número de peticiones (Fig. 4) muestra un cambio relevante en el servidor a partir del 20/05/2017, fecha a partir de la que se invierte la proporción entre el tráfico sin cifrar y el cifrado. Dentro de esta tendencia, se observan discrepancias puntuales (los tres picos correspondientes al 13/6, 15/6 y 20/6).

B. *Obtención de la base de datos*

La metodología propuesta tiene como objetivo la obtención de un conjunto de trazas limpias, es decir, libres de ataques de los servicios monitorizados y, en su caso, de un conjunto de ataques. Para ello, se extraerán los URI de las trazas y se filtrarán los ataques mediante las herramientas adecuadas. Tras el procesamiento automático, se supervisarán las trazas para la potencial eliminación de URI inadecuadas o de ataque que hayan podido superar las fases automáticas.

B.1. *Preprocesado y filtrado*

Se extraen las líneas que contengan URI y se normalizan los campos. En particular, se abordan algunos problemas de formato observados:

- Se eliminan las líneas cuyo campo URI tiene valor '*'.
- Se sustituyen las dobles barras al principio del *path* por una barra simple para evitar avisos de fuera de especificación.

En relación al filtrado, en una primera aproximación únicamente se aplica para la extracción de los URI correspondientes a los métodos de interés, esto es, GET, POST, HEAD y PROPFIND. Los resultados respecto del número de URI extraídos se muestran en la Tabla I.

Tabla I
Resumen de características de las trazas de biblio

Bloque	*Tamaño*	*N. arch.*	*N. líneas*	*N. URI*
access_log	4,15 GB	198	34 573 623	34 074 832
ssl_access_log	1,17 GB	172	13 328 700	13 328 163
ssl_error	1,72 GB	185	14 633 292	0

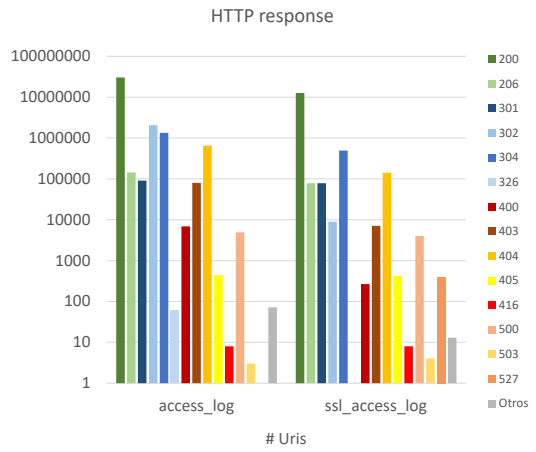


Fig. 5. Distribución de peticiones por código de respuesta.

Tras una primera iteración, se detectaron algunos problemas que se comentarán más adelante, y que hicieron modificar los criterios de filtrado. En particular, se consideró incluir un filtrado por código de respuesta, considerándose válidos solo los códigos de la serie 200 y 300. En la Figura 5 se muestra la distribución de peticiones por código.

B.2. Detección de ataques

Los ficheros resultantes se analizan mediante la herramienta *Inspectorlog* (v3.1), filtrándose aquellas líneas que generan alertas. Como resultado se generan archivos de informe de los ataques y archivos que se considerarán "limpios" en lo sucesivo.

Esta fase tiene un doble objetivo: detectar los posibles ataques existentes y generar la base de datos inicial que se utilizará para su procesamiento posterior (análisis de vocabularios, particionado, etc.). Por tanto, se consideran todas las reglas disponibles para limpiar las trazas (experimento EL) y una versión con reglas seleccionadas, para detectar los ataques (experimento DET). El uso de estos dos conjuntos de reglas se justifica por la posible aparición de falsos positivos como resultado de reglas excesivamente genéricas. Los resultados se muestran en la Tabla II.

Los conjuntos de reglas han sido obtenidos a partir de las reglas TALOS de Snort de número 2990 (denominadas *sn2990* en lo sucesivo) y las oficiales de Suricata del mes de junio de 2017 (*su-201707*) a partir de la selección de aquellas que afectan únicamente a las URI de peticiones HTTP. Para ello se han filtrado las reglas oficiales de Snort y Suricata seleccionando las que afectan a los puertos HTTP y que incluyen los campos relacionados con el contenido de la URI (*http_uri, uricontent, http_raw_uri* o *content* -en última instancia-), pero que en ningún caso incluyen campos relacionados con otros elementos de las cabeceras de HTTP.

Se ha obtenido la distribución de ataques por SID para cada archivo de traza y globales. En la Tabla III se muestran

Tabla II
RESULTADOS DE LA FASE DE DETECCIÓN DE ATAQUES (BIBLIO)

Bloque	*URI limpios*	*Conjunto EL*		*Conjunto DET*	
		N. at.	*URI con at.*	*N. at.*	*URI con at.*
access_log	32933343	88355	85695	13636	11678
ssl_access_log	13145253	28850	28676	2201	1158

Tabla III
ATAQUES MÁS FRECUENTES (TOP 10)

Frec	
6976	OS-WINDOWS Microsoft generic javascript handler in URI XSS attempt - sid: 20258]
1415	SERVER-APACHE Apache mod_proxy reverse proxy information disclosure attempt - sid: 20528]
785	SERVER-IIS Microsoft Windows IIS 6 multiple executable extension access attempt - sid: 21600]
767	ET WEB_SERVER Possible Microsoft Internet Information Services (IIS) .asp Filename Extension Parsing File Upload Security Bypass Attempt (asp) - sid: 2010592]
767	SERVER-IIS multiple extension code execution attempt - sid: 16356]
739	SERVER-WEBAPP PHP xmlrpc.php post attempt - sid: 3827]
670	SERVER-WEBAPP Setup.php access - sid: 2281]
649	SERVER-WEBAPP Checkpoint Firewall-1 HTTP parsing format string vulnerability attempt - sid: 2381]
439	ET WEB_SERVER Possible SQL Injection Attempt UNION SELECT - sid: 2006446]
436	SQL union select - possible sql injection attempt - GET parameter - sid: 13990]
785	SERVER-IIS Microsoft Windows IIS 6 multiple executable extension access attempt - sid: 21600]

los 10 más comunes. Como se puede observar, hay un elevado número de alertas asociadas a SID (reglas) que podemos considerar muy genéricos por lo que podrían ser falsos positivos dependiendo de las políticas de seguridad en vigor. Sin embargo, existe un número significativo de alertas que evidencian la existencia de instancias de ataques en las trazas. Es significativo indicar que la tasa de ataques (DET) es del 0,0271%.

B.3. Segmentación y análisis de vocabularios

Los archivos "limpios" obtenidos en la fase anterior se procesan para su segmentación en campos y la extracción de los vocabularios asociados a cada estado. Durante este proceso se descartan las secuencias que no cumplen las especificaciones (OOS). El analizador de URI utilizado sigue estrictamente las especificaciones del RFC 3986, con la salvedad de que no contempla la aparición de segmentos (#).

A continuación, se procede a la supervisión de los vocabularios y de los archivos generados.

- *Análisis de OOS*

Se han identificado 441 (142 diferentes) URI en *access_log* y 267 (49 diferentes) URI en *ssl_access_log* que no han podido ser segmentados (OOS). Una inspección visual de los mismos muestra que parecen responder a problemas de codificación de caracteres extendidos, p.ej. [*/educaci\xc3\xb3n*].

- *Análisis del vocabulario*

Los tamaños de los vocabularios por estado obtenidos se muestran en la Tabla IV. Para proceder a su análisis se obtienen los histogramas de las palabras en el vocabulario en función de los tamaños, tanto para el número total de palabras como para el número de palabras diferentes. Se pretende

Tabla IV
VOCABULARIOS POR ESTADOS

Bloque	*Path*		*Atributo*		*Valor*	
	N	*Dif*	*N*	*Dif*	*N*	*Dif*
access_log	>197 M	26251	>5 M	1537	> 3,5M	25229
ssl_access_log	>77 M	13349	> 1,7 M	1000	> 1 M	8763

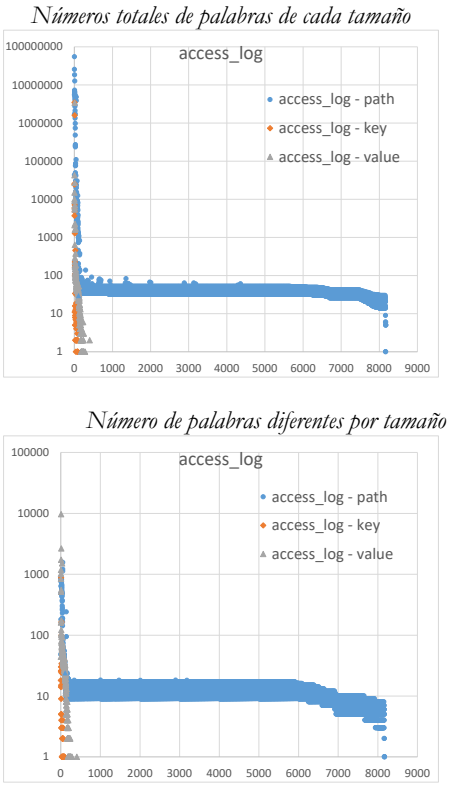


Fig. 6. Histogramas de los vocabularios para *access_log* en función de las longitudes de las palabras en la primera iteración.

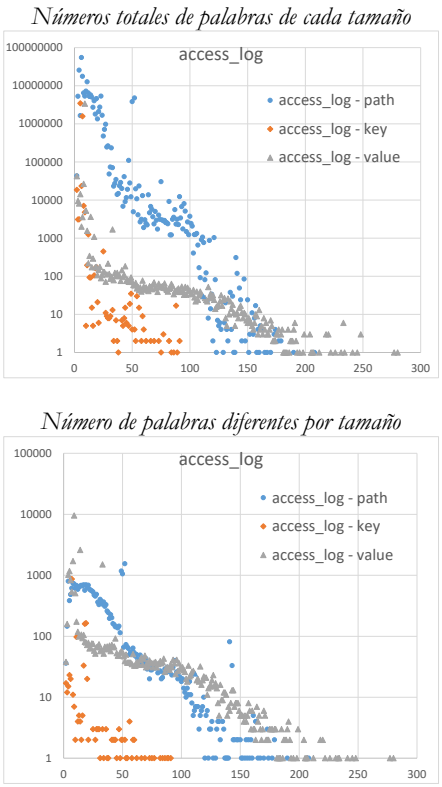


Fig. 7. Histogramas de los vocabularios para *access_log* en función de las longitudes de las palabras tras el filtrado por código.

determinar la existencia de palabras relevantes para un análisis posterior.

Los histogramas obtenidos en la primera aproximación en la que no se aplicó filtrado por código de respuesta evidenciaban anomalías en los datos. En particular, se identifica un comportamiento anómalo en los valores del *path*, especialmente en el bloque *access_log* (Fig. 6), apareciendo una cola anómala con longitudes de palabra excesivas. Se identifican en torno a 10 valores diferentes con alguna repetición para prácticamente todo el rango de valores de longitud. El efecto también se observa en el bloque *ssl_access_log*, aunque con menor incidencia. Un análisis manual muestra la existencia de cadenas incluyendo la secuencia *"%25252525...25"* insertada en valores existentes del path, y cuya longitud va aumentando progresivamente lo que constituye un ataque (posiblemente de denegación de servicio) relacionado con la codificación/decodificación de las URI. Se detecta que aparecen en ráfagas y que las peticiones consecutivas aumentan progresivamente la longitud de la petición, lo que también apunta a un intento de provocar un *buffer overflow*. A modo de ejemplo, se detecta la secuencia de peticiones (sólo se incluyen las primeras):

/salud/noticias/acceso-en-prueba-3-manuales-universitarios-de-salud-de-la-editorial-m%252525c3dica-panamericana
/salud/noticias/acceso-en-prueba-3-manuales-universitarios-de-salud-de-la-editorial-m%25252525c3dica-panamericana
/salud/noticias/acceso-en-prueba-3-manuales-universitarios-de-salud-de-la-editorial-m%2525252525c3dica-panamericana
/salud/noticias/acceso-en-prueba-3-manuales-universitarios-de-salud-de-la-editorial-m%252525252525c3dica-panamericana
/salud/noticias/acceso-en-prueba-3-manuales-universitarios-de-salud-de-la-editorial-m%25252525252525c3dica-panamericana

Tras una exploración de estos casos, y tras comprobar que aparece el código de respuesta HTTP en las trazas, se concluyó que era necesario filtrar las peticiones atendiendo al código de respuesta del servidor a fin de modelar correctamente el servicio. En particular, las peticiones con respuestas erróneas (códigos 4?? y 5??) deben ser descartadas del análisis durante el filtrado, salvo para identificar ataques existentes.

Una vez introducido el filtrado por código de respuesta, que elimina este problema, se procedió a filtrar las palabras del vocabulario mediante inspección manual en base a anomalías en el tamaño (histograma) o en el contenido (p.e. mediante la búsqueda de caracteres no permitidos o poco habituales). En la Fig. 7 se muestran los histogramas obtenidos para el bloque *access_log*, donde se identifican algunos casos que deben ser revisados.

Para facilitar la inspección, se obtienen los listados de palabras ordenadas por tamaño. Se observan abundantes cadenas con el formato *css_[*].css* y *js_[*].js*, cuyas longitudes están entre 48 y 52 caracteres. Por tanto, los picos en el *path* en torno a esta longitud se consideran normales.

Durante la inspección manual se observan *paths* con estructura que podría corresponder a *query* (incluyen *'&'* y *'='*), por lo que se extrae un listado de las mismas. Según la RFC 3986, los caracteres permitidos en el *path* son a-z A-Z 0-9 . - _ ~ ! $ & ' ( ) * + , ; = : @ / y los caracteres codificados mediante *%*. Por lo tanto, las cadenas son válidas desde el punto de vista sintáctico. Sin embargo, es evidente que algunas de estas cadenas corresponden a la sustitución del delimitador de *query '?'* por *'&'*, por lo que se eliminarán las

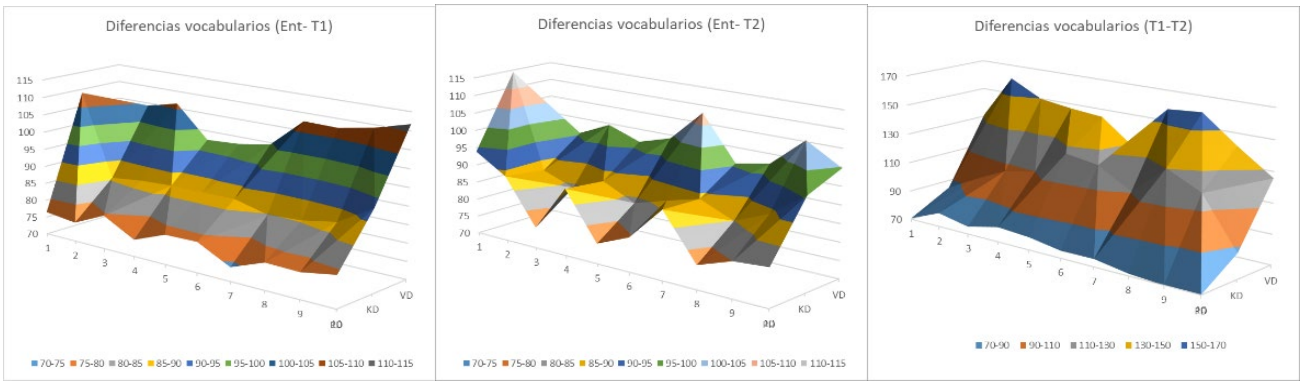


Fig. 8. Diferencias porcentuales en los tamaños de los diccionarios de entrenamiento (ENT) y test (T1 y T2) para las diferentes configuraciones de entrenamiento.

URI correspondientes. También se identifica el uso de *&[]acute*; y análogos *(<)*, que también serán eliminados.

Se procede de forma análoga para *key* y *value*. Si se analiza el vocabulario final, encontramos valores inadecuados (incluyen delimitadores). Sin embargo, la búsqueda de los mismos en los URI da resultado negativo en la mayoría de los casos. Esto es debido a que se encuentran codificados mediante *percent encoding* y éste es eliminado tras el segmentado. Por tanto, se consideran correctos. En el caso de *value*, se obtienen los listados de palabras que incorporan caracteres no permitidos o anómalos y se procede a la inspección de los mismos, determinándose que, en la mayor parte de los casos, son valores correctos por encontrarse adecuadamente codificados. Se eliminarán algunos casos en los que, aparentemente, se usa ";" como separador (p.e. *C=N;O=D*), algunos casos anómalos (*1'A=0* y la aparición de *<a href* en el *path*, que se detecta a raíz de este análisis).

B.4. Particionado y análisis

Establecemos un primer particionado de la base de datos a partir de la combinación por días de los archivos correspondientes a ambos servicios (*access_log* y *ssl_access_log*) y la posterior agrupación en bloques de 30 días. Se obtienen así 7 bloques, que se etiquetan mediante números del 1 al 7. Se reservan los bloques 6 y 7 para validación, mientras que los restantes se combinan siguiendo un esquema de validación cruzada *leave-2-out*. Se generan, por tanto, 10 conjuntos de entrenamiento diferentes, cada uno compuesto por 3 de las 5 primeras particiones.

Como se ha comentado previamente, un aspecto relevante de los conjuntos de entrenamiento y test está relacionado con la inclusión de elementos novedosos, a fin de poder evaluar el comportamiento del sistema ante el problema de entrenamiento insuficiente y los cambios originados por el dinamismo del sistema modelado (*data shift*). A este fin la metodología propuesta incorpora un análisis de las particiones y combinaciones a utilizar en entrenamiento y test que permita evaluar los solapamientos y diferencias en los vocabularios. Para ello se generan los vocabularios asociados a cada partición y se comparan sus tamaños tanto en términos de número de palabras diferentes como de sus frecuencias de observación. En la Fig. 8 se muestran gráficamente las diferencias porcentuales en los tamaños de los vocabularios. Se observan diferencias porcentualmente significativas entre los vocabularios de entrenamiento y test, lo que es de interés para la correcta evaluación del sistema. Se considera, por tanto, que no es necesario revisar el criterio de particionado.

C. Aplicación a la base de datos inves

La misma metodología ha sido aplicada a las trazas obtenidas a partir del servidor del servicio de investigación de la US. En este caso se dispone de 31 archivos correspondientes al periodo del 1 al 31 de mayo de 2018. El volumen total de datos es de 4,62 GB con más de 15 millones de líneas. Tras la detección de ataques, se extraen 14071622 URI considerados limpios, habiéndose detectado 79874 y 4382 ataques, respectivamente, en las fases EL y DET. Se determina, por tanto, una tasa de ataques de 0,023% (similar al caso de *biblio*).

Los histogramas de los vocabularios obtenidos (Fig. 9) muestran valores (*value*) con longitudes excesivas (mayores de 300), con un número relativamente alto de repeticiones pero que corresponden a una o dos palabras diferentes, por lo que deben ser examinados. También se identifican algunos puntos que podrían interpretarse como anomalías y que serán revisados.

Durante la inspección manual se detectan numerosos artefactos generados por un aparente truncamiento de algunos campos/valores, un uso inadecuado de los delimitadores y el uso de ';' como delimitador de campos. También se detectan los mismos artefactos que en el caso de *biblio* respecto de paths cuya estructura responde a *query* y el uso de *&[]acute;* y análogos (*< &amp*). También se detectan algunos valores que parecen ataques (p.e. *user%2Fpassword&name%5B%23post_render%5D%5B%5D=passthru&name%5B%23type%5D=markup&name%5B%23markup%5D=wget+https%3A%2F%2Fraw.githubusercontent.com%2FRxR-HaCkEr%2Fdrupal%2Fmaster%2Fd7.php)*. En relación a los valores con longitudes elevadas, se determina que se debe a la concatenación de valores de búsqueda, siendo correctas todas ellas.

V. Conclusiones

El desarrollo de AIDS requiere de conjuntos de datos para entrenar, evaluar y validar los sistemas en condiciones análogas a su explotación real. La generación de estos conjuntos de datos resulta costosa ya que implica su etiquetado, lo que requiere de una supervisión manual de todos los datos. En este trabajo se plantea una metodología

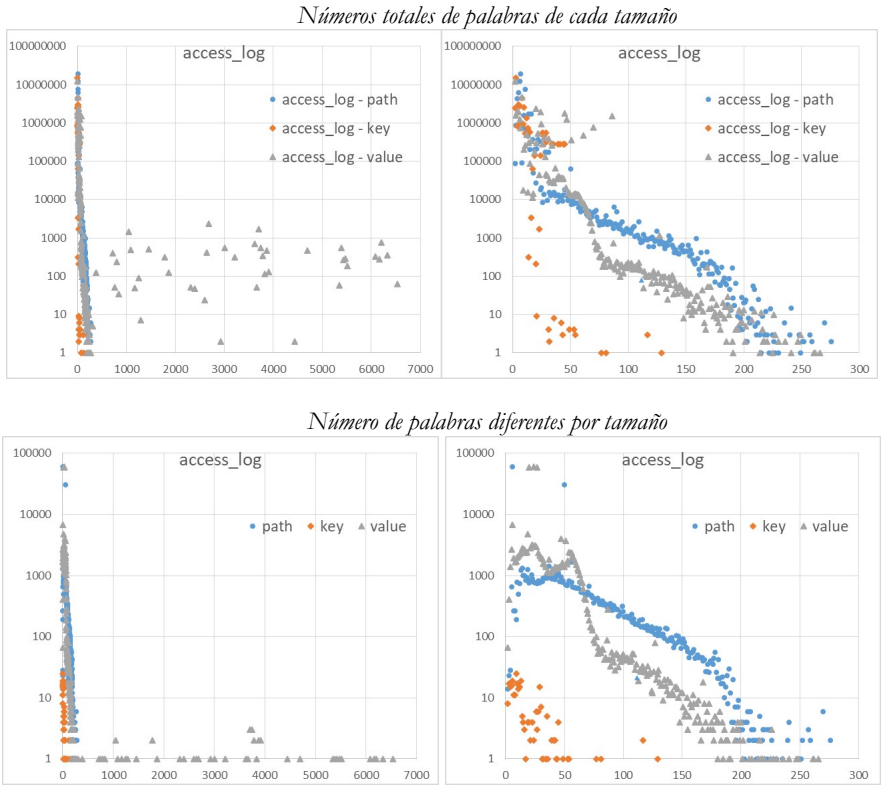


Fig. 1. Histogramas obtenidos para la base de datos *inves* (izda) y detalle hasta longitud 300 (dcha.).

que, utilizando las características de las peticiones HTTP, permite automatizar en gran medida el proceso de generación de estas bases de datos, simplificando y reduciendo los datos a supervisar.

La metodología ha sido aplicada con éxito a dos bases de datos diferentes, habiéndose procesado un elevado número de peticiones (decenas de millones) en un tiempo reducido (inferior a un día). Se han detectado artefactos y ataques en los resultados obtenidos tras la aplicación de un SIDS para filtrar el tráfico, lo que constituye la aproximación más habitual en estos casos. Adicionalmente, durante la aplicación de la metodología a uno de los casos de estudio se han determinado modificaciones en la misma que mejoran los resultados.

AGRADECIMIENTOS

El presente trabajo ha sido parcialmente subvencionado por los proyectos CTA 16/909, PI-1786/22/2018 y PI-1736/22/2017.

REFERENCIAS

[1] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez; Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges", Computers & Security, 28:18-28, 2009.

[2] C. Sanders, J. Smith, Applied Network Security Monitoring, Syngress 2014. ISBN: 978-0-12-417208-1.

[3] M. Bermúdez-Edo, R. Salazar-Hernández, J. Díaz-Verdejo, P. García-Teodoro, "Proposals on Assessment Environments for Anomaly-Based Network Intrusion Detection Systems", Lect. Notes in Computer Science, 4347, pp. 210-221, 2006.

[4] R. Sommer, V. Paxson; Outside the Closed World: On Using Machine Learning For Network Intrusion Detection, Proc. IEEE Symp. On Security and Privacy, 305-316, 2010.

[5] G. Cretu, A. Stavrou, M. Locasto, S. Stolfo, A. Keromytis, Casting out Demons: Sanitizing Training Data for Anomaly Sensors, Proc. IEEE Symposium on Security and Privacy, 81-95, 2008.

[6] http://www.snort.org

[7] http://suricata-ids.org

[8] MIT Lincoln Labs. (2008, Feb.). DARPA intrusion detection evaluation [Online]. Disponible en: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html.

[9] J. McHugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory", in ACM Transactions on Information and System Security, 3(4):262-294, 2000.

[10] M. Tavallaee, E. Bagheri, W. Lu, A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, Proc. IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009.

[11] I. Sharafaldin, A. H. Lashkari and A: A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", Proc. Int. Conf. on Information Systems Security and Privacy (ICISSP 2018), pp. 108-116, 2018.

[12] M. Tavallaee, N. Stakhanova, A. Ghorbani, "Toward Credible Evaluation of Anomaly-Based Intrusion-Detection Methods", IEEE Transactions on Systems, Man, And Cybernetics —Part C, 40(5):516-524, 2010.

[13] S. Zanero, S. Savaresi, Unsupervised Learning Techniques for an Intrusion Detection System, Proc. SAC, 2004.

[14] Data shJ. Quiónero-Candela, M. Sugiyama, A. Schwaicghofer, N. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009, IBSN 978-0-262-17005-5.

[15] T. Berners-Lee, R. Fielding, L. Masinter, Uniform Resource Identifiers, RFC2396, 1998

[16] T. Berners-Lee, R. Fielding, L. Masinter, Uniform Resource Identified (UIR): Generic Syntax, RFC3986, 2005.

Extracción de conocimiento a partir de fuentes de datos reales procedentes de la monitorización de eventos de seguridad

Alberto Bravo Gómez
Universidad de Extremadura
Escuela Politécnica
Av. Universidad s/n – Cáceres
albertobg@unex.es

José Carlos Sancho Núñez
Cátedra Viewnext-UEx
Escuela Politécnica
Av. Universidad s/n - Cáceres
jcsanchon@unex.es

Andrés Caro Lindo
Universidad de Extremadura
Escuela Politécnica
Av. Universidad s/n – Cáceres
andresc@unex.es

Resumen- **La monitorización de eventos de seguridad es una práctica cada vez más utilizada por las organizaciones, para detectar amenazas, vulnerabilidades y estimaciones de riesgos de seguridad. La gestión de eventos e información relacionada con la seguridad se realiza mediante sistemas comerciales que facilitan toda la información, procesando diferentes fuentes de datos. La posibilidad de desarrollar modelos alternativos que, en base a las mismas fuentes de datos, proporcionen información complementaria a los sistemas comerciales se plantea como un reto novedoso e interesante, no solo para las organizaciones, sino también para la comunidad científica. Este artículo presenta un novedoso sistema de extracción de conocimiento basado en la monitorización de eventos de seguridad que permite complementar la información de los sistemas comerciales y también predecir conductas futuras de riesgo que permitan anticiparse a situaciones de posibles riesgos.**

***Index Terms*- SIEM, Ciberseguridad, STRIDE, Extracción de conocimiento, Procesamiento de datos.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La Gestión de Eventos e Información de Seguridad (*Security Information and Event Management - SIEM*) está cada vez más implantada en las organizaciones, debido a la importancia que en los últimos años ha adquirido la seguridad en los Sistemas Informáticos. Estos sistemas proporcionan información muy útil sobre eventos relacionados con la seguridad y con potenciales amenazas de riesgos y vulnerabilidades, ayudando en la detección de conductas inusuales, generando alertas, y siendo capaces de monitorizar e incluso predecir comportamientos futuros.

La Gestión de Eventos de Seguridad (SEM) tiene que ver con la monitorización y correlación de eventos de seguridad en tiempo real, mientras que la Gestión de Información de Seguridad (SIM) procesa esos datos, almacena, analiza y genera reportes.

Muchas organizaciones disponen de sistemas SIEM basados en conocidas soluciones comerciales, como pueden ser QRadar, de IBM [1], Arc Sight, de HP [2] o soluciones alternativas a estas grandes corporaciones, como Symantec Security Services [3], McAfee SIEM [4], Alien Vault [5] o FortiSIEM de Fortinet [6].

Cuantas más fuentes de datos alimentan un SIEM, y cuanto más heterogéneas sean éstas entre sí, más posibilidades de éxito tendrá la implementación del SIEM. No solo a la hora de detectar y responder ante amenazas, sino también, sumando las posibilidades que aportan la gestión masiva de datos, incluso anticipando posibles riesgos. Las tareas de predicción se plantean, pues, como un reto y una nueva modalidad de aportar conocimiento a partir de estados actuales.

Cuando los datos provienen de múltiples fuentes es fundamental gestionar adecuadamente el proceso de extracción, transformación y carga (*Extract, Transform and Load - ETL*). Este proceso es crítico, ya que es muy posible que se utilicen varias soluciones, y desde diferentes perspectivas de seguridad.

Un punto interesante que ofrece el formateo inicial de los datos, y que se ha considerado en este artículo, es aprovechar esta etapa de preprocesado de datos para categorizar las amenazas según el modelo STRIDE [7]. STRIDE permite clasificar amenazas en seis categorías: ***S***uplantación de identidad, ***T***ampering o manipulación de información, ***R***epudio, divulgación de ***I***nformación, ***D***oS y ***E***scalada de privilegios.

Otro aspecto importante tiene que ver con la posibilidad diseñar una metodología que genere conocimiento complementario a la información obtenida por el propio SIEM, basándose propiamente en la información de estas múltiples fuentes de datos. Este conocimiento podría utilizarse para complementar la información base obtenida por el SIEM. Y mucho más atractivo resultaría que el conocimiento obtenido se oriente a comportamientos futuros (predicción de actuaciones). Las técnicas de minería de datos pueden ayudar mucho en este sentido.

Este artículo se fundamenta en datos reales obtenidos del SOC (Centro de Operaciones de Seguridad) de Viewnext, que han sido conveniente anonimizados para usarse en el diseño experimental del sistema propuesto en este artículo. La empresa se encuentra formada por un equipo de más de 4.500 profesionales especializados en el desarrollo de software. Distribuida de manera descentralizada en varias oficinas y centros de innovación tecnológica ubicados en España y Portugal.

Como consecuencia de los trabajos realizados, en este artículo se presenta un completo sistema de extracción de conocimiento a partir de datos SIEM. Es la primera investigación de extracción de conocimiento relacionada con eventos de sistemas SIEM. Se persigue una doble finalidad: (i) completar y complementar la monitorización proporcionada por el SIEM de la organización; y (ii) predecir conductas futuras que permitan la anticipación a potenciales situaciones de riesgo, con una precisión bastante fiable.

II. Sistema de Extracción de Conocimiento SIEM

El sistema propuesto de extracción de conocimiento a partir de datos SIEM se muestra en la *Fig. 1*.

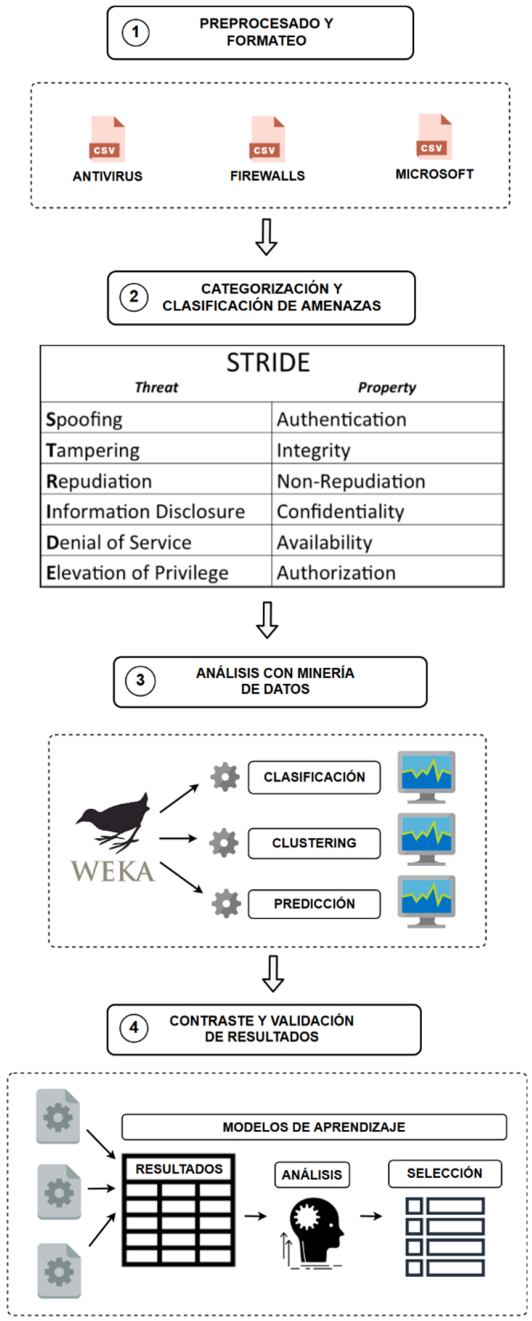


Fig. 1. Esquema del sistema de extracción de conocimiento SIEM dividido por fases.

Como puede apreciarse, el sistema se fundamenta en cuatro fases fundamentales que se explican a continuación.

III. Fase 1: Preprocesado y Formateo de Datos.

Para la fase experimental, se consideran tres fuentes de datos diferentes, heterogéneas en cuanto a contenidos, finalidades y objetivos. Estas fuentes de información proceden del SIEM QRadar de IBM, el cual se encarga de centralizar y monitorizar diferentes sistemas y dispositivos informáticos, gestionados por el SOC de Viewnext, como se ha mencionado anteriormente. Según esta compañía, un sistema SIEM instalado en una infraestructura de una empresa media genera 1000 eventos por segundo. Esto supone aproximadamente 86 millones de eventos diarios.

Las fuentes de datos facilitadas por el SOC de Viewnext se corresponden con datos reales, que fueron convenientemente anonimizados y tratados exclusivamente con fines de investigación. Cada uno de estos tres archivos de datos constan de 10.000 tuplas, número que se entiende suficiente para los fines de investigación que se persiguen en este trabajo. A continuación, se comenta brevemente el contenido de cada archivo:

- ***Microsoft***: Archivo con información sobre servidores de Windows que están siendo monitorizados. De sus 9 columnas, se destacan las siguientes:
  - ***EventID:*** Código del evento que se ha detectado.
  - ***OriginatingComputer:*** IP del equipo afectado.
  - ***EventType:*** Código del tipo de evento producido.
  - ***EventCategory:*** Código de la categoría del evento.
- ***Firewalls***: Archivo con información sobre el cortafuegos o firewall utilizado, concretamente FortiGate [8]. De sus 29 columnas, se destacan las siguientes:
  - ***devid:*** Identificador del dispositivo afectado.
  - ***logid:*** Identificador del archivo de log del evento producido.
  - ***eventtype:*** Tipo de evento producido e identificado.
  - ***subtype:*** Subtipo del evento producido.
  - ***level:*** Nivel de severidad del evento producido.
- ***Antivirus***: Archivo con información sobre los eventos capturados por el antivirus utilizado, concretamente Symantec Endpoint Protection de la empresa Symantec. En la cuarta fase de validación de resultados se muestran los campos más significativos sobre este archivo.

Todos los archivos se encuentran en formato ***CSV*** y con una subestructura interna denominada SYSLOG [9] asignada por el SIEM (ver *Fig. 2*).

Fig. 2. Transformación de fuentes de datos RAW en ficheros csv con formato Syslog interno asignado por IBM QRadar.

El preprocesado y formateo de datos procedentes de estos tres archivos se fundamenta en un proceso semiautomático que realiza los siguientes pasos:

1. Detección y borrado del contenido considerado como prescindible, relativo a valores o campos con información poco útil o sin relevancia para el estudio.
2. De los tres archivos preprocesados, dos de ellos (*Antivirus* y *Firewalls*) presentaban irregularidades en su contenido, como diferentes tipos de filas o vectores de log, porque no todas las filas poseían el mismo número de columnas. Así, el preprocesado implementado consigue que tengan un

contenido unificado.

3. Tras ello, el preprocesado realiza una serie de mapeos o modificaciones en la información que presentan. Por ejemplo, columnas que almacenan valores del tipo "IP_Source: 129.345.0.128" se mapean a columnas denominadas "IP_Source", que almacenan los valores especificados. Se puede resumir este proceso como construcción de cabeceras y extracción de valores.
4. Por último, se realiza un proceso de validación que garantice el correcto formateo de cada valor. Esto se realiza con la librería *pandas* implementada en Python.

De este modo, se generan archivos de salida en la extracción de datos con el mismo tipo que los de entrada (*CSV*), ya correctamente formateados para ser analizados.

IV. Fase 2: Categorización y clasificación de amenazas.

La segunda etapa del sistema propuesto consiste en realizar una clasificación de amenazas, realizada de forma automática.

Para ello, el sistema desarrollado realiza una búsqueda sobre diferentes patrones, métodos y estándares que permiten la detección de amenazas [10]. Una de las principales aportaciones del sistema propuesto es el desarrollo de una nueva metodología para clasificar las amenazas en base al conjunto de datos procesado. Esta clasificación se realiza siguiendo el modelo STRIDE [7], que, como se ha comentado, categoriza las amenazas en seis grupos: ***S***uplantación de identidad, ***T***ampering o manipulación de información, ***R***epudio, divulgación de ***I***nformación, ***D***oS y ***E***scalada de privilegios.

Se han tenido en cuenta diferentes implementaciones y estudios realizados en torno a la categorización de amenazas con STRIDE [11][12][13][14][15], particularizando una versión propia adaptada al sistema propuesto.

Entre las variantes del modelo STRIDE [7] destacan dos principalmente:

- STRIDE por Elemento: este modelo se aplica cuando se dispone de un diagrama sobre una plataforma en el que se identifica cada elemento potencialmente vulnerable y se categoriza.
- STRIDE por Interacción: este modelo se concentra en las interacciones vulnerables en un sistema, que son finalmente categorizadas. Esta variante considera el mismo número de posibles amenazas que la anterior, pero desde otra perspectiva.

En el modelo propuesto se considera la segunda variante, ya en que los conjuntos de datos de que se dispone solo se tiene conocimiento sobre las amenazas o posibles vulnerabilidades producidas, pero no sobre los sistemas o plataformas afectadas. Por ello, no es posible aplicar la primera variante.

En la *Fig. 3* se ilustra el proceso seguido para la clasificación de las amenazas.

A. *Identificación de las amenazas*

El sistema implementado identifica amenazas en los archivos considerados, analizando campos concretos para identificar y clasificar amenazas:

- ***Antivirus.csv***: Se utiliza el identificador de los posibles eventos identificados (sid). El siguiente es un ejemplo de su formato y valor: [SID: 21331]. El número de eventos diferentes identificados en el archivo asciende a 62.
- ***Firewalls.csv:*** Se utiliza el identificador de log (logid) que clasifica sus eventos. Como ejemplo de su formato y valor se tiene: [logid: 4190]. El número de eventos diferentes identificados en este archivo es 10.
- ***Microsoft:*** Se utiliza el identificador del evento (eventid) que clasifica sus eventos. Un ejemplo de su formato y valor sería: [EventID: 4624]. Se identifican 27 eventos diferentes en este archivo.

Con los campos anteriormente especificados, el algoritmo propuesto agrupa los eventos por los valores únicos de cada campo seleccionado. De esta manera, no se clasificarán eventos repetidos ni duplicados para fases posteriores.

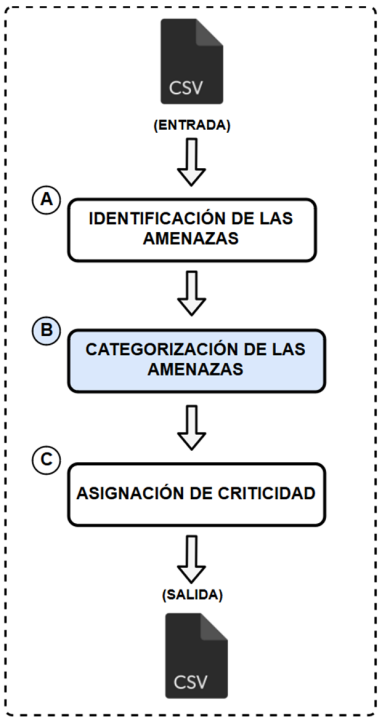


Fig. 3. Procesos en la etapa de Clasificación de Amenazas. El proceso de categorización (tarea 2, mostrado en azul) se realiza acorde al definido por Adam Shostack en [7].

B. *Categorización de las amenazas*

El sistema propuesto realiza la categorización de las amenazas según el proceso descrito por Adam Shostack [10]. Como resultado de este proceso de categorización, a cada evento identificado se le asigna una de las categorías que se cumplen en STRIDE. La *Tabla I* muestra un ejemplo del resultado de la categorización de amenazas.

Tabla I:
EJEMPLO DEL RESULTADO DE LA CATEGORIZACIÓN DE AMENAZAS

Amenaza	S	T	R	I	D	E
SMB Double Pulsar Ping	X	X		X		X
Fake TechSupport Domains	X	X		X		
Microsoft Windows SMB Remote Code	X	X	X	X		X
Microsoft SMB Disclosure Attempt	X	X	X	X	X	X

C. Asignación de Criticidad

Tras la categorización de amenazas, el modelo propuesto establece un sistema de puntuación para determinar el grado de criticidad. De este modo, el sistema se encarga de asignar a cada amenaza su impacto, en función a los valores establecidos en la *Tabla II*. Puede apreciarse que el valor de criticidad asignado (Alto, Medio o Bajo) se establece en función del impacto determinado en STRIDE.

Tabla II:
IMPACTO DE LAS AMENAZAS CON SU CRITICIDAD DESIGNADA

IMPACTO STRIDE	CRITICIDAD
1 – 2	Bajo
3 – 4	Medio
5 – 6	Alto

V. Fase 3: Análisis de datos.

El análisis de datos se realiza mediante la herramienta Weka [16] para completar la etapa de extracción de conocimiento. Este entorno permite la ejecución de algoritmos y modelos de aprendizaje de forma sencilla y flexible.

A. Clasificadores basados en Modelos de Aprendizaje

De entre todos los algoritmos de clasificación y predicción disponibles, se han realizado pruebas experimentales en base a los más habituales en la literatura científica. A continuación, se describen brevemente los algoritmos y modelos empleados:

a. ***Árboles de Decisión:*** Los árboles de decisión se forman mediante construcciones lógicas que *categorizan* los datos de entrada. De esta manera se ofrece el mejor resultado considerado en el diagrama para el conjunto de datos concreto. Los árboles de decisión seleccionados en los experimentos son los siguientes:
- ***J48/C4.5*** [17].
- ***Random Forest*** [18].
- ***Random Tree:*** Construye un árbol que considera K atributos elegidos al azar en cada nodo. No realiza podas. También tiene una opción para permitir la estimación de las probabilidades de clase (o media objetivo en el caso de regresión) en base a la estimación de backfitting [19].

b. ***Modelos Bayesianos:*** Son modelos basados en el teorema de Bayes, que se emplea para actualizar o inferir la probabilidad de que una hipótesis sea cierta. Los modelos concretos seleccionados son los siguientes:
- ***Naive Bayes*** [20].
- ***Bayes Net*** [21].

c. ***Máquinas de Vectores de Soporte (SVM):*** Se trata de modelos relacionados con problemas de clasificación y regresión, que emplean hiperplanos relacionados con el *vector soporte*. Aunque las SVM están relacionados con las redes neuronales, el entrenamiento de éstas últimas es más costoso que en SVM. En ambos casos se obtienen clasificaciones muy eficientes y se estima que funcionan muy bien en problemas como el planteado en esta investigación. Sin embargo, en los experimentos desarrollados, se ha optado por explorar el funcionamiento de modelos SVM y no el de redes neuronales. Por ello, el modelo seleccionado es:
- ***LibSVM*** [22].

d. ***Agrupamiento o Clustering:*** Comprende modelos de aprendizaje automático basados en vecindad. Las agrupaciones de los ejemplos se basan en conceptos de vecindad o proximidad que permiten medir distancias con diferentes métodos: Distancia Euclídea, distancia de Manhattan o distancia de Chebychev entre otros. De este modo es posible saber la similitud entre dos instancias o individuos, que será la determinada en función de su distancia. Como algoritmo, se utiliza:
- ***SimpleKMeans*** [23].

B. Modo de ejecución de los Modelos de Aprendizaje

Tras seleccionar los clasificadores para esta etapa de extracción del conocimiento, se establece el modo en el que serán ejecutados. Se selecciona *Cross-validation*, que consiste en realizar una validación cruzada estratificada del número de particiones dado (*Folds*). En la validación cruzada, dado un número *n,* se dividen los datos en *n* partes y, por cada parte, se construye el clasificador con las $n-1$ partes restantes realizando la ejecución con esa partición y de la misma manera con las restantes.

VI. Fase 4: Validación de resultados.

Esta fase de contraste y validación de resultados se realiza sobre los 3 archivos analizados (*Antivirus*, *Firewalls* y *Microsoft*). Debido a la limitada extensión de este artículo, se mostrarán únicamente los resultados obtenidos en el archivo *Antivirus*. No obstante, se desea hacer constar que los resultados obtenidos son muy similares en los otros 2 archivos analizados *(Firewalls y Microsoft)* y la discusión y conclusiones son extrapolables a todos los conjuntos de datos.

El fichero de *Antivirus* contiene 10.000 amenazas (tuplas), aunque finalmente se utilizan 6672 del total, tras eliminar los falsos positivos detectados, para mejorar la precisión de los análisis realizados.

Como campos de entrada en todos los análisis realizados se utilizan los siguientes, siendo diferente la clase en base a la que se clasifican:
- ***Descripción***: Especifica la descripción de cada ataque. Este campo ha sido adaptado para el análisis, limitando su contenido y agrupándolo con descripciones con el mismo significado.
- ***Protocolo***: Especifica el protocolo de acción utilizado (TCP, UDP, etc.).
- ***Aplicación***: Nombre de la aplicación afectada por el ataque.
- ***Puerto Local***: Número del puerto local del equipo afectado.
- ***Amenaza***: Especifica el nombre de la amenaza que ha sido producida.

A. Análisis sobre la comparativa de criticidades

Se establece un análisis para clasificar las amenazas en base a las dos criticidades existentes:
- ***Antivirus****:* Se corresponde con la criticidad o severidad establecida por el proveedor de antivirus, ya que otorga a cada amenaza el grado de importancia que considera. Está compuesto por tres niveles de menor a mayor importancia: *Low, Medium o Moderate y High* [24].
- ***STRIDE****:* Es la medida del impacto que cada amenaza posee basada en el sistema establecido en la *Tabla II.*

La distribución de los datos para cada una de las dos criticidades sobre el conjunto de datos se muestra en la *Fig. 4*.

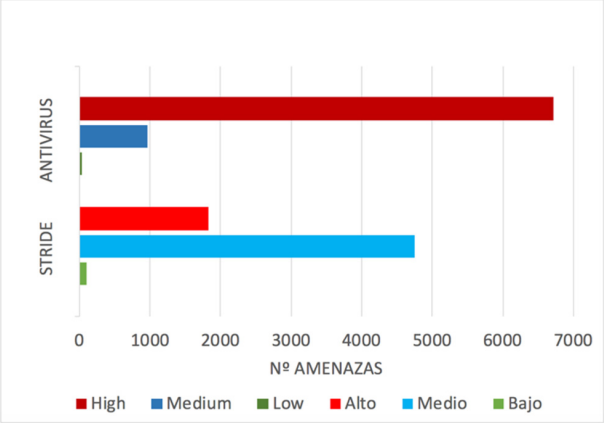


Fig. 4. Distribución en base a los niveles de las criticidades (proveedor del Antivirus y propuesta en este trabajo en base a STRIDE).

B. *Discusión de resultados sobre las criticidades*

En la *Tabla III* se aprecia que los resultados obtenidos en las medidas de ambas criticidades son muy similares. La capacidad de predicción o *ICC* (Instancias Correctamente Clasificadas) obtiene una ligera mejora con la nueva criticidad de STRIDE propuesta (*Fig. 5*). El modelo propuesto también mejora las Instancias Incorrectamente Clasificadas (*IIC*). El estadística *Kappa* [25], que determina la precisión del modelo a la hora de predecir la clase verdadera, también es mejor en el modelo propuesto, así como el Error Medio Absoluto (EMA).

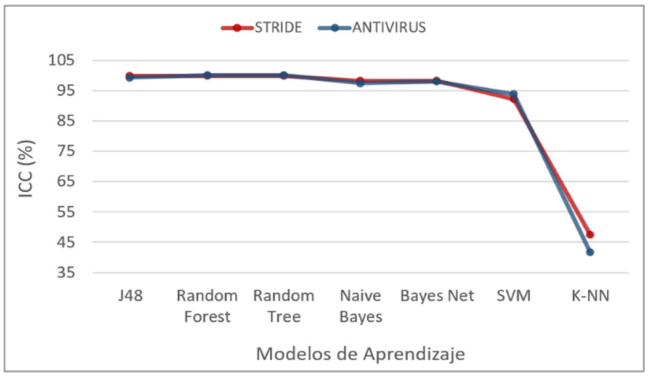


Fig. 5. Efectividad entre la criticidad propuesta (STRIDE) y la del proveedor de Antivirus.

C. *Análisis sobre la clasificación de categorías*

Tanto en la *Tabla III* como en la *Tabla IV* se aprecian valores muy bajos en los resultados de modelos basados en SVM y clustering. En consecuencia, se descartan:

- Los modelos basados en clustering, debido a que obtienen los peores resultados con respecto al resto de modelos.
- Los modelos basados en SVM, ya que consumen una gran cantidad de tiempo en su ejecución (son los más costosos computacionalmente) sin mejorar los resultados de los otros modelos.

La distribución de los datos para cada una de las 6 categorías de STRIDE se muestra en la *Fig. 6*.

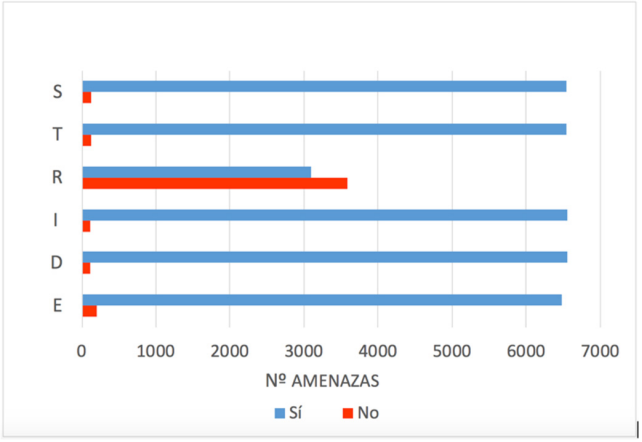


Fig. 6. Distribución de los datos del proveedor de Antivirus que cumplen con las categorías de STRIDE indicadas.

Finalmente, los resultados del análisis de todas las categorías se muestran de forma conjunta en la Tabla IV.

D. *Discusión de resultados sobre las categorías*

En la *Tabla IV* puede comprobarse que existe una gran similitud en los resultados obtenidos entre las diferentes categorías, lo cual indica una cierta homogeneidad entre estas categorías. Se destacan los siguientes aspectos:

- La distribución de datos de la mayoría de las categorías no se encuentra muy balanceada o equilibrada. Esto provoca que las estadísticas Kappa sean generalmente inferiores al valor medio de 0,5, indicando una baja concordancia.
- Por otra parte, los resultados referentes al ICC son muy altos en la mayoría de modelos, al igual que sucedía en la comparativa de la *Tabla III*.

Tabla III:
COMPARATIVA DE LAS CRITICIDADES EN BASE A LA CLASIFICACIÓN DE STRIDE Y AL PROVEEDOR DE ANTIVIRUS.

Criticidad	Medidas	J48	Random Forest	Random Tree	Naive Bayes	Bayes Net	SVM (libSVM)	K-NN (Clusters = 2)
STRIDE	ICC (%)	99,84	99,82	99,84	98,10	98,20	92,19	47,42
	IIC (%)	0,16	0,18	0,16	1,90	1,80	7,81	52,58
	Kappa	0,9961	0,9957	0,9961	0,9555	0,9579	0,8009	- - -
	EMA	0,0010	0,0022	0,0010	0,0150	0,0128	0,0521	- - -
ANTIVIRUS	ICC (%)	99,30	99,93	99,93	97,39	97,84	93,77	41,62
	IIC (%)	0,70	0,07	0,07	2,61	2,16	6,24	58,38
	Kappa	0,9360	0,9934	0,9934	0,8086	0,8380	0,0416	- - -
	EMA	0,006	0,0013	0,0006	0,0209	0,0155	0,2039	- - -
Aceptado		X	X	X	X	X		

Tabla IV:
CLASIFICACIÓN EN BASE A LAS CATEGORÍAS DE STRIDE DEL CONJUNTO DE DATOS DE ANTIVIRUS.

Categorías	Medidas	J48	Random Forest	Random Tree	Naive Bayes	Bayes Net	SVM (libSVM)	K-NN (Clusters = 2)
S	ICC (%)	98,10	97,27	97,24	93,27	93,56	98,07	66,46
	IIC (%)	1,90	2,73	2,76	6,73	6,45	1,93	33,54
	Kappa	0	0,1440	0,1423	0,2544	0,2612	-0,0006	- - -
	EMA	0,0373	0,0324	0,0331	0,0683	0,0665	0,0193	- - -
T	ICC (%)	97,92	97,72	97,71	93,92	94,13	97,99	66,97
	IIC (%)	2,08	2,28	2,29	6,08	5,87	2,01	33,03
	Kappa	0,0745	0,3445	0,3374	0,3265	0,337	-0,0012	- - -
	EMA	0,0354	0,0278	0,0263	0,0616	0,0602	0,0201	- - -
R	ICC (%)	97,81	97,65	97,44	98,25	98,26	88,50	57,63
	IIC (%)	2,19	2,35	2,56	1,75	1,74	11,50	42,37
	Kappa	0,9561	0,9527	0,9485	0,9648	0,9651	0,7674	- - -
	EMA	0,0272	0,0270	0,0259	0,0187	0,0183	0,1150	- - -
I	ICC (%)	98,47	97,75	97,75	93,90	94,08	98,29	66,83
	IIC (%)	1,53	2,25	2,56	6,10	5,92	1,71	33,17
	Kappa	0,2763	0,2461	0,2387	0,2802	0,2868	-0,0012	- - -
	EMA	0,0255	0,0258	0,0255	0,0635	0,0609	0,0171	- - -
D	ICC (%)	98,47	97,75	97,75	93,90	94,08	98,29	66,83
	IIC (%)	1,53	2,25	2,56	6,10	5,92	1,71	33,17
	Kappa	0,2763	0,2461	0,2387	0,2802	0,2868	-0,0012	- - -
	EMA	0,0255	0,0258	0,0255	0,0635	0,0609	0,0171	- - -
E	ICC (%)	97,06	97,15	97,08	94,04	94,23	96,96	68,00
	IIC (%)	2,94	2,85	2,93	5,97	5,77	3,04	32,00
	Kappa	0	0,4606	0,4419	0,4453	0,4526	-0,002	- - -
	EMA	0,0570	0,0324	0,0330	0,0599	0,0582	0,0304	- - -
Aceptado		X	X	X	X	X		

- También se descartan en este análisis los modelos basados en SVM y en clustering, debido a las mismas razones que han sido expuestas con anterioridad.

Como se puede ver en la *Fig. 7* los resultados más consistentes sobre los modelos aceptados se cumplen con la categoría R (*Repudiation*). Esto se debe principalmente a que su distribución de datos es la más equilibrada de todas las categorías, aspecto fácilmente apreciable en la *Fig. 6*.

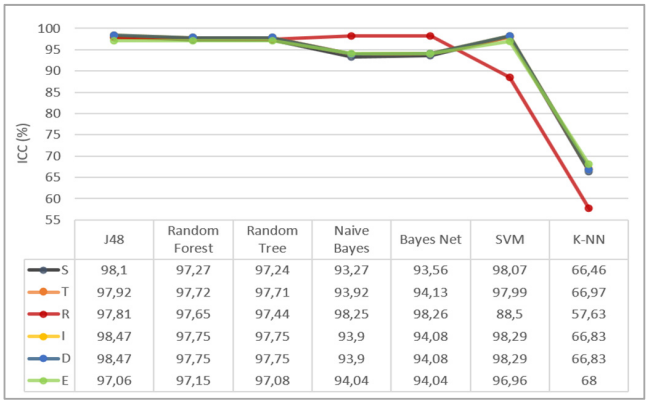


Fig.7. Efectividad entre las categorías de STRIDE de Antivirus.

E. Análisis sobre la clasificación de categorías balanceadas

Para conseguir mayor veracidad en las medidas obtenidas, se decide repetir el mismo análisis realizado sobre las categorías, balanceando los conjuntos de datos. Con ello, además, se puede corroborar la existencia de diferencias entre las medidas obtenidas sin balanceo y con balanceo de datos. La distribución que se obtiene finalmente tras este proceso se muestra en la *Fig. 8*.

Todas las categorías han sido limitadas al máximo número de amenazas que se poseían correspondientes a ambas clases ('Sí' y 'No'). Este proceso produce a su vez un enorme decremento del número de amenazas para la mayoría de las categorías y se debe al poco equilibrio entre diferentes valores con el que se cuenta inicialmente.

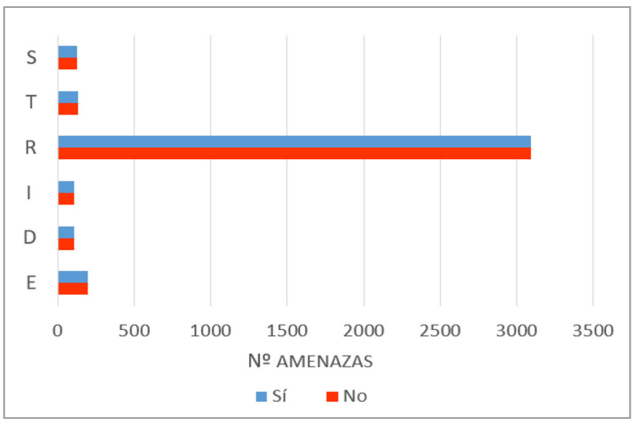


Fig. 8. Distribución balanceada de los datos del proveedor de Antivirus que cumplen con las categorías de STRIDE indicadas.

Finalmente, los resultados del análisis todas las categorías se muestran de forma conjunta en la *Tabla V*.

Tabla V:
CLASIFICACIÓN EN BASE A LAS CATEGORÍAS DE STRIDE DEL CONJUNTO DE DATOS BALANCEADOS DE ANTIVIRUS.

Categorías	Medidas	J48	Random Forest	Random Tree	Naive Bayes	Bayes Net	SVM (libSVM)	K-NN (Clusters = 2)
S	ICC (%)	74,02	75,20	74,41	76,38	76,77	73,23	75,20
	IIC (%)	25,98	24,80	25,59	23,62	23,23	26,77	24,80
	Kappa	0,4803	0,5039	0,4882	0,5276	0,5354	0,4646	- - -
	EMA	0,3114	0,2797	0,2681	0,2550	0,2571	0,2677	- - -
T	ICC (%)	82,31	86,54	86,62	83,46	84,23	82,70	80,00
	IIC (%)	17,69	13,46	15,38	16,54	15,77	17,30	20,00
	Kappa	0,6462	0,7308	0,6923	0,6692	0,6846	0,6538	- - -
	EMA	0,2393	0,2063	0,1995	0,1673	0,1604	0,1731	- - -
R	ICC (%)	97,96	97,46	97,48	98,11	98,11	82,70	80,00
	IIC (%)	2,04	2,54	2,52	1,89	1,89	17,30	20,00
	Kappa	0,9592	0,9492	0,9495	0,9621	0,9621	0,6538	- - -
	EMA	0,0293	0,0295	0,0279	0,0208	0,0202	0,1731	- - -
I	ICC (%)	84,55	87,27	83,64	85,00	85,91	82,73	73,19
	IIC (%)	15,45	12,73	16,36	15,00	14,09	17,27	26,81
	Kappa	0,6909	0,7455	0,6727	0,7000	0,7182	0,6545	- - -
	EMA	0,1948	0,1982	0,1934	0,1530	0,1436	0,1727	- - -
D	ICC (%)	84,55	87,27	83,64	85,00	85,91	82,73	73,19
	IIC (%)	15,45	12,73	16,36	15,00	14,09	17,27	26,81
	Kappa	0,6909	0,7455	0,6727	0,7000	0,7182	0,6545	- - -
	EMA	0,1948	0,1982	0,1934	0,1530	0,1436	0,1727	- - -
E	ICC (%)	88,52	88,52	88,78	90,05	91,33	85,97	79,33
	IIC (%)	11,48	11,48	11,22	9,95	8,70	14,03	20,67
	Kappa	0,7704	0,7705	0,7755	0,801	0,8265	0,7194	- - -
	EMA	0,1746	0,1601	0,1493	0,1051	0,0937	0,1403	- - -
Aceptado		X	X	X	X	X		

F. Discusión de resultados sobre las categorías balanceadas

En la *Tabla V* se aprecia mayor heterogeneidad en las medidas obtenidas, destacando los siguientes aspectos:

- Se produce un importante descenso de *ICC* en la mayoría de categorías, sobre todo en comparación al anterior análisis sin balanceo. Al haber realizado una disminución tan drástica del conjunto de datos, se hace más plausible el número de *IIC* (Instancias Incorrectamente Clasificadas).
- Por otra parte, se produce un notable aumento en los valores *Kappa* de la mayoría de las categorías, superando el valor medio de 0,5 con los modelos aceptados. La correlación entre los clasificadores aumenta, debido a que la distribución de entre las clases es equitativa y ello facilita el aumento del coeficiente *Kappa*.
- Se descartan nuevamente los algoritmos basados en SVM y clustering, ya que sus medidas continúan situándose por debajo del resto de modelos.

Los resultados obtenidos son coherentes con los esperados. El aspecto más destacable de este tercer análisis se enfoca en una categoría que no pasa desapercibida por sus resultados. En *Fig. 9,* se aprecia fácilmente cómo la categoría R (*Repudiation*) obtiene los mejores resultados de *ICC*. En la *Tabla V* se observa que sucede lo mismo con en el resto de medidas. Esto es debido a que la distribución de datos que posee dicha categoría es la más equilibrada con gran diferencia respecto al resto. De esta manera, como se ve en la *Tabla VI* se obtienen prácticamente los mismos resultados con y sin balanceo sobre los algoritmos aceptados.

Se desea hacer constar que los resultados obtenidos a través de esta categoría son extrapolables en gran medida al resto, siempre que se cuente con una distribución de clases similar. Por tanto, los resultados actuales en el resto de las categorías son susceptibles a mejorar en futuros análisis, con los que se cuente con una mayor cantidad de datos.

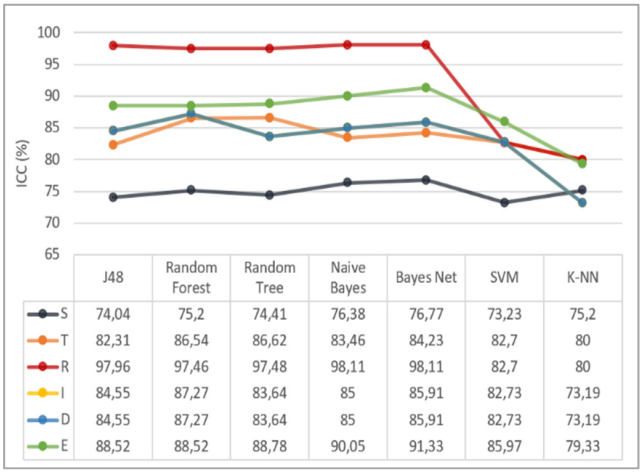


Fig. 9. Efectividad entre las categorías de STRIDE balanceadas de Antivirus.

TABLA VI:
COMPARATIVA ENTRE LOS RESULTADOS SIN Y CON BALANCEO DE LA CATEGORÍA R (REPUDIATION) DE STRIDE SOBRE LOS MODELOS ACEPTADOS.

R	J48	R. F.	R. T.	N. B.
Sin Balanceo	97,81	97,65	97,44	98,26
Con Balanceo	97,96	97, 46	97,48	98,11

G. *Categorías con mayor impacto en el sistema*

Otro de los objetivos de la investigación realizada, consistía en establecer las categorías con mayor impacto a partir de los archivos analizados. En la *Fig. 10* se muestra el resultado de este objetivo a partir de los datos extraídos del Antivirus.

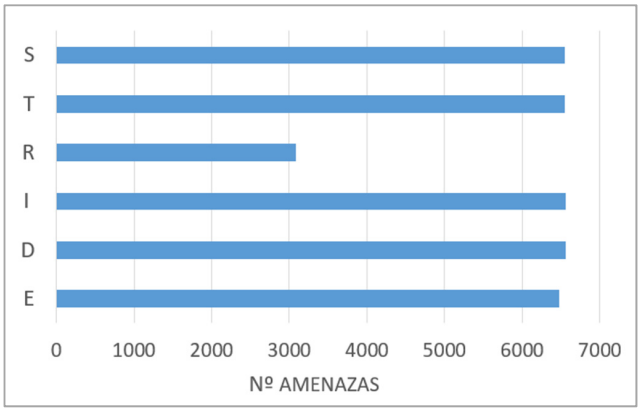


Fig. 10. Impacto de las categorías de STRIDE de Antivirus según el número de amenazas que posee cada una.

Existen una gran cantidad de amenazas que están presentes en la mayoría de categorías. Solamente se aprecia una categoría con un menor número de amenazas con respecto al resto, concretamente la categoría R (*Repudiation*).

Por lo tanto, puede resultar de vital importancia detectar las cadenas de mayor impacto en STRIDE en un sistema que se encuentre protegido por un proveedor de Antivirus. Esto ayudará a identificar los componentes más afectados o débiles del sistema, que deberán ser reforzados para evitar futuras amenazas y vulnerabilidades.

VII. Líneas futuras.

Como líneas futuras, los experimentos diseñados consideran incrementar el número de archivos de fuentes de datos, dado que en el trabajo actual se han centrado sobre 3 archivos de 10000 tuplas cada uno relacionados con sistemas (*Antivirus*, *Firewalls* y *Microsoft*) y se disponen de otros 3 archivos, también de 10000 tuplas cada uno, relacionados con redes (*ControladoresWiFi*, *AccesoRemoto* y *ElectronicaRED*).

También se continuará trabajando para incrementar los tamaños de los ficheros disponibles, así como su variabilidad y criticidad, con el objetivo de afinar al máximo los resultados obtenidos.

Finalmente, se incluirán nuevos modelos de regresores estadísticos, para estudiar las consecuencias de su utilización.

VIII. Conclusiones

En este trabajo se ha presentado un completo sistema de monitorización de eventos e información de seguridad, obtenidos de múltiples fuentes de datos. El modelo propuesto, utilizando las mismas fuentes que el SIEM comercial QRadar, obtiene unos resultados comparables a los del proveedor de Antivirus, en base a implementaciones propias desarrolladas en las investigaciones realizadas.

La distribución de datos de la mayoría de las categorías se encuentra muy poco balanceada o equilibrada. Al balancear los conjuntos de datos se ha conseguido una mayor veracidad en las medidas obtenidas.

Por consiguiente, el modelo propuesto podría completar y complementar la monitorización proporcionada por modelos comerciales, además de permitir la predicción de conductas de riesgo para anticipar respuestas ante estas situaciones.

Agradecimientos

Los autores agradecen la financiación recibida por parte de la Junta de Extremadura (Fondo Europeo de Desarrollo Regional), Consejería de Economía e Infraestructuras (Proyecto GR18138).

Referencias

[1] IBM, “IBM QRadar SIEM.” [Online]. Available: https://www.ibm.com/es-es/marketplace/ibm-qradar-siem.
[2] A. E. S. M. (ESM), “Security Information and Event Management (SIEM).” [Online]. Available: https://www.microfocus.com/en-us/products/siem-security-information-event-management/overview.
[3] Symantec, “Symantec Managed Security Services.”
[4] McAfee, “Información de seguridad y administración de eventos (SIEM).” [Online]. Available: https://www.mcafee.com/enterprise/es-es/products/siem-products.html.
[5] Alienvault, “Alienvault Cibersecurity.”
[6] Fortinet, “FortiSIEM: Powerful Security Information and Event Management.” [Online]. Available: https://www.fortinet.com/products/siem/fortisiem.html.
[7] AdamShostack, *Threat Modeling： Designing for Security*.
[8] Fortinet, “Fortinet named a Leader in the 2018 Gartner Enterprise Firewall Magic Quadrant.” [Online]. Available: https://www.fortinet.com/products/next-generation-firewall.html.
[9] IETF (Internet Engineering Task Force), “The Syslog Protocol,” 2009. [Online]. Available: https://tools.ietf.org/html/rfc5424.
[10] M. Jouini, L. B. A. Rabai, and A. Ben Aissa, “Classification of security threats in information systems,” *Procedia Comput. Sci.*, vol. 32, pp. 489–496, 2014.
[11] M. Abomhara and M. Gerdes, “A STRIDE-Based Threat Model for Telehealth Systems Mohamed Abomhara , Martin Gerdes , Geir M . Køien Department of Information and Communication Technology,” no. November, 2015.
[12] J. Lopez, J. Zhou, M. S. Eds, and D. Hutchison, “STRIDE to a Secure Smart Grid in a Hybrid Cloud,” pp. 77–90, 2018.
[13] J. Lopez, J. Zhou, M. S. Eds, and D. Hutchison, “Towards Security Threats that Matter,” vol. 1, pp. 47–62, 2018.
[14] S. Krishnan, “A Hybrid Approach to Threat Modelling,” 2017.
[15] R. Klöti, V. Kotronis, and P. Smith, “OpenFlow: A security analysis,” *Proc. - Int. Conf. Netw. Protoc. ICNP*, 2013.
[16] I. H. Witten, E. Frank, and M. a Hall, *Data Mining:Practical Machine Learning Tools and Techniques second edition*. 2011.
[17] J. Quinlan Ross, “C4. 5: Programs For Machine Learning,” *Mach. Learn.*, vol. 240, p. 302, 1993.
[18] L. Breiman, “Breiman2001 - Random forests,” pp. 1–33, 2001.
[19] J. M. Parra Murciego, “Estimación De Un Modelo Aditivo No Paramétrico,” 2011.
[20] P. H. Jhon, George; Langley, “Estimating Continuos Distributions in Bayesian Classifiers.”
[21] R. Bouckaert, “Bayesian network classifiers in Weka,” *Dep. Comput. Sci. Univ. ...*, pp. 1–23, 2004.
[22] C. Chang, C. Lin, and T. Tieleman, “LIBSVM : A Library for Support Vector Machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 307, pp. 1–39, 2008.
[23] D. ACM Special Interest Group for Algorithms and Computation Theory., S. SIAM Activity Group on Discrete Mathematics., Association for Computing Machinery., and Society for Industrial and Applied Mathematics., “Kmeans++,” *Proc. eighteenth Annu. ACM-SIAM Symp. Discret. Algorithms*, p. 1317, 2007.
[24] Symantec, *Severity Assessment: Threats, events, vulnerabilities, risks*, no. February. 2006.
[25] A. J. Viera and J. M. Garrett, “Vierra 2005 Interrater agreement Kappa statistic” no. May, pp. 360–363, 2005.

Categorización automática de la severidad de un ciberincidente. Un caso de estudio mediante aprendizaje automático supervisado

Noemí DeCastro-García
Dpto Matemáticas. Universidad de León
Campus de Vegazana s/n 24071 León
ncasg@unileon.es

Mario Fernández-Rodríguez
RIASC - Universidad de León
Campus de Vegazana s/n 24071 León
mfernr@unileon.es

Ángel Luis Muñoz Castañeda
Dpto Matemáticas. Universidad de León
Campus de Vegazana s/n 24071 León
amunc@unileon.es

***Resumen*—En este trabajo se presenta un caso de estudio en el que se construye un modelo que permite categorizar automáticamente la severidad de un incidente de ciberseguridad. Se ha aplicado la categorización a tres tipos de eventos diferentes. La metodología utilizada sigue las fases de la ciencia de datos. El modelo se ha realizado mediante técnicas de aprendizaje automático supervisado sobre una base real de registros de ciberincidentes. Los resultados muestran que los algoritmos ensamblados y aquellos basados en diagramas de construcción lógicos aportan los mejores modelos predictivos de clasificación, obteniendo un ajuste para cada evento con una tasa de acierto superior al 99 %.**

***Index Terms*—Ciberseguridad, severidad, aprendizaje automático**

Tipo de contribución: *Investigación original*

I. Introducción

Caracterizar la severidad de un incidente, entendida como la importancia o la gravedad del mismo, es un procedimiento clave en el campo de la ciberseguridad para poder detectar, predecir y reaccionar ante un evento de una manera óptima y eficaz. Efectivamente, podemos encontrar el nivel de peligrosidad de un evento como uno de los indicadores principales en la *Guía Nacional de Notificación y Gestión de Ciberincidentes*, marco actual de referencia para el reporte y tratamiento de incidentes de ciberseguridad que se produzcan dentro de territorio español, veáse [1].

La necesidad de la obtención de métricas significativas, que permitan cuantificar y valorar el *riesgo* que supone un ciber-incidente, ha llevado a la creación de diferentes metodologías de evaluación o estándares que usan distintas formas de medición de la severidad de las vulnerabilidades descubiertas. Dichas metodologías se basan en rangos de puntuaciones, esquemas cualitativos de valores nominales (crítico, importante, moderado o bajo) o sistemas, como el creado por el *Forum of Incident Response and Security Team* (FIRST), que combinan el impacto y la explotabilidad de los componentes involucrados (ver [2], [3], [4], [5], [6]). Esta variedad de frameworks *ad hoc* disponibles está basada en diferentes combinaciones de taxonomías que caracterizan las amenazas existentes en el ámbito de la ciberseguridad. La primera taxonomía de caracterización existente la encontramos en [7], en la que se define un incidente como un ataque que modifica el estado de un sistema o dispositivo que, desde el punto de vista de las redes y la computación, es el resultado de una acción dirigida a un objetivo específico con una temporalización concreta. El trabajo se centra en la caracterización de los incidentes a través del *objetivo* al que se dirige el ataque aprovechando vulnerabilidades en el diseño, la implementación o la configuración de los sistemas para obtener resultados no autorizados. Otro trabajo relevante en el campo lo podemos encontrar en [8], en el que se propone una taxonomía basada en *dimensiones* que aporta una visión más holística de un ataque. Entre las dimensiones incluidas en la metodología nos encontramos con el vector de ataque, el objetivo de ataque, las vulnerabilidades y exploits (que no tienen una clasificación estructurada), el impacto, el daño ocasionado, los costes que supone la recuperación del ataque, o la defensa ante el incidente. Aunque esta taxonomía enriquece a la anterior, considerando dimensiones importantes relacionadas con la severidad de un incidente, no aporta de manera detallada una forma exhaustiva de medir los factores implicados. De manera paralela a la anterior, en [9] se expone un mecanismo de medición del impacto, entendido como el efecto del ataque (disrupción, distorsión, destrucción, publicación y desconocido), además de ofrecer nuevas categorías para el vector de ataque re-denominado como *Método de Operación*. Además, se introduce el sector objetivo centrándose en sectores comerciales y gubernamentales. Finalmente, y basándonos en el estudio realizado en [10], la taxonomía más completa y rigurosa la encontraríamos en [11]. Esta metodología se basa en cinco elementos principales que son el vector de ataque (antigua dimensión de vulnerabilidades), el impacto operacional (antiguo vector de ataque), las defensas frente al ataque, el impacto sobre la información y el objetivo. Permite la caracterización de ataques mezclados mediante el etiquetado de los vectores de ataque en una estructura de árbol. Además, tiene en cuenta el impacto sobre la información cuya medición se aborda en [2] mediante diferentes pesos y/o ponderaciones.

Sin embargo, las metodologías de evaluación descritas se basan en clasificar ciberincidentes mediante niveles que se centran en características intrínsecas al tipo de amenaza y su comportamiento, apoyándose en distintos factores y usando intervalos de puntuación no formalizados. Estos indicadores se suelen asignar manualmente mediante descripciones cualitativas y ejemplos, lo que dificulta la capacidad de categorizar la severidad de un evento de una manera automática que pueda resultar eficaz en la gestión de incidentes, además de resultar una tarea ardua y laboriosa. A su vez, la asignación de los

niveles implica una especial complejidad en la catalogación de ataques mixtos (que contienen otros ataques) que pueden pertenecer a diferentes categorías si la taxonomía no está bien definida. En este escenario, la aplicación de técnicas de aprendizaje automático (*machine learning*) constituye una alternativa a tener en cuenta debido a que estas analíticas son adecuadas en aquellas situaciones en las que se quiere encontrar una tendencia escondida o creada manualmente en una base de datos, construyendo un modelo que aprende a generalizar el patrón y lo aplica de manera automática a otros datos diferentes pero de la misma naturaleza, reemplazando la heurística a pequeña escala por estadística a gran escala [12].

La aplicación de aprendizaje automático para obtener el riesgo de un ciber-incidente ha dado lugar a la obtención modelos de clasificación que nos permiten, en cierta medida, determinar la severidad de ciber-incidentes concretos con una tasa de acierto elevada frente a los sistemas descritos anteriormente. Este el caso, por ejemplo, de los modelos de clasificación que consiguen discriminar URLs maliciosas y no maliciosas con tasas de acierto elevadas pero limitándose a la categorización binaria, y al uso de un número reducido de features léxicas y/o intrínsecas de los *hosts* ([13]). En esta misma línea de investigación también encontramos trabajos que tratan de establecer si un informe sobre un *bug* puede considerarse como severo o no con el fin de otorgarles un valor de *peligrosidad* ([14], [15], [16]). Por otra parte, en [17], se han aplicado técnicas de aprendizaje supervisado a la evaluación de la severidad en ataques pertenecientes a la tipología *phishing*. En dicho estudio se tiene en cuenta el conocimiento embebido en ataques de *phishing* previos para la creación de modelos que permiten evaluar la severidad de futuros ataques, siendo un indicador fundamental el impacto financiero que implican. Este trabajo supone un incremento de la complejidad del proceso con respecto al anterior, dado que este tipo de ataque se compone de diferentes fases a diferencia de los estudios sobre URLs. Como se puede observar, aunque las técnicas de aprendizaje automático ya se han aplicado con anterioridad a categorizar el riesgo o peligrosidad de un ciberincidentes, estas presentan algunas limitaciones. Entre ellas podemos destacar que los modelos se han centrado en el tratamiento individualizado de incidentes de ciberseguridad muy concretos, y suelen utilizar *features* como la inmediatez y la probabilidad de explotación (difíciles de determinar de manera exhaustiva, eficiente y automática para centros como los Security Operation Center (SOC) o Computer Emergency Response Team (CERT)). Además, los modelos no suelen ofrecer una categorización con diferentes niveles de severidad, siendo la clasificación binaria insuficiente en una gran cantidad de ciberincidentes.

El objetivo de este trabajo es crear un modelo de categorización automática que permita asignar un valor multinivel de *severidad* a diferentes incidentes de ciberseguridad mediante los habituales registros que se generan en un CERT. El modelo, además de ser automático y veraz, debe cumplir las características que ha de reunir una taxonomía para que sea válida ([7]): el principio de exclusión mutua, el principio de exhaustividad, no ambigua, aceptada por la comunidad de expertos, útil y fácilmente replicable. Por otra parte, a medida que ha ido creciendo el número de amenazas también lo ha hecho la complejidad de los registros que generan. De este modo, es relevante considerar el mayor número posible de aspectos registrados, [18].

En este trabajo se presenta la construcción del modelo de categorización automática de la severidad de $N = 5232207$ reportes reales de incidentes de ciberseguridad de diferente naturaleza. El caso de estudio que se desarrolla es el de aquellos ciberincidentes del almacén de datos cuya asignación de severidad requiere algoritmos de aprendizaje automático supervisado, es decir, aquellos en los que tenemos una cantidad de datos (balanceada o no) de casos o eventos cuya severidad ha sido manualmente catalogada por expertos. Pertenecen a tres tipos de incidentes diferentes, y suponen más del 60 % del almacén de datos analizado. Los algoritmos que se han utilizado son aquellos habitualmente utilizados en problemas de clasificación y predicción: Random Forest (RF), Gradient Boosting (GB), Decision Tree (DT), Ada Boosting (AB), K-means (KM) y Multi-Layer Perceptron (MLP).

Los análisis muestran que los mejores resultados se han obtenido con métodos de aprendizaje no paramétricos y que funcionan de manera iterativa como DT, y con los métodos de tipo ensamblado RF y GB. Para determinar el mejor modelo para cada evento se han analizado los indicadores habituales de las técnicas de bondad de ajuste basadas en aprendizaje automático (para datos balanceados y no balanceados) y se ha realizado una fase de validación post construcción del modelo.

El artículo está organizado de la siguiente manera: en la sección II se desarrollan los materiales y métodos utilizados en la investigación. En la sección III se desglosan e interpretan los resultados obtenidos en el caso de estudio. Finalmente, se incluyen las conclusiones y referencias.

II. Materiales y métodos

En esta sección se desarrollan detalles sobre el conjunto de datos utilizado, la metodología de investigación seguida, y las especificaciones técnicas.

II-A. Conjunto de datos

Sea $\mathcal{D}$ un almacén de datos con registros de ciberseguridad. Denotaremos por N el número de casos o eventos, y por V el número de variables (*features*). Los datos que forman el almacén proceden de diferentes fuentes dinámicas, $\mathcal{F}_j$ con $j = 1, \ldots, m$, que aportan datos sobre diversos tipos de eventos $\mathcal{E}_j$, con $j = 1, \ldots, n$. Tanto n como m son variables en el tiempo.

Denotaremos mediante $R_k^{i,j} \in \mathcal{D}$ al evento k-ésimo de $\mathcal{D}$, reportado por la fuente i-ésima y perteneciente al tipo de evento j-ésimo con $k = 1, \ldots, N$, $i = 1, \ldots, m$ y $j = 1, \ldots, n$.

En el caso de estudio presentado en este artículo, el almacén de datos $\mathcal{D}$ contiene $N = 5232207$ casos o reportes reales de ciberseguridad con $V = 113$ *features* de diferente naturaleza. Los incidentes incluidos en la muestra representan un total de $n = 38$ eventos distintos reportados por $m = 27$ fuentes. Este conjunto de datos ha sido aportado por INCIBE bajo acuerdo de confidencialidad, y ha sido seleccionado mediante muestreo bajo criterios de representatividad de los datos.

Cabe destacar que no todas las fuentes aportan la misma cantidad ni el mismo tipo de información (en lo referente a los

distintos tipos de eventos que reportan) debido a la distinta naturaleza de las mismas. A su vez, los datos almacenados son heterogéneos, presentando diferentes tipos de procesado y tasas de creación no constantes, lo que da lugar a eventos correlados en algunos casos y no balanceados en su mayoría. Además, no todos los $\mathcal{E}_j$ disponen de datos en todas las *features* $\mathcal{V}$, o esta puede ser incompleta.

II-B. Metodología

La metodología de investigación es cuantitativa y cuasi-experimental. Se enmarca en el cuarto paradigma de la investigación científica o ciencia intensiva de datos: analizar el almacén de datos, crear una base de datos, construir los modelos de aprendizaje automático y seleccionar el que se considera más óptimo.

II-C. Analíticas

Los algoritmos de aprendizaje automático supervisado que se han aplicado son aquellos habitualmente utilizados en problemas de clasificación y predicción: RF, GB, DT, AB, KM y MLP. Además, a todos los algoritmos mencionados se les han aplicado métodos automáticos de selección de hiperparámetros y muestreo para que los modelos obtenidos estén optimizados en la mayor medida posible ([19]).

Una vez que los modelos han sido entrenados y testeados, se han de calcular las métricas habituales utilizadas en la medición de la bondad de un modelo de aprendizaje automático ([20]). El primer indicador que se toma en cuenta es el ajuste, tasa de acierto o *accuracy*. Sin embargo, para aquellos casos en los que las matrices de confusión resultan demasiado descompensadas, resultado habitual al trabajar con conjuntos de datos no balanceados, se hará uso del coeficiente de correlación de Mathews (MCC), [21]:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Además, y aunque no se han ponderado de manera global debido a que su importancia depende del tipo de evento con el que estemos trabajando, sí se han tenido en cuenta indicadores como la sensibilidad, especificidad, la puntuación F_1 o la precisión. Para el cálculo de estos índices, y nuevamente debido a las clases no balanceadas, se ha optado por utilizar la *micro average* en lugar de la media habitual.

Por otra parte, y con la finalidad de analizar la capacidad de generalización del modelo ante patrones no conocidos, se ha utilizado la curva de aprendizaje del modelo.

II-D. Especificaciones técnicas

Los recursos materiales con los que se ha llevado a cabo la experimentación son los recogidos en la tabla I.

Tabla I
EQUIPAMIENTO TÉCNICO DISPONIBLE

Componente	Detalles
Memoria RAM	54 GiB
Modelo Procesador	Intel Xeon(R) CPU X5670 @ 2.93GHz
Número Procesadores	2
Sistema Operativo	Ubuntu 18.04.1 LTS 64bits
Lenguaje	Python 2.7.15

Los modelos se han creado usando la librería Scikit-learn en Python 2.7, [22].

III. RESULTADOS

Esta sección se desarrolla siguiendo los puntos de la metodología llevada a cabo. En este trabajo se presentan aquellos eventos para los que se ha utilizado aprendizaje automático supervisado.

III-A. Análisis del almacén de datos y Selección de analíticas

Como se ha indicado con anterioridad, $N = 5232207$, $n = 38$ y $m = 27$ fuentes.

La *Severidad* se presenta como una variable objetivo multiclase que dispone de 4 etiquetas diferenciadas:

$$\mathcal{S}_{R_k} = \begin{cases} 0 = \text{ desconocida} \\ 1 = \text{ baja} \\ 2 = \text{ media} \\ 3 = \text{ alta} \end{cases} \quad (1)$$

Cada tipo de evento puede disponer de todas las etiquetas, algunas, o simplemente desconocer el valor de la severidad. Por ese motivo, el primer paso ha sido reagrupar los tipos de eventos en tres categorías, ya que esta clasificación determinará el tipo de analítica que se ha de utilizar.

1. $G_1 = \{\mathcal{E}_j \text{ tal que } \mathcal{S}_{R_k^{i,j}} = 0 \ \forall \ i = 1, \ldots, m\}$.
2. $G_2 = \{\mathcal{E}_j \text{ tal que } \mathcal{S}_{R_k^{i,j}} = \text{ conocida y veraz } \forall \ i = 1, \ldots, m\}$.
3. $G_3 = \{\mathcal{E}_j \text{ tal que } \exists \ R_k^{i,j}, R_l^{i,j} \in \mathcal{D} \text{ con } \mathcal{S}_{R_k^{i,j}} = 0 \text{ y } \mathcal{S}_{R_l^{i,j}} \neq 0\}$.

En función de la categoría a la que pertenezca un incidente, se determinará el tipo de analítica que necesita. Así, los eventos del grupo G_1 han de analizarse mediante técnicas de aprendizaje automático no supervisado. En el caso del G_2 no será necesario realizar ningún tipo de modelado. Finalmente, el caso G_3 implica la aplicación de técnicas de aprendizaje automático supervisado.

En este estudio nos centraremos únicamente en aquellos tipos de eventos $\mathcal{E}_j \in G_3$. En el caso del almacén de datos $\mathcal{D}$, tendremos tres tipos de eventos en este grupo, que denotaremos por $\mathcal{E}_1, \mathcal{E}_2$, y $\mathcal{E}_3$, respectivamente. Estos eventos son de diferentes tipologías entre las que se encuentran algunas relacionadas con contenido dañino o vulnerabilidades ([1]).

Los tipos de eventos pertenecientes al grupo G_3, que son reportados por las fuentes recogidas en la tabla II, representan, sobre el total de eventos recogidos en $\mathcal{D}$, el 61 % en el caso de $\mathcal{E}_1$, el 0,07 % en el caso del $\mathcal{E}_2$ y el 0,02 % para el $\mathcal{E}_3$.

Tabla II
FUENTES Y EVENTOS QUE REPORTAN

Fuente	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$
$\mathcal{F}_5$	X	X	
$\mathcal{F}_7$	X		
$\mathcal{F}_{10}$	X		
$\mathcal{F}_{15}$	X		
$\mathcal{F}_{18}$	X		
$\mathcal{F}_{21}$	X		X
$\mathcal{F}_{23}$	X		

El balanceo de los datos etiquetados disponibles en el conjunto de datos $\mathcal{D}$ para los eventos del grupo G_3 se encuentra disponible en la tabla III.

Tabla III
DISTRIBUCIÓN DISPONIBLE DE DATOS ETIQUETADOS

Tipo de evento	$\mathcal{S}=0$	$\mathcal{S}=1$	$\mathcal{S}=2$	$\mathcal{S}=3$
$\mathcal{E}_1$	0 %	2,44 %	51,89 %	45, 66 %
$\mathcal{E}_2$	53,29 %	0 %	1,80 %	44,89 %
$\mathcal{E}_3$	45,12 %	8,69 %	46,18 %	0 %

III-B. Creación del dataset

III-B1. Transformación: El conjunto de observaciones para cada uno de los eventos de G_3 se divide en primer lugar en dos subconjuntos: $\mathcal{D}_j^{train-test}$ (90 %) y $\mathcal{D}_j^{validation}$ (10 %) para cada uno de los tipos de eventos $\mathcal{E}_j,\ j=1,2,3$. Las particiones se han realizado de manera aleatoria debido a que en un entorno real, no siempre es posible balancear los conjuntos de entrenamiento y validación bajo las condiciones más favorables. Como los modelos de aprendizaje automático necesitan ser entrenados y posteriormente testados, para obtener el que mejor cataloga las observaciones para un evento determinado, el subconjunto de datos $\mathcal{D}_j^{train-test}$ se vuelve a dividir en dos nuevos subconjuntos de datos, $\mathcal{D}_j^{train-test} = \mathcal{D}_j^{train} \cup \mathcal{D}_j^{test}$ con un 80 % y 20 % de las observaciones, respectivamente. Dado que los datos catalogados varían tanto en número de observaciones como en número de etiquetas disponibles para cada uno de los eventos, el proceso de separación de datos en subconjuntos de datos (validación, entrenamiento y testado) se repite para cada uno de los eventos.

Partiendo de la distribución de los datos descrita en III se han dividido los datos pertenecientes a cada evento en los subconjuntos de datos recogidos en la tabla IV. Para la creación de dichos subconjuntos, se filtran aquellos $R_k^{i,j}$ con $\mathcal{S}_{R_k^{i,j}} = 0$ para separarlos del resto, puesto que no forman parte ni de la etapa de entrenamiento ni de la de validación posterior.

Tabla IV
DIVISIÓN DE LOS DATOS EN SUBCONJUNTOS POR EVENTO

Evento	Subconjunto	$\mathcal{S}=1$	$\mathcal{S}=2$	$\mathcal{S}=3$
$\mathcal{E}_1$	Entrenamiento	55727	1183860	1041671
	Testado	13932	295965	260418
	Validacion	7740	164426	144677
$\mathcal{E}_2$	Entrenamiento	0	49	1235
	Testado	0	13	309
	Validacion	0	7	172
$\mathcal{E}_3$	Entrenamiento	64	344	0
	Testado	17	86	0
	Validacion	9	48	0

III-B2. Filtrado, recodificación y selección de features*:* El proceso de selección de features se ha realizado en diferentes etapas. Se ha comenzado con una etapa de descarte, eliminando primeramente aquellas *features* que los expertos conocedores del almacén de datos recomendaron no tener en cuenta para catalogar la severidad. A continuación se han eliminado aquellas features que presentan una redundacia superficial (features vacías, constantes y/o equivalentes) ([23]).

Las *features* resultantes pasan a ser pre-tratadas acorde a su semántica con el fin de obtener la máxima información de las mismas y para poder seleccionar las más relevantes de manera automática. La recodificación se ha basado en el uso de diccionarios estáticos cuando el número de valores distintos que toman las *features* es acotado. Sin embargo, para la mayoría de las *features* no hay una lista acotada de valores, por lo que la recodificación se ha basado en diccionarios con asignaciones secuenciales. Por otra parte, las *features* textuales han requerido recodificación en su mayoría.

Una vez recodificadas, comienza el proceso de selección de las features relevantes, que puede realizarse mediante diferentes índices (índice de Gini, χ^2, etc), siendo el coeficiente de la Función de Información Mutua normalizado el que se ha utilizado en este trabajo (ver ecuación 2). Este método no paramétrico está basado en la estimación de la entropía entre dos features, aportando un coeficiente que mide la dependencia entre ellas ([24], [25]). Al estar normalizada, se han obtenido valores entre 0 y 1.

$$MI(U,V) := \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|} \qquad (2)$$

Además, una vez calculado este coeficiente para cada *feature* y para poder decidir cuáles entran al modelo, se ha introducido un valor umbral (percentil 80) que nos permita introducir aquellas *features* más relevantes.

Observación III.1 *Es importante remarcar que esta selección de* features *se realiza para cada tipo de evento $\mathcal{E}_1$, $\mathcal{E}_2$ y $\mathcal{E}_3$ de manera diferenciada.*

Los resultados se muestran en la tabla V.

Tabla V
Features USADAS PARA LA CREACIÓN DE LOS MODELOS SEGÚN EVENTO

Tipos de *feature* eliminada	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$
Features expertos	23	23	23
Features vacías, constantes y equivalentes	30	54	62
Features candidatas	59(+1)	35(+1)	27(+1)
Features seleccionadas (FIM)	15	7	5

Las *features* seleccionadas tras la aplicación de la Función de información Mutua ya pueden ser usadas en la siguiente etapa de creación de modelos.

III-C. Construcción de modelos de aprendizaje automático

III-C1. Selección del modelo: Los ajustes de cada uno de los modelos, para los distintos eventos, obtenidos en la etapa de testado se recogen en la tabla VI.

Los indicadores correspondientes a las matrices de confusión y curvas de aprendizaje para esos mismos modelos, por cada uno de los eventos, obtenidos durante la etapa de testado, se recogen.

Para el caso de $\mathcal{E}_1$, que es el evento más frecuente en el almacén de datos suponiendo más del 60 % de los casos, podemos observar que los modelos que tienen un mayor ajuste son DT, GB y RF. Además, su coeficiente MCC es muy cercano a la unidad, aportando una predicción prácticamente perfecta ($MCC \in [-1,1]$). Con respecto a la capacidad de los modelos para detectar cada una de las clases de severidad, la sensibilidad y la especificidad muestran valores muy elevados en casi todos los modelos. Cabe destacar que para los modelos

Tabla VI
INDICADORES DE CADA MODELO POR EVENTO

Modelo	Indicador	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$
AB	Ajuste	0. 86	1.0	1.0
	Mcc	0. 73	1.0	1.0
	Sensibilidad	0.86	1.0	1.0
	Especificidad	0.9131	1.0	1.0
	Precisión	0.86	1.0	1.0
	F_1-*score*	0.86	1.0	1.0
DT	Ajuste	0. 99	1.0.	1.0
	Mcc	0.9999	1.0	1.0
	Sensibilidad	1.0	1.0	1.0
	Especificidad	0.9999	1.0	1.0
	Precisión	1.0	1.0	1.0
	F_1-*score*	1.0	1.0	1.0
GB	Ajuste	0.99	1.0	1.0
	Mcc	0.9999	1.0	1.0
	Sensibilidad	1.0	1.0	1.0
	Especificidad	0.9999	1.0	1.0
	Precisión	1.0	1.0	1.0
	F_1-*score*	1.0	1.0	1.0
KM	Ajuste	0.65	0.86	0.73
	Mcc	0.41	0.14	-0.15
	Sensibilidad	0.65	0.86	0.73
	Especificidad	0.8168	0.75	0.62
	Precisión	0.65	0.86	0.73
	F_1-*score*	0.65	0.86	0.73
RF	Ajuste	0.99	0.99	1.0
	Mcc	0. 9999	0.96	1.0
	Sensibilidad	1.0	0.9983	1.0
	Especificidad	0.9999	0.9989	1.0
	Precisión	1.0	1.0	1.0
	F_1-*score*	1.0	1.0	1.0
MLP	Ajuste	0.52	0.96	0.83
	Mcc	0.0	0.0	0.0
	Sensibilidad	0.52	0.96	0.83
	Especificidad	0.6666	0.66	0.66
	Precisión	0.52	0.96	0.83
	F_1-*score*	0.52	0.96	0.83

con mayor ajuste, la sensibilidad obtiene su valor máximo y la especificidad alcanza un valor de 0.999. En cuanto a la precisión y la puntuación F_1, podemos observar que éstas varían desde valores bajos como 0.52 hasta su valor máximo de 1. Podemos comprobar que el comportamiento de los indicadores es similar para los modelos DT, GB y RF, pudiendo descartar que el modelo para $\mathcal{E}_1$ lo obtengamos desde MLP, KM o AB.

Como podemos observar, para $\mathcal{E}_1$ existe un triple empate para los modelos DT, GB y RF. Esta situación nos lleva a plantear un mecanismo de selección del mejor modelo en el que se tenga en cuenta el comportamiento esperado de estos ante la generalización con patrones no conocidos, ver Fig. 1.

A continuación, pasaremos a analizar el evento $\mathcal{E}_2$. Aunque este tipo de incidente no sea tan frecuente en el conjunto de datos, está caracterizado por estar claramente descompensado desde el punto de vista de las asignaciones que tenemos disponibles de este (únicamente $\mathcal{S} = 2, 3$). Si nos fijamos en los resultados incluidos en la tabla VI, podemos ver un ajuste perfecto en los modelos AB, GB y DT. Cabe destacar que el modelo construido mediante el algoritmo MLP no resulta mejor que una clasificación aleatoria ($MCC = 0$), aún cuando su ajuste sea muy elevado (Acc=0.96). Esta diferencia se debe, principalmente, al desequilibrio presente en los datos. Aún teniendo presente esta diferencia, este modelo obtiene mejores indicadores que KM cuyos resultados son los peores.

Para poder resolver el empate que ha resultado, acudiremos de nuevo a las curvas de aprendizaje, ver Fig. 2.

Finalmente, pasamos a analizar el evento $\mathcal{E}_3$, el menos frecuente en la base de datos y, al igual que el anterior, con clases no balanceadas ($\mathcal{S} = 1, 2$). Si volvemos a fijarnos en los resultados mostrados en la tabla VI, podemos observar que tenemos un comportamiento similar,y con un ajuste perfecto, de los modelos creados mediante los algoritmos AB, GB, DT y RF. Además, se vuelve a obtener una diferencia muy elevada entre los ajustes resultantes de los modelos generados con MLP y KM si tenemos en cuenta el desequilibrio entre las clases que se obtienen en las matrices de confusión de dichos algoritmos. Cabe destacar que incluso en el modelo construido con el algoritmo KM, el valor del MCC es negativo, comenzando a mostrar una correlación negativa entre los casos reales y los predichos en cada clase.

Para poder seleccionar el modelo más apropiado para la clasificación automática de la severidad de un determinado evento, el procedimiento seguido es la elaboración de un ranking que determina cuál es el mejor modelo de categorización. Este ranking está basado en los indicadores mencionados, ponderando de manera positiva el ajuste y las predicciones correctas. Debido a que se producen empates en la primera posición en algunas ocasiones, y a que las curvas de aprendizaje son similares para los modelos en las primeras posiciones, se ha introducido el tiempo de creación de los modelos como medida de desempate. Pueden consultarse los resultados en la tabla VII.

Tabla VII
RANKING DE LOS MODELOS

Algoritmo	$\mathcal{E}_1$	$\mathcal{E}_2$	$\mathcal{E}_3$
AB	4º	3º	2º
DT	2º	**1º**	3º
GB	**1º**	2º	4º
KM	5º	6º	6º
RF	3º	4º	**1º**
MLP	6º	5º	5º

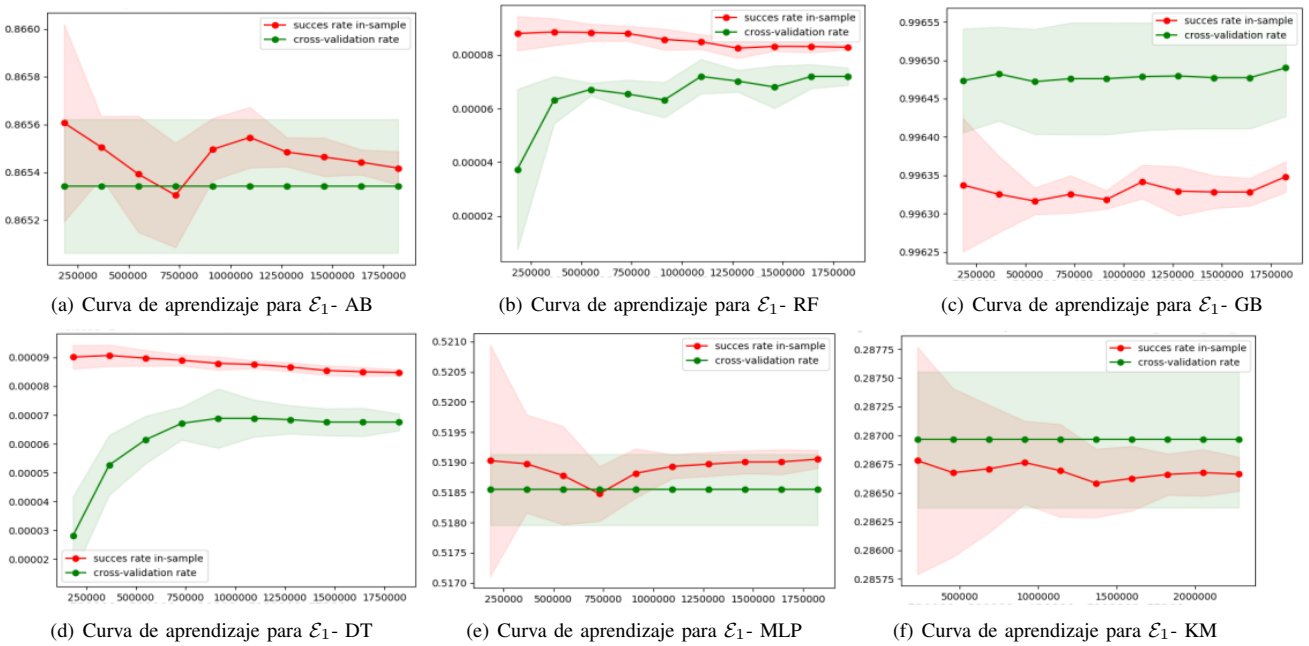
III-C2. Validación: Una vez se han seleccionado los modelos, se ha procedido a realizar una última etapa: la validación. Para poder llevar a cabo esta fase, se ha obtenido el ajuste de los modelos seleccionados cuando estos son aplicados a $\mathcal{D}^{valid}$ para cada tipo de evento.

Los resultados han sido del 99,9 % para el modelo obtenido para $\mathcal{E}_1$, y del 100 % para los modelos obtenidos para $\mathcal{E}_2$ y $\mathcal{E}_3$.

III-D. Discusión y limitaciones

Los modelos obtenidos han sido generados con un único software que automáticamente ha separado los datos catalogados de los no catalogados de cada evento dentro del almacén de datos, acudiendo automáticamente al aprendizaje automático supervisado para aquellos eventos que tenían datos catalogados. Además, también se realiza de manera automática la creación de la base de datos, la selección de *features*, la construcción y cálculo de los modelos y sus indicadores, y la selección y validación final.

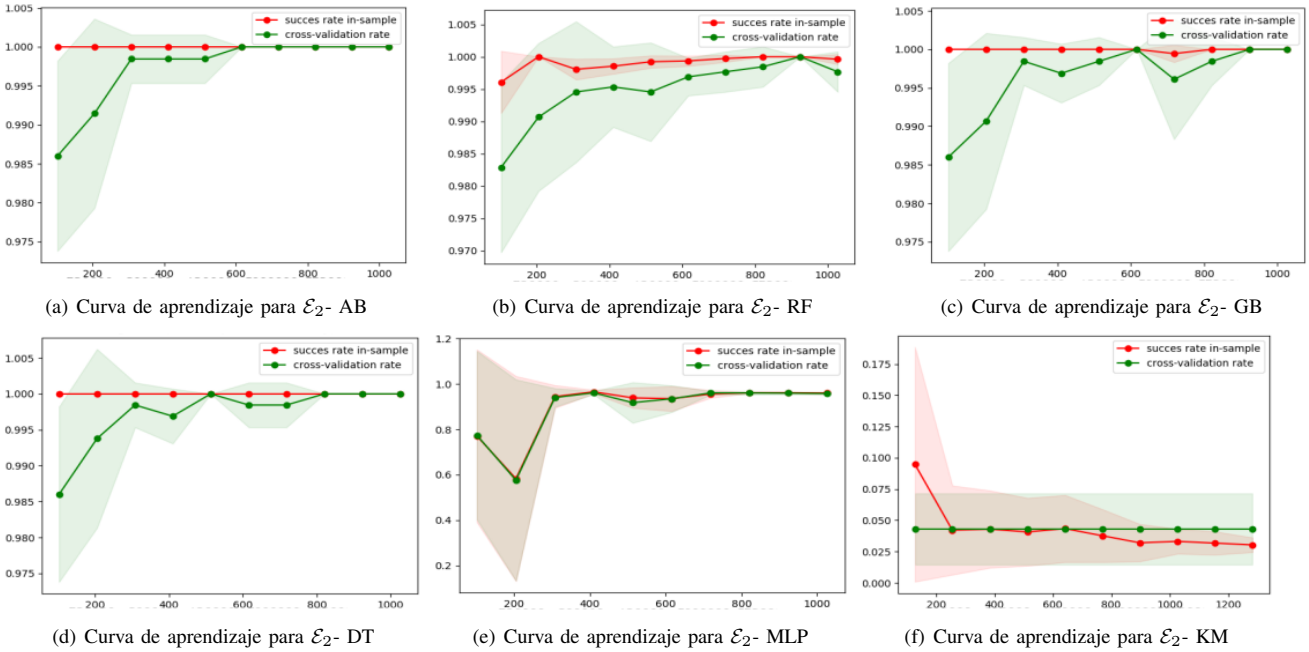
Los modelos creados son aplicables a cualquier entorno real en el que se incluyan registros de los ciberincidentes analizados, y sea necesario catalogar su severidad. Debido al dinamismo de este tipo de almacén de datos, la aplicación



(a) Curva de aprendizaje para $\mathcal{E}_1$- AB (b) Curva de aprendizaje para $\mathcal{E}_1$- RF (c) Curva de aprendizaje para $\mathcal{E}_1$- GB

(d) Curva de aprendizaje para $\mathcal{E}_1$- DT (e) Curva de aprendizaje para $\mathcal{E}_1$- MLP (f) Curva de aprendizaje para $\mathcal{E}_1$- KM

Figura 1. Curvas de aprendizaje para E_1. Eje X: tasa de acierto. Eje Y: tamaño de muestra de entrenamiento



(a) Curva de aprendizaje para $\mathcal{E}_2$- AB (b) Curva de aprendizaje para $\mathcal{E}_2$- RF (c) Curva de aprendizaje para $\mathcal{E}_2$- GB

(d) Curva de aprendizaje para $\mathcal{E}_2$- DT (e) Curva de aprendizaje para $\mathcal{E}_2$- MLP (f) Curva de aprendizaje para $\mathcal{E}_2$- KM

Figura 2. Curvas de aprendizaje para E_2. Eje X: tasa de acierto. Eje Y: tamaño de muestra de entrenamiento

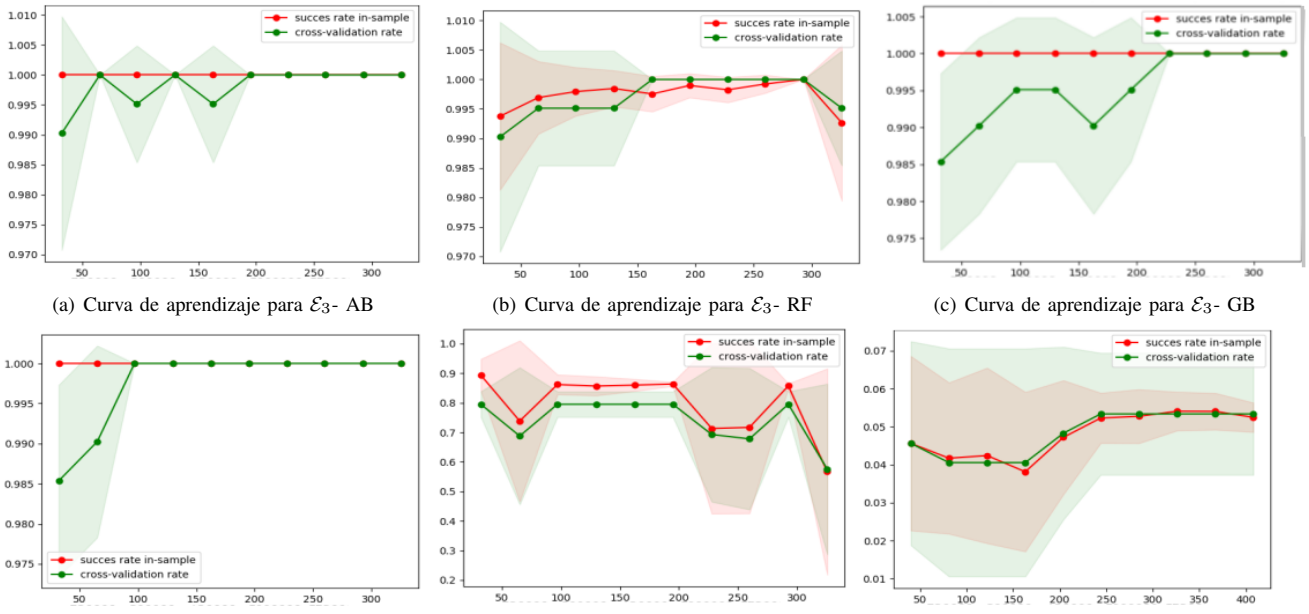
del modelo requeriría un re-entrenamiento de lo modelos de forma periódica. Este entrenamiento ha de ser creado mediante procesado en *batching*, aunque la aplicación de lo modelos obtiene su máxima eficiencia cuando se realiza a tiempo real.

Una de las principales limitaciones del trabajo es la confidencialidad de los datos con los que se ha trabajado.

IV. Conclusiones

En este trabajo se ha desarrollado un modelo de categorización automática de la severidad de ciberincidentes a través de los registros generados por estos en los centros de gestión de eventos de ciberseguridad.

La solución propuesta puede ser aplicada al cálculo del nivel de peligrosidad de los ciberincidentes con la finalidad de poder establecer medidas preventivas y reactivas ante eventos de ciberseguridad. La posibilidad de realizar esta asignación de manera automática permite ofrecer un sistema de gestión de incidentes más eficaz que no suponga demoras o atrasos, además de optimizar los recursos materiales y personales de las instituciones o empresas.



(a) Curva de aprendizaje para $\mathcal{E}_3$- AB (b) Curva de aprendizaje para $\mathcal{E}_3$- RF (c) Curva de aprendizaje para $\mathcal{E}_3$- GB

(d) Curva de aprendizaje para $\mathcal{E}_3$- DT (e) Curva de aprendizaje para $\mathcal{E}_3$- MLP (f) Curva de aprendizaje para $\mathcal{E}_3$- KM

Figura 3. Curvas de aprendizaje para E_3. Eje X: tasa de acierto. Eje Y: tamaño de muestra de entrenamiento

AGRADECIMIENTOS

Este trabajo se enmarca dentro de los contratos de investigación con clave orgánica X50 y X54 financiados por el Instituo Nacional de Ciberseguridad de España (INCIBE), y realizado en RIASC (ULE). Además, los recursos del Centro de Supercomputación de Castilla y León (SCAYLE,www.scayle.es), financiados por "European Regional Development Fund (ERDF)", han sido utilizados para realizar esta investigación.

REFERENCIAS

[1] Ministerio de Interior, "Guía Nacional de Notificación y Gestión de Ciberincidentes", 2019. [Online]. Disponible en: http://www.interior.gob.es/documents/10180/9814700/Gu\%C3\%ADa+Nacional+de+notificacin+y+gestin+de+ciberincidentes.pdf/f01d9ed6-2e14-4fb0-b585-9b0df20f2906.

[2] National Cybersecurity and Communications Integration Center, "NCCIC Cyber Incident Scoring System", 2016. [Online]. Disponible en: https://www.us-cert.gov/NCCIC-Cyber-Incident-Scoring-System.

[3] Open Web Application Security Project (OWASP), "OWASP Testing Guide: OWASP Risk Rating Methodology", 2003. [Online]. Disponible en: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology#Step_4:_Determining_the_Severity_of_the_Risk.

[4] NCISS Incident Scoring Demo-US-CERT, "NCISS Incident Scoring Demo", 2014. [Online]. Disponible en: https://www.us-cert.gov/nciss/demo.

[5] Microsoft, "Description of the standard terminology that is used to describe Microsoft software updates", 2014. [Online]. Disponible en: https://support.microsoft.com/en-us/help/824684/description-of-the-standard-terminology-that-is-used-to-describe-micro.

[6] Forum of Incident Response and Security Teams, "Common Vulnerability Scoring System SIG", 2005. [Online]. Disponible: en https://www.first.org/cvss/

[7] J. D. Howard y T. A. Longstaff: "A common language for computer security incidents", en *Technical report, Sandia National Laboratories*, 1998.

[8] S. Hansman y R. Hunt: "A Taxonomy of Network and Computer Attacks", en *Computers and Security*, vol. 24, n. 1, pp. 31–43, 2005.

[9] M. Kjaerland: "A taxonomy and comparison of computer security incidents from the commercial and government sectors", en *Computers and Security*, vol. 25, n. 7, pp. 522–538, 2006.

[10] N. Abrek: "Attack Taxonomies and Ontologies", en *Seminar Future Internet SS2014*, 2014.

[11] C. Simmons, C. Ellis, S. Shiva, D. Dasgupta y Q. Wu: "AVOIDIT: A cyber attack taxonomy", en *9th Annual Symposium on Information Assurance*, 2014.

[12] M. Bozorgi, L. K. Saul, S. Savage y G. M. Voelker: "Beyond heuristics: Learning to classify vulnerabilities and predict exploits", in *Proc. ACM international conference on knowledge discovery and data mining (SIGKDD)*, pp. 105-114, 2010.

[13] J. Ma, L.K. Saul, S. Savage y G.M. Voelker: "Beyond blacklists: learning to detect malicious web sites from suspicious URLs", en *Proceedings of the 15th ACM international conference on knowledge discovery and data mining (SIGKDD)*, pp 1245–1254.

[14] K.K. Chaturvedi y V.B. Singh: "Determining bug severity using machine learning technique", en *Proc International Conference on Software Engineering(CONSEG)*, 2012. pp. 378-387. doi: 10.1109/ABLAZE.2015.7154933

[15] A. F. Otoom, D. Al-Shdaifat, M. Hammad y E. E. Abdallah: "Severity prediction of software bugs", en *7th International Conference on Information and Communication Systems (ICICS)*, pp. 92-95, 2016.

[16] A. Lamkanfi, S. Demeyer, Q. Soetens, y T. Verdonck: "Comparing mining algorithms for predicting the severity of a reported bug", en *19th Working Conference on Reverse Engineering (CSMR)*, 2011. doi: 10.1109/WCRE.2012.31

[17] X. Chen, I. Bose, A.C.M. Leung y C. Guo: "Assessing the severity of phishing attacks: a hybrid data mining approach", en *Decision Support Systems*, vol. 50, n. 4, pp. 662-672, 2011.

[18] J. Happa y M. Goldsmith: "On properties of cyberattacks and their nuances", en *PSU Research Review*, vol. 1, n. 2, pp. 76-90, 2017. doi: 10.1108/PRR-04-2017-0024

[19] N. DeCastro-García, Á. L. Muñoz Castañeda, D. Escudero García y M. V. Carriegos: "Effect of the Sampling of a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm", en *Complexity*, vol. 2019, Article ID 6278908, 2019, doi: 10.1155/2019/6278908.

[20] D. M. W. Powers: "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation", en *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37-63, 2011.

[21] B. W. Matthews: "Comparison of the predicted and observed secondary structure of T4 phage lysozyme", en *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, n. 2, pp. 442–451, 1975. doi:10.1016/0005-2795(75)90109-9.

[22] F. Pedregosa, G.Varoquaux, A. Gramfort et. al: "Scikit-learn: Machine Learning in Python", en *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011. Disponible: https://github.com/scikit-learn/scikit-learn.

[23] N. DeCastro-García, A. L. Muñoz Castañeda, M. Fernández Rodríguez y M. V. Carriegos: "On Detecting and Removing Superficial Redun-

dancy in Vector Databases", en *Mathematical Problems in Engineering*, vol. 2018, Article ID 3702808, 2018. doi:10.1155/2018/3702808.

[24] L. F. Kozachenko y N. N. Leonenko: "Sample Estimate of the Entropy of a Random Vector", en *Problems of Information Transmission*, vol. 23, n. 2, pp. 9-16, 1987.

[25] A. Kraskov, H. Stogbauer y P. Grassberger: "Estimating mutual information", en *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, n. 066138, 2004. doi:10.1103/PhysRevE.69.066138

OSINT is the next Internet goldmine: Spain as an unexplored territory

Javier Pastor-Galindo*, Pantaleone Nespoli, Félix Gómez Mármol, and Gregorio Martínez Pérez
Department of Information and Communications Engineering, University of Murcia, 30100 Murcia, Spain
Email: {*javierpg, pantaleone.nespoli, felixgm, gregorio*}@um.es

***Abstract*—Phenomenons like Social Networks, Cloud Computing or Internet of Thing are unknowingly generating unimaginable quantities of data. In this context, Open Source Intelligence (OSINT) exploits such information to extract knowledge that is not easily appreciable beforehand by the human eye. Apart from the political, economic or social applications OSINT may bring, there are also serious global concerns that could be covered by this paradigm such as cyber crime and cyber threats. The paper at hand presents the current state of OSINT, the opportunities and limitations it poses, and the challenges to be faced in the future. Furthermore, we particularly study Spain as a potential beneficiary of this powerful methodology.**

***Index Terms*—OSINT, Cyber security, Cyber defense, Cyber intelligence, Spain, Law Enforcement Agencies, Threat Intelligence**

I. Introduction

Open Source Intelligence (OSINT) embraces a set of techniques collecting information from different open sources (e.g., legally available documents, social networks, public activities of states, companies and society, etc.) in order to infer knowledge to be used for a specific purpose [24]. Although it might seem to be a novel paradigm, it has actually been around for a long time. For instance, during the World War II the radio was snooped to spy the adversaries. Already in the year 1941, the *Foreign Broadcast Information Service (FBIS)* was created by the USA to gather public information from other countries. Even during the Cold War the Soviet Union, China and other countries used OSINT through the exploration of public documents and technical information of foreign developments [18].

Traditional OSINT was conducted in a manual fashion in the sense that it was necessary to have analysts in charge of collecting public data and analyzing it in order to extract knowledge. However, the current era of the information has provoked that such a growing and huge amount of data is available on the Internet [23]. As a consequence, original OSINT processes become ineffective with this modern demanding conditions. This issue motivates the development of innovative tools for automating the collection and analysis of data.

Nowadays, OSINT is widely used by governments and intelligence services to conduct their investigations [1]. Nevertheless, it is not only utilised for state affairs, but also new research lines are taking advantage of this paradigm for many goals. Indeed, actual research works tend to follow three main applications which are represented in Figure 1 and are described next:

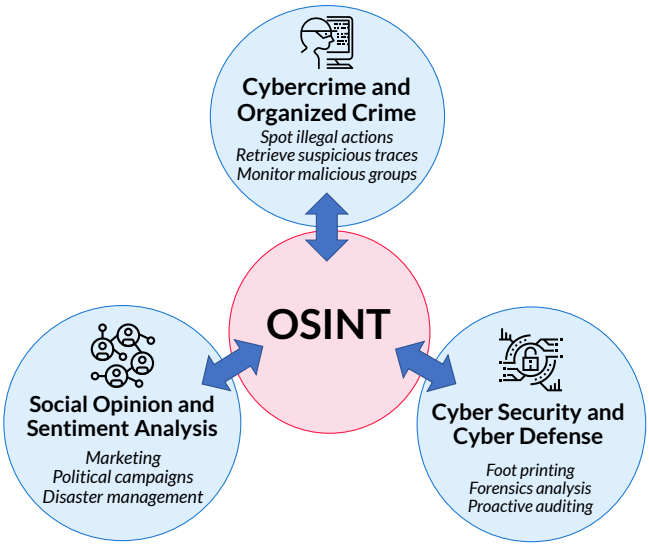


Fig. 1. OSINT principal use cases

- *Social opinion and sentiment analysis:* Related to the boom of social networks, it is possible to collect user's interactions, messages, interests and preferences to extract non-explicit knowledge. Such collection and analysis could be applied to marketing, political campaigns, disaster management or even cyber defense [3].
- *Cybercrime and organized crime:* The open data is continuously analyzed and matched by OSINT processes in order to spot criminal intentions at an early stage. Taking into account adversaries' patterns and relationships between felonies provides to security forces an opportunity to promptly detect illegal actions [16].
- *Cyber security and cyber defense:* Information and Communication Technology systems are continuously attacked by criminals [12]. Research becomes hence crucial to defend ourselves from cyber attackers, concretely by facing the challenges that are still open in the field of cyber security [10]. In this sense, data sciences are not only being applied to the footprinting in pentestings, but also to the preventive protection of the organizations and companies. Concretely, by performing analysis of daily

*Corresponding author

attacks, correlating them and supporting decision making for an effective defense, but also for a prompt reaction [19]. In the same way, OSINT can be also considered in this context as a source of information for tracebacks and investigations.

Additionally, it is important to note that the utilization of public data has also compromising issues. There is a strong ethical component which is linked to the user's privacy. In particular, the profiling of people [15] could reveal personal details such as their political preference, sexual orientation or religious beliefs.

This article addresses the current state of OSINT since, to the best of our knowledge, there is no published work that integrates the recent advances of this paradigm, the opportunities it offers and the existing facilities to support OSINT processes. Specially, we study the spread and employment of OSINT techniques in Spain.

Furthermore, our purpose is to stimulate researches and advances in OSINT. As we have seen so far, OSINT is a promising mechanism that concretely improves the traditional cyber intelligence and cyber defense fields. However, there is still a long way to go to explore in this topic, and this article presents some future lines of research.

The remainder of this paper is organized as follows. Section II offers a review of recent research works in the field of OSINT. Section III discusses the motivation, pros and cons of the development of OSINT. Then, Sections IV and V describe some techniques and tools that facilitate searches through open data sources. Section VI contextualizes OSINT in Spain by describing some evidences of its usage and presenting certain Spanish public databases. Section VII poses some open challenges relative to research in OSINT. Finally, Section VIII concludes with some keys remarks, as well as future research directions.

II. State of the art

In recent years, with the advances of big data and data mining techniques, the research community has noticed that open data is a powerful source of analyzing social behaviors and obtaining relevant information [4].

With regards to the use of OSINT for **extracting social opinion and emotions**, Santarcangelo et al. [22] proposed a model for determining user opinions about a given keyword through social networks, specifically studying the adjectives, intensifiers and negations used in tweets. Unfortunately, it is a simple keyword-based solution only designed for Italian language not taking into account semantic issues. On the other hand, Kandias et al. [14] could relate people's usage of social networks (in particular, Facebook) to their stress level. However, the experiments were carried out only with 405 users, while nowadays there is a chance of processing much larger amounts of data.

In the context of **cybercrime and organized crime**, there are several works that explore the application of OSINT for criminal investigations. For example, OSINT could increase the accuracy of persecutions and arrests of culprits with frameworks like the proposed by Quick et al. in [21]. Concretely, authors apply OSINT to digital forensic data of a variety of devices to enhance the criminal intelligence analysis.

In this field, another opportunity that OSINT offers is the detection of illegal actions as well as the prevention of future crimes such as terrorist attacks, murders or violations. In fact, the European projects ePOOLICE [20] and CAPER [2] were designed to develop effective models for scanning open data automatically in order to analyze the society and detect emerging organized crime. In contrast to the previous mentioned projects, whose proposals were not practically used in real cases, Delavallade et al. [6] describe a model based on social networks data that is able to extract future crime indicators. Such model is then applied to copper theft and to jihadist propaganda use cases.

From the point of view of **cyber security and cyber defense**, OSINT represents a valuable tool for improving our protection mechanisms against cyber attacks. Pinto et al. [11] propose the use of OSINT in the Colombian context to prevent attacks and even to allow strategic anticipation. It includes not only plugins for collecting information, but also machine learning models to perform sentiment analysis. Moreover, the DiSIEM european project [7] maintains as a first goal the integration of diverse OSINT data sources in current SIEMs (*Security Information and Event Management*) to help reacting to recently-discovered vulnerabilities in the infrastructure or even predict possible emerging threats. Lee et al. [17] also designed an OSINT-based framework to inspect cyber security threats of critical infrastructure networks. However, all these approaches have not been applied to real world scenarios, thus their effectiveness remains questionable.

III. Background

The incredible growth of new technologies, services and social networks based on the Internet is putting information on the central axis of the world. In fact, a large part of it is publicly available, which means that anyone at any time in any place has access to this data.

Another phenomenon that is going on nowadays is the evolution from traditional criminal techniques to cybercrime. Extortion, fraud, identity theft or child exploitation are now carried out through the network, burglary has become hacking and fraudulent calls are recently known as phishing.

Fortunately, the good news is that almost every cybercriminal uses the Internet not only for illegal actions, but also for personal purposes. Leveraging this fact, OSINT seeks to connect both issues through the analysis of public data to produce cyber intelligence.

From a technical point of view, as we can see in TABLE I, OSINT exposes a number of benefits although it has also to deal with some restrictions. Regarding the positive points, we could highlight the following:

- Huge amount of worthwhile open source data to be analyzed, crossed and linked [23]. It includes social networks, public government documents and reports, online

Pros	Cons
Huge amount of public information	Complexity of data management
High capacity of computing	Unstructured information
Big data and machine learning	Misinformation
Complementary types of data	Data sources reliability
Flexible purpose and wide scope	Strong ethical/legal considerations

TABLE I
OSINT PROS AND CONS

multimedia content, newspapers and even the Deep web and Dark web, among others. The latest commented sources are especially interesting for OSINT [13]. Both the Deep Web and the Dark Web (the latter circumscribed within the former) contain even more information than the Surface Web (the Internet known by most users). In order to be able to access these networks, it is necessary to use specific tools since their contents are not indexed by traditional search engines.
Unlike the Surface Web and most of the Deep Web, the Dark Web offers anonymity and privacy to users who utilize it. This property makes criminals use this network to surf, conduct their searches and publish for illegitimate purposes while hiding their identity. Therefore, the Dark Web is an ideal source to apply OSINT and fight against cybercrime, organized crime or cyber threats. On the other hand, the persecution and de-anonymization of these people are a challenge for OSINT to work.

- Powerful computing capacity to mix large sets of data, relationships and patterns from different types of open sources. In particular, it allows the creation of complex inferences that are naturally unpredictable to humans [9].
- Emerging proliferation of big data and data mining techniques, as well as machine learning algorithms, which can automate and make investigation processes and decision making more intelligent and efficient [9].
- Possibility of completing OSINT with other types of information [5]. The system's inherent structure is open enough to accept also classified data or citizen collaboration within its engine.
- Generic implementation that allows different kinds of targets and several paths of exploration. As a consequence, OSINT applications could monitor suspicious people or dangerous groups, detect influence profiles related to radicalization, study worrying trends of the society, support the relative attribution of cyber attacks and crimes, etc [1].

However, using open source data also presents disadvantages which need to be considered as well:

- The quantity of data is immeasurable and, logically, it is challenging to handle it efficiently and effectively [8].
- The public information available on the Internet is inherently unstructured. This means that the data collected by OSINT is so heterogeneous that makes it difficult to classify, link and examine it in order to extract relationships and knowledge [3].
- Social networks and communication media are flooded with subjective opinions, *fake news* and canards [3]. For this reason, the existence of misinformation has to be considered in the implementation of OSINT mechanisms. The reliability and authority of the information are indeed the key to success.
- Ideally, the collected data should come from authoritative and reviewed sources (official documents, scientific reports, reliable communication media) [8]. In practice, OSINT will also deal with subjective or non-authoritative sources, as it could be the content of social networks or manipulated media.
- Ethical and legal considerations are fundamental in the development of OSINT. The results should respect user's privacy and not reveal intimate and personal issues [15]. In fact, the scope of the searches should be, by definition, limited to open data sources.

Since we can not allocate the police within each possible communication of the world, there is still an opportunity of using the public data to detect anomaly patterns and malicious behaviours. How calm would the cybercriminals be if, not only every single step of their telematic actions, but also their daily life, were relevant clues for investigators?

IV. OSINT TECHNIQUES

As it has been shown, OSINT is quite promising and powerful, but its implementation is also challenging. Thus for instance there are several manual techniques that provide public data to the end user, as we will see next.

A. Search engines

Everyone knows of the existence of *Google*, *Bing* or *Yahoo* search engines, among others. The traditional use of them is the simplest way of applying OSINT.

Moreover, services like *Google* support filters to refine searchs [1]. For instance, the use of *""* permits exact-matches, *OR* and *AND* act as logical operators, or * as a wildcard. It also allows the introduction of conditions like *filetype* to specify a certain file type, *site* to limit results to those from a specific website, or *intitle* to find pages with certain keywords within their title.

It is worth noting that, for example, a search in Google for DNIs (i.e., Spanish ID cards) within the *Region of Murcia* website outputs more than 15,000 results in less than half a second through the following query:

```
site:carm.es filetype:pdf intext:dni
```

B. Social networks

Nowadays, services like *Facebook* or *Twitter* have invaded our society. Any curious person has realized that lot of personal information can be found without advanced knowledge of these platforms. Thus, these applications offer precise search possibilities in the context of OSINT.

[1]https://support.google.com/websearch/answer/2466433?hl=en

Facebook permits specific queries by visiting elaborated URLs. For example, *www.facebook.com/search/facebook-id/search-token*, where *facebook-id* is the user identificator and *search-token* defines the criterion of the search (namely, *pages-liked*, *photos-liked*, *places-visited*, etc). *Twitter* not only supports advanced searches through URLs, but also implements a user-friendly interface in *https://twitter.com/search-advanced*.

Logically, these characteristics can be extended to the rest of social networks in some way.

C. Other OSINT services

There are other specific websites that offer relevant information given a certain kind of input:

- *Email address*: The website *hunter.io* returns whether an email address is valid or not, *haveibeenpwned.com* informs whether an email address has been hacked and *pipl.com* finds information related to the owner of such email address.
- *Username*: The service *knowem.com* checks the availability of a given username in social networks or domains.
- *Real name*: Apart from social networks, there are genealogy sites like *FamilySearch* or *GENi* that provide kinship information.
- *Location*: *Google Maps* or *Wikimapia* are well known sites to find out locations from GPS coordinates. On the contrary, it is also possible to get the GPS coordinates from a location name at *www.gps-coordinates.net*.
- *IP Address*: The service *www.iplocation.net* gets the location from a given IP address, whereas *viewdns.info* provides more technical information (*whois, reverse IP lookup, traceroute,* etc.).
- *Domain*: It is possible to visualize domain connections through *www.threatcrowd.org* or *www.visualsitemapper.com*. Furthermore, checking DNS and mailservers is also useful, by visiting *www.domaincrawler.com* or *who.is/dns*. There are also services like *www.alexa.com* and *www.similarweb.com* which calculate traffic statics and others like *findsubdomains.com* which search for subdomains. Finally, the site *web.archive.org* explores content within a number of archived domains.

It is clear that, by combining different techniques, it is possible to produce extremely useful knowledge about any connected target. Nevertheless, the scope of these resources have a general purpose and it is limited to specific fields. For that reason, researchers and developers have implemented more precise solutions for gathering better quality information.

V. OSINT TOOLS

Fortunately for people conducting OSINT activities, there are also more sophisticated tools that automate the collection of public information and infer interesting relationships. TABLE II presents the main features of the most popular and relevant open source OSINT tools today. Nevertheless, a complete view of the variety of OSINT resources can be displayed on the OSINT framework[2].

A. FOCA (Fingerprinting Organizations with Collected Archives)

This product[3], designed by *ElevenPaths*, analyzes the metadata of documents (Microsoft Office, PDF, Open Office, etc) available on the Internet. The software finds the hidden information, unifies it, and recognises the files that have been created in the same computer, or servers and clients that could be related to them. The server discovery module also includes more functionalities like web search, DNS search or IP resolution.

B. IntelTechniques

IntelTechniques consists in a website[4], created by Michael Bazzel, which offers hundreds of online search utilities. There are several modules divided by the target data that allow searching by email, social network profile, real name and user name, among others, in order to present to the end user the collected public information. It is a comprehensive tool that makes use of other simpler techniques, as the ones commented previously.

C. Maltego

Maltego is a well-known application[5] that finds public information within different sources about a certain target and presents it in the form of a directed graph for its analysis. Specifically, this tool infers advanced relationships (from data X to data Y) automatically with the so-called *transforms*. Although Maltego implements its generic *transforms*, it is also possible to implement and include custom ones for more specific purposes. For example, it would be very interesting to develop OSINT *transforms* for the Spanish context in order to take advantage of the existing open sources of Spain.

D. Metagoofil

Metagoofil is an information gathering tool[6] that extracts metadata of the files found for a specific domain or URL target. It is usually used for pentesting as it is able to reveal usernames, software versions and servers or machine names.

E. Recon-NG

Recon-NG is a web recognition framework[7] similar to Metasploit[8] which focuses its search depending on the loaded modules and the introduced input. It could obtain emails of the organization, locations, information of the administrator and users, *whois* information, etc.

[2]osintframework.com
[3]https://www.elevenpaths.com/es/labstools/foca-2
[4]https://inteltechniques.com
[5]https://www.paterva.com/web7/buy/maltego-clients.php
[6]https://github.com/laramies/metagoofil
[7]https://bitbucket.org/LaNMaSteR53/recon-ng/wiki/browse
[8]https://www.metasploit.com/

OSINT tools	Input				Output	Extensibility	Interface	Platform	Other features
	Identity Data	Network Data	File	Data Source					
FOCA	✗	Domain	File type	Google, Bing, Exalead	Metadata	✗	Program	Linux, Windows	Web, DNS and IP refeed
InterTechniques	Personal information, company, community	Domain, IP Address	File name, File type, File URL	Several	Multiple results	✗	Web interface	Online	Location, Public records, OSINT virtual machine
Maltego	Personal information, company, community	Domain	File URL	✗	Multiple results	Custom transforms	Program	Linux, Windows, MAC	Location, Auto input/ output refeed, Results in oriented graph
Metagoofil	✗	Domain	File type	✗	Metadata	✗	Command line	Linux, Windows	Limit of results
Recon-NG	Personal information	Domain	✗	Several	Multiple results	✗	Command line	Linux	Location, Modules for discovery and exploitation
Shodan	Country, City, Keyword, Hostname	Operating system, IP Address, Port	✗	✗	Network info	✗	Web interface	Online	Location, Webcam captures
Spiderfoot	Email	Domain, IP Address, Subnet	✗	Several	Multiple results	Custom modules	Web interface	Linux, Windows	Modules for discovery, Results in oriented graph
The Harvester	Company	Domain, DNS server	✗	Several	Network info	✗	Command line	Linux, Windows, MAC	Results in reports, Limit of results

TABLE II
MAIN FEATURES OF THE SELECTED OSINT TOOLS

F. Shodan

Shodan is a search engine[9] that provides public information of Internet-connected nodes, including *IoT* devices. The recollection of information is made through protocols like HTTP or SSH, so it allows search filters such as IP address, country name or even keywords. In general, it is used for network purposes, as it could be the monitoring of the network security or exploring network topologies.

G. Spiderfoot

Similar to Maltego, Spiderfoot is a reconnaissance tool[10] that automatically goes through lots of public data sources to compile intelligence related to IP addresses, domain names, e-mail addresses, names and more. Given the target, Spiderfoot uses the selected modules (equivalent to *transforms*) to perform its analysis. The results are represented in a node graph with all the found entities and relationships. In this case it is also possible to define our own modules.

H. The Harvester

This software[11] allows the collection of public information relative to a domain or company name. In particular, it is capable of listing emails of the company or hosts related to the domain. It also permits user-friendly HTML/XML representations of the results.

[9]https://www.shodan.io
[10]https://www.spiderfoot.net
[11]https://github.com/laramies/theharvester

I. OSINT tools comparison

Depending on the user needs (see TABLE II), some tools will be more suitable than others for a given task.

If we want to extract **hidden information from files**, *FOCA* and *Metagoofil* are specific tools designed for this purpose. In particular, the first product seems to be more complete than the second one, in the sense that it is able to infer more information from the metadata.

If we are looking for **network-focused information**, *Shodan* and *The Harvester* are interesting options for this certain task. However, we would recommend *Shodan* as it permits wider variety of inputs, it offers a user-friendly interface and it does not require installation.

Finally, if the aim of the search is to gather **as much information as possible** for a given input, the resources *InterTechniques, Maltego, Recon-NG* and *Spiderfoot* will return diverse data and relationships. Among them, the most sophisticated ones would be *InterTechniques* and *Maltego*. The first website offers different types of search which will operate through a very large number of data sources, but it is not as integrated as *Maltego*. In fact, this last tool implements automated inference processes between inputs and outputs that raise the scope of the original search. Moreover, it is extensible with custom discovery procedures.

Logically, although this comparison has been made according to the desired output, in practice the user will be limited by the available input and the data type accepted by OSINT tools.

Finally, note that these tools are complementary, meaning that a deep OSINT investigation could profit from all of them.

VI. OSINT in Spain

Intelligence services have been traditionally associated with the labour of Law Enforcement Agencies (LEAs) and Military Bodies. In the same way, OSINT is considered nowadays as an important key of classified investigations and secret operations in state affairs [1].

As far as we were able to explore in the official websites, reports and documentation, government organizations seem to implement internal mechanisms which basically consist in gathering raw information and transforming it into useful knowledge. In a representative way, we could mention the United Estates Federal Bureau of Investigation (FBI), United States Central Intelligence Agency (CIA), Canadian Security Intelligence Service (CSIS), EUROPOL, North Atlantic Treaty Organization (NATO), US DA (Department of Army) or NPIA (National Policing Improvement Agency) of England.

In Spain, the situation is quite similar. It is not easy to find clear evidences of the application of OSINT by the state forces. The confidentiality of this type of agencies makes it difficult to discover their internal operating mode and the impact of OSINT in their current investigations. Nevertheless, as a result of a deep search, we have some subtle findings that confirm, indeed, that OSINT is currently used by Spanish LEAs:

- Yet in 2007, the director of the CNI (i.e., Spanish National Intelligence Agency) said[12] that open sources were "*fundamental to the elaboration and work of Intelligence*".
- CIFAS (i.e., Spanish Military Intelligence Agency) also seems to use OSINT as a way of obtaining information. We have found some slides that confirm this, dated in 2008, which are uploaded in the Spanish Defense Staff website[13].
- In 2010, when the director of the CNI announced[14] the creation of an ethical code for special agents, he also insisted on the fact that modern intelligence was not just based on physical presence, as today "*you might get more information sitting on a computer, exploring messages from the bad guys*".
- In 2017, the Spanish Ministry of Defense opened a public call[15] for the contract called "*Development of OSINT tool based on IDOL HAVEN platform*".
- In the present, the Spanish Army is designing a new model called *Brigade 2035*[16] which incorporates innovative technological advances for enhancing operations. In this project, one of the defined combat functions is *Intelligence*, which clearly states OSINT as a key responsibility: "*Other facilities of growing importance will be open source obtainment (including social networking)...*".
- The Spanish Ministry of the Interior has published in the Annual Recruitment Plan for 2019[17] some investments in "*systems for obtaining OSINT in the cyberspace*"

Bearing in mind all these points, it seems that currently OSINT is indeed relevant in the internal affairs of Spain. In addition, note that to be effective, this paradigm depends on the public data available on the Internet, among other sources. In this regard, apart from social networks and other open data sources, there are authoritative Spanish sites where public information is published.

According to the European Data Portal and its official reports[18] about Open Data maturity across Europe, Spain is one of the most advanced countries in transparency and open data. In fact, it has been in first or second position in the ranking of Open Data Maturity in the last four years. As it is indicated, the Spanish Government has promoted more than 160 open data initiatives and has over 23,800 public information catalogues. For example, the Open Data Initiative of the Government of Spain[19] is a clear proof of how Spain encourages transparency. OSINT could benefit from that, but it should deal with aggregated and statistical information by linking it and inferring new knowledge.

However, it would be more interesting to analyze governmental platforms which are not anonymized. For instance, the Spanish Ministry of the Treasury, the Spanish Ministry of the Interior or the Spanish Ministry of Defense usually publish documents with personal information (*site:hacienda.gob.es filetype:pdf intext:dni*, for example). In the same way, this could be also applied to Spanish Autonomous Communities websites.

Moreover, Europe has also a public data platform[20] where we could find a lot of public information. For instance, in the context of foreign policy and security, an updated list of financial sanctions is presented in the "*European Union Consolidated Financial Sanctions List*" document. In particular, it reveals personal information about individuals, groups and entities.

All the aforementioned facts demonstrate that Europe, and especially Spain, are adopting strong Open Data policies. As a direct consequence, the amount of objective data available on the Internet is rapidly increasing. OSINT should, in addition to other open sources of information, take advantage of this powerful opportunity to collect, analyze, link and infer knowledge from reliable and official sources.

VII. Open challenges

After a review of the existing OSINT techniques, tools and status, it is also necessary to enumerate some gaps of this

[12] https://www.elconfidencialdigital.com/articulo/vivir/CNI-califica-fundamental-abiertas-contradice/20071023000000049386.html

[13] http://www.emad.mde.es/Galerias/EMAD/novemad/fichero/EMD-CIFAS-esp.pdf

[14] https://www.lavanguardia.com/politica/20100624/53951898847/el-director-del-cni-anuncia-un-codigo-etico-para-los-agentes-secretos.html

[15] https://contrataciondelestado.es/wps/wcm/connect/ff96fa82-7fd6-40bd-be5b-36ef3fd4e65b/DOC_CN2017-498874.pdf?MOD=AJPERES

[16] http://www.ejercito.mde.es/estructura/briex_2035/principal.html

[17] http://www.defensa.gob.es/Galerias/gabinete/ficheros_docs/2019/PACDEF_2019_Documento_Pxblico.pdf

[18] https://www.europeandataportal.eu/en/dashboard#2018

[19] https://datos.gob.es/es

[20] http://data.europa.eu/euodp/en/data

paradigm. Although OSINT seems to be ideal, in reality it is necessary to make it more sophisticated and applicable to uncontrolled scenarios of the real world. As far as we know, there are some challenges that are not solved yet and should be faced by the research community:

A. Propagation of the gathering process

With the development of big data and data mining techniques, it should not be a problem to avoid collecting data in a manual manner. Although OSINT techniques (Section IV) and tools (Section V) improve this, they work with single and basic explorations. In this sense, it would be appealing to implement refeed mechanisms by concatenating searches from outputs to inputs. As a consequence, the original search would also have, not only direct inferences, but also indirect and not explicit relationships.

B. Integration of several open data sources

OSINT activities should consult as many sources as possible in order to cover the widest possible spectrum. This means that the system has to normalize the gathered information, which is typically unstructured, in order to perform an effective analysis. In this context, it is important to filter repeated items.

C. Detection of irrelevant data and misinformation

Due to the huge amount of data publicly available, an OSINT process needs to be capable of distinguishing the relevance of each piece of information, discarding data which do not add value. Furthermore, it is crucial to detect misinformation that would corrupt the results. In fact, it would be interesting to analyze information as well fake with the aim of extracting intelligence.

D. Extension across the whole world

One of the main drawbacks of the existing OSINT resources is that they are usually oriented to specific countries. As a result, they are leaving aside other interesting open data sources from different territories. Taking this negative issue into account, interoperability is a desirable property to be considered in OSINT design. Note that it would increase the scope of the searches and the usage by end-users.

In Spain, for instance, we use tools that are designed in (and for) foreign countries. However, there are not OSINT solutions which include Spanish public repositories in the gathering phase (as government open data platforms could be). In this sense, we are not benefiting from the goldmine that supposes being one of the most transparent countries of Europe.

E. Enhancement of the analysis process

OSINT analysis is not as intelligent as it could be. The existing tools are limited to throwing all the information found and its relationships. However, the analysis process should incorporate semantic analysis, study of patterns, correlation with other events, occurrences or datasets. Ideally, the OSINT of the future should be able to provide the end user with the specific piece of information he/she is searching, as well as to return convincing answers in investigations.

F. Awareness of ethical and legal considerations

The use of OSINT should be restricted to legal activities and non-malicious purposes. To achieve this, OSINT has to be designed respecting the user's privacy and data protection laws. Furthermore, OSINT tool developers should also take into account that the end-user could be a delinquent trying to commit a crime. For this reason, the use of the most powerful tools should be limited to LEAs and Intelligence Services.

G. Summary

All the abovementioned challenges build the path between the Second Generation and the Third Generation of OSINT. As it is presented in [24], the Second Generation started with the rise of Internet and Social Media, and the challenges were "*technical expertise, virtual accessibility and constant acquisition*". In contrast, the evolution to the Third Generation is supposed to appear nowadays and will have to include "*direct and indirect machine processing of data, machine learning, and automated reasoning*".

VIII. Conclusions and Future Work

OSINT is changing the traditional intelligence processes into an automated procedure capable of taking investigations to all parts of the world. In fact, it is not only available for Law Enforcement Agencies (LEAs) and Intelligence Services, but also for curious people without technical training. However, there is still a lack of serious approaches for transforming OSINT into a robust and self-managed solution.

The paper described the status of this paradigm today. It revealed that the effectiveness of current works is questionable due to their poor application in real scenarios. The article also presented some OSINT techniques for basic searches and described the most sophisticated OSINT tools for advanced investigations.

In the context of Spain, we pointed out some indications which might confirm that Spanish LEAs use OSINT in their internal procedure. Furthermore, we categorized Spain as a goldmine due to its Open Data maturity. Actually, it is one of the highest one of Europe according to the European Data Portal.

Finally, the article outlined some open challenges related to gathering, analyzing and extracting real knowledge from the immersion of the internet. The future directions could address such challenges by including advanced techniques in OSINT processes in order to improve the current performance. To this extent, the OSINT ultimate goal is to be able to ensure the desired finding for a certain purpose, in an automated and a self-driven way.

Acknowledgment

This work has been supported by a Leonardo Grant 2017 for Researchers and Cultural Creators awarded by the BBVA Foundation; and by a Ramón y Cajal research contract (RYC-2015-18210) granted by the MINECO (Spain) and co-funded by the European Social Fund.

REFERENCES

[1] B. Akhgar, P. S. Bayerl, and F. Sampson. *Open Source Intelligence Investigation: From Strategy to Implementation*. Springer Publishing Company, Incorporated, 1st edition, 2017.

[2] C. Aliprandi, J. Arraiza Irujo, M. Cuadros, S. Maier, F. Melero, and M. Raffaelli. Caper: Collaborative information, acquisition, processing, exploitation and reporting for the prevention of organised crime. In C. Stephanidis, editor, *HCI International 2014 - Posters' Extended Abstracts*, pages 147–152, Cham, 2014. Springer International Publishing.

[3] G. Bello-Orgaz, J. J. Jung, and D. Camacho. Social big data: Recent achievements and new challenges. *Information Fusion*, 28:45 – 59, 2016.

[4] H. Chen, R. H. L. Chiang, and V. C. Storey. Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, 36(4):1165–1188, 2012.

[5] T. Day, H. Gibson, and S. Ramwell. *Fusion of OSINT and Non-OSINT Data*, pages 133–152. Springer International Publishing, Cham, 2016.

[6] T. Delavallade, P. Bertrand, and V. Thouvenot. Extracting Future Crime Indicators from Social Media. In *Using Open Data to Detect Organized Crime Threats*, pages 167–198. Springer International Publishing, Cham, 2017.

[7] DiSIEM project. Diversity Enhacements for Security Information and Event Management Project: http://disiem-project.eu/.

[8] C. S. Fleisher. Using open source data in developing competitive and marketing intelligence. *European Journal of Marketing*, 42(7/8):852–866, 2008.

[9] A. Gandomi and M. Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137 – 144, 2015.

[10] F. Gómez Mármol, M. Gil Pérez, and G. Martínez Pérez. I don't trust ict: Research challenges in cyber security. In S. M. Habib, J. Vassileva, S. Mauw, and M. Mühlhäuser, editors, *Trust Management X*, pages 129–136, Cham, 2016. Springer International Publishing.

[11] M. J. Hernández, C. C. Pinzón, D. O. Díaz, J. C. C. García, and R. A. Pinto. Open source intelligence (OSINT) as support of cybersecurity operations. Use of OSINT in a colombian context and sentiment Analysis. *Revista Vnculos: Ciencia, tecnologa y sociedad*, 15:29–40, 2018.

[12] J. Jang-Jaccard and S. Nepal. A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, 80(5):973–993, 2014.

[13] G. Kalpakis, T. Tsikrika, N. Cunningham, C. Iliou, S. Vrochidis, J. Middleton, and I. Kompatsiaris. *OSINT and the Dark Web*, pages 111–132. Springer International Publishing, Cham, 2016.

[14] M. Kandias, D. Gritzalis, V. Stavrou, and K. Nikoloulis. Stress level detection via OSN usage pattern and chronicity analysis: An OSINT threat intelligence module. *Computers & Security*, 69:3–17, aug 2017.

[15] M. Kandias, L. Mitrou, V. Stavrou, and D. Gritzalis. Which side are you on? A new Panopticon vs. privacy. In *2013 International Conference on Security and Cryptography (SECRYPT)*, pages 1–13, July 2013.

[16] H. L. Larsen, J. M. Blanco, R. Pastor Pastor, and R. R. Yager, editors. *Using Open Data to Detect Organized Crime Threats*. Springer International Publishing, Cham, 2017.

[17] S. Lee and T. Shon. Open source intelligence base cyber threat inspection framework for critical infrastructures. In *2016 Future Technologies Conference (FTC)*, pages 1030–1033. IEEE, dec 2016.

[18] S. C. Mercado. Sailing the Sea of OSINT in the Information Age. *Journal of the American Intelligence Professional*, 48(3), 2004.

[19] P. Nespoli, D. Papamartzivanos, F. Gómez Mármol, and G. Kambourakis. Optimal Countermeasures Selection against Cyber Attacks: A Comprehensive Survey on Reaction Frameworks. *IEEE Communications Surveys and Tutorials*, 20(2):1361–1396, 2018.

[20] R. P. Pastor and H. L. Larsen. Scanning of Open Data for Detection of Emerging Organized Crime ThreatsThe ePOOLICE Project. In *Using Open Data to Detect Organized Crime Threats*, pages 47–71. Springer International Publishing, Cham, 2017.

[21] D. Quick and K.-K. R. Choo. Digital forensic intelligence: Data subsets and Open Source Intelligence (DFINT+OSINT): A timely and cohesive mix. *Future Generation Computer Systems*, 78:558–567, jan 2018.

[22] V. Santarcangelo, G. Oddo, M. Pilato, F. Valenti, and C. Fornaro. Social opinion mining: An approach for italian language. In *2015 3rd International Conference on Future Internet of Things and Cloud*, pages 693–697, Aug 2015.

[23] B. L. William Wong. *Fluidity and Rigour: Addressing the Design Considerations for OSINT Tools and Processes*, pages 167–185. Springer International Publishing, Cham, 2016.

[24] H. J. Williams and I. Blum. Defining Second Generation Open Source Intelligence (OSINT) for the Defense Enterprise, 2018.

Evaluación de características de fuentes de datos en ciberseguridad para su aplicabilidad a algoritmos de aprendizaje basados en redes neuronales

Xavier A. Larriva-Novo, Mario Vega-Barbas, Víctor A. Villagrá, Mario Sanz
Universidad Politécnica de Madrid (UPM). DIT, ETSI Telecomunicaciones. Avda. Complutense, 30. 28040 Madrid
xlarriva@dit.upm.es, mario.vega@upm.es, victor.villagra@upm.es, msanz@dit.upm.es

Resumen- **Los algoritmos de inteligencia artificial ya tienen un papel protagonista en el ámbito de la ciberseguridad y la detección de ataques, pudiendo presentar mejores resultados en algunos escenarios que sistemas de detección de intrusiones clásicos como Snort o Suricata. Dentro de los algoritmos de aprendizaje automático, este artículo se centra en la evaluación de la aplicabilidad de uno de los más populares: las redes neuronales. Para ello, se plantea en primer lugar una categorización para datasets de ciberseguridad que divide sus características en varios grupos. Haciendo uso de dicha división, este trabajo busca determinar qué modelo de red neuronal (multicapa o recurrente), función de activación y algoritmo de aprendizaje arroja valores más elevados de precisión en función del grupo de características del que se disponga. Asimismo, y con estos resultados, se pretende deducir qué tipo de características presentes en un dataset son más relevantes y representativas para la detección y así, hacer más ligera la carga computacional de la red.**

***Index Terms*- Dataset, Ciberseguridad, Aprendizaje Automático, Redes Neuronales**

Tipo de contribución: *Investigación*

I. Introducción

El incremento de la complejidad de los sistemas de información ha justificado, en cierta medida, la proliferación del uso y aplicación de métodos y técnicas de Inteligencia Artificial (IA) en el ámbito de la seguridad informática. En concreto, la IA ha tenido una mayor incidencia en la detección de software dañino o la detección de anomalías e intrusiones, generando nuevos módulos de soporte a las decisiones más eficientes y robustos [1]. Esto favorece, entre otras cosas, que la interacción humana se pueda centrar en acciones más abstractas como puede ser la supervisión general de los sistemas o el análisis de errores, i.e. falsos positivos. Además, las técnicas de IA también ayudan a las personas encargadas de la seguridad informática a gestionar y analizar la ingente cantidad de datos que los nuevos sistemas de información son capaces de generar.

Como se ha mencionado, uno de los usos más habituales de la IA es la generación de sistemas de detección de intrusiones (IDS) [2], que completan IDS clásicos como pueden ser Snort o Suricata. Dichos sistemas manejan un gran volumen de datos que deben ser evaluados de forma rápida y actuar conforme a dicha evaluación para minimizar los riegos. Además del desarrollo de nuevos y más eficaces IDS, la IA se ha utilizado como base de implementación de sistemas de detección de intrusiones mediante técnicas de aprendizaje automático para la categorización de patrones mediante modelos tanto explícitos como implícitos [3]. Estas técnicas ofrecen también una capacidad de adaptación elevada ante la adhesión y procesamiento de nueva información.

Entre todas las técnicas de aprendizaje automático disponibles, este trabajo de investigación se ha centrado en el estudio de modelos computacionales basados en redes neuronales. Así, el objetivo principal de este trabajo es determinar qué modelo de red neuronal ofrece mejores resultados de análisis para diferentes tipos de datos propios del contexto de la seguridad de la información. Es decir, teniendo en cuenta que dependiendo del escenario en el que nos encontremos dispondremos de un tipo de datos concreto, este trabajo pretende mostrar qué conjunto de parámetros de una red neuronal favorecen la creación de mecanismos de detección que proporcionen una respuesta óptima. En concreto, este trabajo se ha centrado en el estudio y comparación de redes neuronales multicapa y recurrentes, con especial atención a datos de carácter temporal. Por último, se aborda la categorización de un dataset para determinar qué tipo de datos son más relevantes para la detección de un determinado tipo de ataque.

Para lograr el objetivo de este trabajo de investigación, el presente artículo se ha organizado como sigue. En primer lugar, la sección II presenta una perspectiva teórica de las redes neuronales (multicapa y recurrentes) y los parámetros de éstas que pueden ser modificados. A continuación, la sección III analiza los trabajos más relevantes que han optado por la aplicación de estos tipos concretos de redes neuronales para la detección de ciberataques. Las secciones IV y V ofrecen el estudio y justificación de la elección de los datos utilizados en esta investigación. Por último, las secciones VI y VII presentan los resultados y conclusiones de este trabajo.

II. Redes Neuronales

Las redes neuronales artificiales son sistemas complejos construidos mediante unidades computacionales simples llamadas neuronas, de forma análoga al comportamiento de las neuronas en los cerebros biológicos. Dichas neuronas están interconectadas entre sí a través de enlaces que gestionan el estado de activación de las neuronas adyacentes.

Cada una de las neuronas funciona de acuerdo a una función de activación, que relaciona la entrada con la salida de la misma. Las funciones de activación más comunes se detallan en la Tabla I. Las conexiones o pesos que conectan las neuronas se van actualizando de acuerdo al algoritmo de aprendizaje empleado, explorados en la sección II-B, cuyo propósito es disminuir el error entre la salida deseada y la obtenida.

Tabla I

FUNCIONES DE ACTIVACIÓN

Rectificador lineal *n*	**Sigmoide**
$y = x, x \geq n$	$y = \frac{1}{1+e^{-x}}$
Softsign	**Tangente hiperbólica**
$y = \frac{x}{1+\lvert x \rvert}$	$y = \tanh(x) = \frac{1}{1+e^{-x}} - 2$

A. *Tipo de redes neuronales*

Las redes neuronales más habituales son las denominadas *feedforward,* redes donde la señal viaja en un solo sentido (desde la entrada hacia la salida) y no hay bucles. Esto quiere decir que la salida de una capa no afecta a la misma capa.

Pueden estar compuestas de una por una capa de neuronas (monocapa) o varias (multicapa). Este estudio se centra únicamente en las redes multicapa (Fig. 1).

En contraposición a las redes neuronales *feedforward*, aparecen las redes neuronales recurrentes (RNR) donde las señales pueden viajar en ambas direcciones, introduciendo bucles en la red. Esto tiene como consecuencia que la salida de una capa puede afectar a esta misma capa, es decir, puede otorgarle a la red la propiedad de memoria. Por ello, este tipo de red neuronal es empleado para el modelado de series temporales o tareas [4].

El uso de las RNR es menor en comparación a las redes *feedforward*, en parte porque los algoritmos de aprendizaje son mucho menos eficaces (hasta la fecha). Sin embargo, se presentan como una alternativa muy interesante[5].

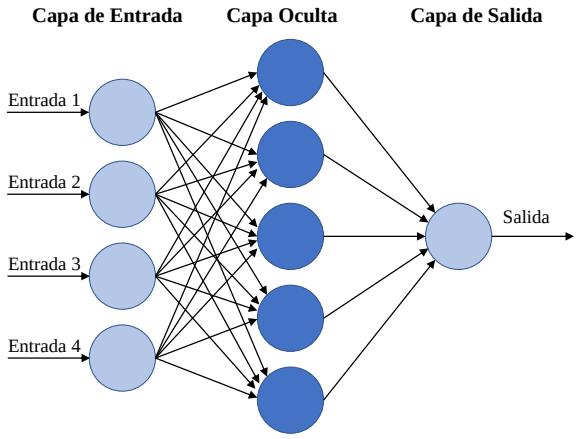


Fig. 1. Red Neuronal Multicapa.

Una de las arquitecturas de RNR más empleadas son las redes LSTM (Long Short-Term Memory) [4], que minimizan el problema del desvanecimiento del gradiente. La Fig. 2 presenta el esquema básico de una unidad de procesamiento de este tipo de redes neuronales.

La clave para entender el funcionamiento de estas redes son los valores C_{t-1} y C_t, que representan el estado de cada celda. Así, una celda puede mantener su estado en el tiempo (a través de la línea horizontal que conecta C_{t-1} y C_t); mientras regula el flujo de información entre la entrada y la salida a través de puertas no lineales.

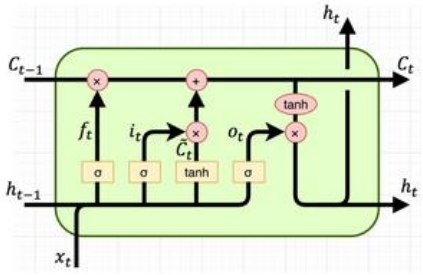


Fig. 2. Unidad LSTM

B. *Algoritmos de aprendizaje*

El aprendizaje es el proceso por medio del cual los grados de libertad de una red neuronal son adaptados a través de un proceso de estimulación por el entorno en el cual la red se encuentra inmersa, es decir, el proceso por el cual la red neuronal modifica sus pesos (conexiones entre neuronas) en respuesta a una información de entrada.

Los principales tipos de aprendizaje son el supervisado y el no supervisado. En el aprendizaje supervisado a la red se le proporciona un conjunto de ejemplos del comportamiento etiquetados de la red; en el no supervisado las entradas son la única fuente de información para el aprendizaje y la propia red aprende a categorizar las entradas.

Para el problema que se aborda en este artículo, nos centraremos únicamente en el aprendizaje supervisado de cara a la corrección del error, donde se pretende minimizar una función de coste basada en la señal de error, de modo que la respuesta de cada neurona de salida de la red se aproxime lo máximo posible a la respuesta objetivo. Los métodos estudiados para esta corrección del error son: *Descenso del gradiente* [6], *RMSProp (Root Mean Square Propagation)* [7] y *Adam* [8].

Los problemas más importantes relacionados con el aprendizaje en las redes neuronales recurrentes se agrupan entorno a dos conceptos, el desvanecimiento del gradiente (*vanishing gradient*) y el gradiente explosivo (*exploding gradient*) [9]. Los problemas relacionados con el *exploding gradient* se refieren a un gran aumento de la norma del gradiente durante el entrenamiento debido a la explosión de los componentes a largo plazo, que crecen de forma exponencial. Por su parte, los problemas de *vanishing gradient* tienen un comportamiento opuesto, es decir, los componentes a largo plazo disminuyen exponencialmente hasta alcanzar norma 0. Ambos problemas hacen imposible que el modelo pueda aprender la correlación entre eventos temporalmente distantes. En [10] se proponen varias soluciones para lidiar con estos problemas como son la reducción de escala de los gradientes cada vez que un umbral es superado (para problemas de *exploding gradient*) o utilizar un término de regularización que presenta una preferencia por algunos valores, lo que implica que los gradientes ni aumenten ni disminuyan en magnitud (para problemas de *vanishing gradient*).

III. EL USO DE LAS REDES NEURONALES EN LA SEGURIDAD INFORMÁTICA

La aplicación de redes neuronales artificiales al contexto de la seguridad informática esta principalmente enfocada a la

detección de intrusiones en una red ya que están consideradas como un enfoque eficaz de clasificación de patrones. El principal problema de estos algoritmos reside en los altos requisitos de cálculo y los largos ciclos de entrenamiento que requieren, obstaculizando su incorporación a aplicaciones comerciales [11].

Las redes neuronales artificiales, como se ha comentado anteriormente, han sido usadas en múltiples y diversos problemas relacionados con los sistemas de detección de intrusiones. Un ejemplo del rendimiento de una red simple puede hallarse en [12] donde se obtiene una precisión del 98,86% haciendo uso de una red neuronal de tres capas con propagación hacia atrás y realimentación. Esta precisión se asemeja a la conseguida por otros algoritmos (SVM, Naïve Bayes y C4.5). En [13] se compara también una red neuronal multicapa con un algoritmo SVM. Los resultados expuestos en este trabajo presentan valores de precisión similares, en torno al 99%.

En [14] se pueden encontrar redes más complejas, donde se hace uso de una red neuronal con tiempo de retardo (TDNN, Time Delay Neural Network) para el desarrollo de un IDS capaz de recoger las características de los paquetes de la red monitorizada. Estas características son agrupadas e introducidas a una red neuronal con tiempo de retardo que las clasificara haciendo saltar una alarma si fuera necesario. En este trabajo se comprueba, mediante la realización de diversas pruebas, que el sistema implementado detecta los ataques más rápidamente que mediante el uso de un sistema basado en reglas como Snort.

Otros trabajos como [15] comparan el uso de una red competitiva de aprendizaje mejorada (ICLN, *Improved Competitive Learning Network*) con el modelo de red neuronal de mapa auto organizado (SOM*, Self-Organizing Maps*). Las redes ICLN son usadas en el aprendizaje no supervisado mientras que el SOM es un modelo muy popular, totalmente conectado y con una sola capa usado en el aprendizaje supervisado. Tras ejecutar experimentos con ambas redes se obtiene una precisión similar, aunque la red SOM hace uso de una mayor cantidad de tiempo de procesamiento.

Otro tipo de arquitectura de redes neuronales artificiales popular para entornos de seguridad informática son las redes recurrentes. En [16] se presenta una arquitectura de IDS donde se hace uso de redes neuronales de tiempo diferido distribuido (DTDNN, *Distributed Time-Delay Neural Network*), que proporciona una manera simple y eficiente de clasificar conjuntos de datos gracias a su alta velocidad y las rápidas tasas de convergencia, con resultados satisfactorios. Otro tipo de arquitectura de red neuronal recurrente muy usada para el desarrollo de IDS son las llamadas memorias de largo-corto plazo (LSTM, Long Short-Term Memory), cuyo uso puede estudiarse en [17],[18],[19],[20]. En [17] se presenta una precisión del 97,54%, que está a la altura de otras arquitecturas de redes neuronales, pero tiene una tasa de falsos positivos del 9,98 %, bastante alta, aunque por debajo de la mayoría las otras arquitecturas de redes neuronales con las que se compara. Por otro lado, [18] presenta una arquitectura que arroja una precisión global de 93,72%, aunque para ataques de reconocimiento, los que se abordan en este trabajo, la precisión es muy baja (56,4 %). Por su parte, el trabajo realizado en [19] consigue una alta precisión en ataques del tipo DoS y conexiones normales, pero un bajo desempeño en ataques de reconocimiento, R2L y U2R. Por último, en [20] se presentan resultados que demuestran que el uso de redes recurrentes para tareas de clasificación de intrusiones es más preciso que con otros algoritmos de aprendizaje. Las redes neuronales, además de representar una unidad funcional independiente, permiten su combinación con otros algoritmos de aprendizaje automático para conseguir un mejor rendimiento. Esta forma de actuación se puede encontrar en [21], donde se hace uso del agrupamiento espectral y redes neuronales profundas (Deep Neural Network) para el desarrollo de un IDS.

Evidentemente, los datos suponen la pieza clave de todo algoritmo de aprendizaje automático debido a que serán la fuente de información del aprendizaje para posteriormente poder clasificar adecuadamente cada nueva entrada. Por esta razón, existen trabajos enfocados a la categorización de patrones de un dataset [22], ya sea centrados en su estudio o a la reducción de características, en el caso de datasets multidimensionales. Sin embargo, estos trabajos no presentan una categorización como la propuesta en esta investigación (Sección IV) y cuyo objetivo es mejorar la eficiencia, rendimiento y fiabilidad de la algorítmica.

IV. CATEGORIZACIÓN DE UN DATASET DE CIBERSEGURIDAD

El análisis de algunos datasets existentes (UNB-ISCX-2012 [23], CTU-13 [24], MACCDC [25] o UGR'16 [26]) permite observar que cuentan con formato y características diferentes, por lo que se puede decir que los datasets de ciberseguridad son altamente heterogéneos.

La metodología que se propone en este caso tiene como objetivo simplificar datasets multidimensionales, escogiendo sólo las características relevantes para el escenario especifico, y así hacer más ligero el algoritmo de aprendizaje (redes neuronales en este caso). En este trabajo se propone reducir dicha multidimensionalidad por grupos de características, en lugar de utilizar un enfoque individual. Para ello, se propone la siguiente clasificación de datos:

- **Características básicas de la conexion:** En esta categoría se incluyen las características básicas que se suelen encontrar en la cabecera TCP. Son características intrínsecas de una conexion y pueden resultar de utilidad para análisis de red de propósito general, y no únicamente para la detección de intrusiones. Ejemplos de estas características son la duración, el servicio, el protocolo o información sobre el origen y el destino de la conexión;
- **Características de contenido:** Estas características se refieren al contenido de los paquetes de la conexion que se está analizando. Es información más específica por lo que su uso está más orientado a la detección de ciertos ataques, en lugar de centrarse en la detección de anomalías en una red. Características que se clasificarían en esta categoría son, por ejemplo, el número de intentos de autenticación fallidos, la información sobre el acceso a una consola *root* o el número de operaciones de creación de ficheros;
- **Características de tráfico estadísticas:** Esta categoría engloba características que no son relativas a una sola conexion, sino información estadística relativa a una propiedad determinada [27]. Es decir, seleccionando una particularidad, como por ejemplo, un mismo host, estas propiedades estadísticas podrían ser el número de conexiones a ese host, o el porcentaje de conexiones a ese host que tienen el mismo servicio. En general, aportan más información que los anteriores grupos de

características. Esta categoría se divide a su vez en subcategorías dependiendo de que la característica estudiada. Algunas subcategorías propuestas son:

- **Características de tráfico basadas en el tiempo**, que son obtenidas en una ventana temporal de 2 segundos, teniendo en cuenta que los ataques de reconocimiento se basan en la generación de muchas conexiones en un periodo corto de tiempo. Pueden ser, por ejemplo, el número de conexiones al mismo host o el número de conexiones que tienen errores SYN durante la ventana de tiempo definida;
- **Características de tráfico basadas en dirección origen,** características referidas a información relativa al mismo host origen. En concreto el dataset analizado utiliza una ventana de 100 conexiones al mismo host en una determinado periodo de tiempo.
- **Características de tráfico basadas en la dirección destino.** Ídem al caso anterior pero agrupando la información en función del host destino. Por ejemplo, una posible característica para esta categoría es el número de conexiones al mismo servicio.

Algunas de estas características tendrán más o menos peso en función del tipo de ataque que se pretenda detectar. Por ejemplo, las características de tráfico basadas en tiempo tienen especial utilidad para detectar elevados volúmenes de datos en un intervalo pequeño de tiempo y, por consiguiente, posibles ataques de denegación de servicio.

V. DATOS UTILIZADOS

Para el problema que se aborda en este trabajo, la base de datos utilizada deberá contener información sobre distintas conexiones de una red junto con una etiqueta que especifique si la conexion es un ataque y su tipo o una conexion normal. El algoritmo utilizado para la detección hará uso del aprendizaje supervisado y, por lo tanto necesitará la etiqueta que clasifique el tipo de dato.

En este caso se ha escogido el dataset UNSW-NB15 [28], [29], ampliamente utilizado en ciberseguridad. La elección de esta base de datos está motivada por varios factores: la vigencia de los ataques y el etiquetado de estos, y la clasificación de los datos, similar a la presentada en la sección anterior.

La base de datos UNSW-NB15 contiene aproximadamente 2.540.046 conexiones simples, etiquetadas como normal o ataque, con 47 características cada una. Se han realizado varios esfuerzos orientados a reducir el número de características representativas de cada una de las conexiones sin que esto suponga una reducción de la precisión de la respuesta. Sucede así en [30], donde se emplea una red neuronal para comprobar la precisión obtenida en la detección tras la reducción de características utilizando técnicas de correlación o entropía. En este artículo se comprueba la precisión que se alcanza utilizando cada grupo de datos para inferir que tipo de características tienen más influencia en la detección del ataque que nos ocupa.

Los ataques modernos que se pueden encontrar en esta base de datos se pueden dividir en nueve categorías: *Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance (Reconosimiento), Shellcode, Worms*. En este trabajo nos centraremos en los ataques de reconocimiento, en parte por la gran cantidad de registros de este tipo presentes en la base de datos como por la diversidad de este tipo de ataques. Asimismo, orientar esfuerzos hacia las fases de reconocimiento de los ataques (que corresponden siempre con la primera fase de un ataque) supone un punto positivo de cara a la detección temprana de ciberataques y reaccionar lo antes posible cuando una organización se enfrenta a un ciberataque. Como se ha comentado anteriormente, cada conexion está definida por 49 características. Para la división de dichas características se ha seguido el esquema detallado en la sección IV.

Cada una de las divisiones propuestas han sido agrupadas para la realización de los experimentos de la siguiente forma: Grupo 1 o características básicas, Grupo 2 o características básicas y características de contenido, Grupo 3 o características básicas, características de contenido y características de tráfico basadas en el tiempo y Grupo 4 o características básicas, características de contenido, características de tráfico basadas en el tiempo, características de tráfico basadas en dirección origen y características de tráfico basadas en dirección destino. La Tabla II muestra un resumen de esta clasificación.

Esta división resulta útil para poder realizar diferentes pruebas con los diferentes grupos y comprobar que variables influyen más en el resultado final del algoritmo. Ciertas características, como *protocol, service* o *flag* no se presentan de forma numérica por lo que se ha recurrido a una codificación *one-hot* [31]. Además, algunas de las características como *duration* o *srcbytes* presentan datos con valores muy dispersos a lo largo de un rango numérico amplio, por lo que se normalizaran de la siguiente forma:

1. *Valor promedio (normalización Min-Max)*

$$x_i = \frac{x - Min}{Max - Min} \tag{1}$$

2. *Normalización Estática (Z-Score)*

$$x_i = \frac{x - \sigma}{\alpha} \tag{2}$$

Donde σ es el promedio y α es la desviación estándar del atributo.

VI. EXPERIMENTACIÓN Y DISCUSIÓN DE RESULTADOS

Para la implementación de las redes neuronales se ha utilizado como lenguaje de programación Python y la biblioteca *TensorFlow*, una biblioteca de código abierto creada por Google Brain Team. Dicha biblioteca ofrece todas las herramientas necesarias para construir, entrenar y comprobar la eficacia de redes neuronales artificiales.

Tabla II

DIVISIÓN DE LAS CARACTERÍSTICAS DEL DATASET UNSW-NB15

Grupo	Característica
Características básicas	srcip sport dstip dsport proto state dur sbytes dbytes sttl dttl sloss dloss service sload dload spkts dpakts
Características de contenido	swin dwin stcpb dtcpb smeansz dmeansz trans_depth res_bdy_len
Características de tráfico basadas en el tiempo	sjit djit stime ltime sintpkt dintpkt tcprtt synack ackdat
Características de tráfico basadas en dirección origen:	ct_srv_src ct_src_ltm ct_src_dport_ltm ct_src_dst_ltm
Características de tráfico basadas en dirección destino	ct_srv_dest ct_dst_ltm ct_dst_dport_ltm ct_dst_src_ltm

A lo largo de esta sección se presentan los resultados obtenidos tras las pruebas realizadas con los dos modelos de red neuronal, las funciones de activación, los diferentes algoritmos de aprendizaje y los diferentes grupos de características. En los distintos experimentos realizados se ha modificado tanto la función de activación de las neuronas como el optimizador.

La metodología seguida es la siguiente: fijar en primer lugar un optimizador (*Adam*) y obtener los resultados para las distintas funciones de activación. Con estos valores de precisión, se selecciona la mejor función de activación y se realizan experimentos con los optimizadores.

Las variables que deben ser monitorizadas para determinar el rendimiento de cada red son la precisión y el coste. Para analizar la precisión se han comparado los datos de test cuyas etiquetas de conexión han sido predichas y el verdadero valor de dichas etiquetas, consiguiendo el número total de aciertos por parte del algoritmo y obteniendo el porcentaje de precisión. En su caso, el coste se ha centrado en medir el error entre los datos de test cuyas etiquetas de conexión han sido predichas y el verdadero valor de dichas etiquetas mediante el cálculo de la entropía cruzada de la función exponencial normalizada; una vez obtenido este error realiza la media y obtiene un valor que se deberá ser reducido en la siguiente iteración de entrenamiento. Finalmente, los pesos de la red neuronal se inicializan con valores aleatorios.

A. Red neuronal multicapa

La red neuronal multicapa desarrollada de tres capas completamente conectadas, una capa de entrada, una capa oculta y una capa de salida. La distribución de neuronas de esta red en cada capa sigue el siguiente conjunto de reglas R_1,R_2,R_3 y R_4 [32], detalladas en la Tabla III.

Tabla III

MÉTODOS PARA EL CÁLCULO DE NODOS EN CAPAS OCULTAS

Código del método de calculo	Método
R_1	$H = 0.75 \times Entrada + Salida$
R_2	$H = \left(Entrada + Salida\right) / 2$
R_3	$H = 0.70 \times Entrada$
R_4	$H = 0.90 \times Entrada$

Para las pruebas realizadas sobre este desarrollo se han probado las combinaciones de función de activación y algoritmo de corrección del error mostradas en la Tabla IV.

Tabla IV

PRUEBAS REALIZADAS PARA CADA GRUPO DE CARACTERÍSTICAS USANDO LA RED NEURONAL MULTICAPA

Configuración	Normalización	Regla	Función de activación	Optimizador
m00	Min-Max	R_1,2,3,4	Rectificador lineal	Adam
m01		R_1,2,3,4	Sigmoide	Adam
m02		R_1,2,3,4	Tangente hiperbólica	Adam
m03		R_1,2,3,4	Softsign	Adam
m04		Mejor Regla	Mejor función de activación	Descenso gradiente
m05		Mejor Regla	Mejor función de activación	RMSProp
m01	Z-Score	R_1,2,3,4	Rectificador lineal	Adam
m02		R_1,2,3,4	Sigmoide	Adam
m03		R_1,2,3,4	Tangente hiperbólica	Adam
m04		R_1,2,3,4	Softsign	Adam
m06		Mejor Regla	Mejor función de activación	Descenso gradiente
m07		Mejor Regla	Mejor función de activación	RMSProp

Cada una de estas configuraciones han sido analizadas con cada grupo de características definido en la Tabla II, prestando especial atención al momento de máxima precisión y determinar, así, la mejor configuración para cada tipo de datos.

Para las pruebas realizadas sobre este desarrollo se han probado las combinaciones de función de activación y algoritmo de corrección del error mostradas en la Tabla IV. Cada una de estas configuraciones han sido analizadas con cada grupo de características definido en la Tabla II, prestando

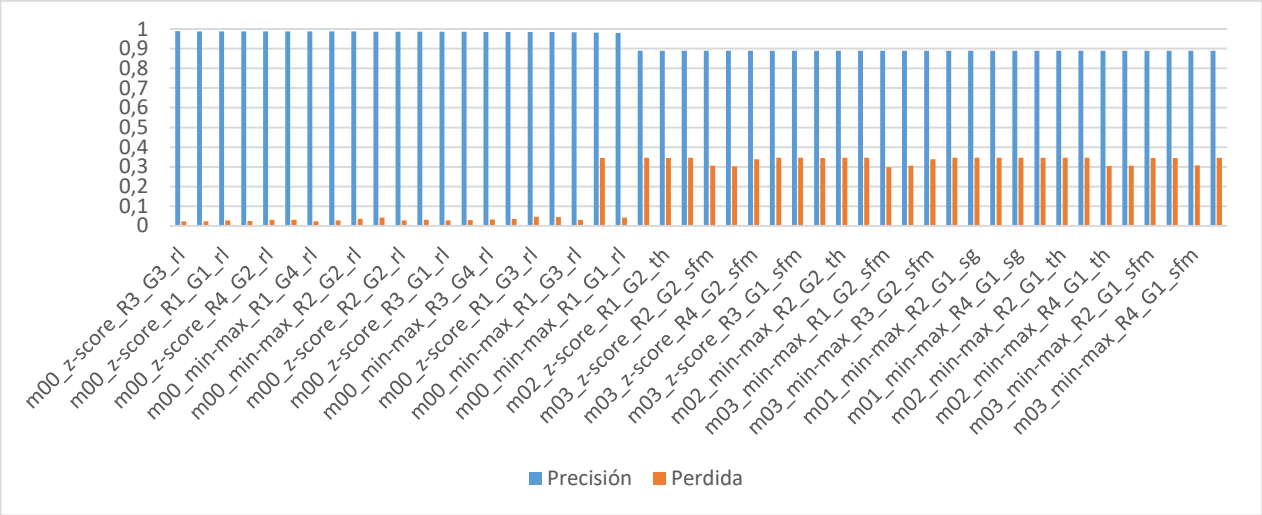


Fig. 3. Resultados Red Neuronal Feedforward.

especial atención al momento de máxima precisión y determinar, así, la mejor configuración para cada tipo de datos.

Los resultados para los experimentos realizados con uso del grupo de características 1, 2, 3 y 4 se detallan a continuación.

En primer lugar, para determinar la función de activación se realizan las pruebas m-00, m-01, m-02 y m-03; todas utilizando el optimizador *Adam*. Los resultados de este experimento se muestran en la Fig. 3, donde se aprecia que los mejores resultados (mayor precisión y convergencia más temprana) son arrojados por el uso de Rectificador lineal, para todos los grupos de características, obteniendo valores de aproximadamente del 98% en precisión utilizando la regla R_1 y función de normalización Z-Score para cada uno de los grupos de características. De igual manera, se puede observar la función de activación, la regla correspondiente para determinar el número de nodos específico para las capas ocultas, como la función de normalización más adecuada. Además, la Fig. 3 presenta las configuraciones más representativas realizadas y sus resultados, identificando las funciones y arquitectura más adecuada para los grupos de características propuestas.

A continuación, en la Fig. 4 se muestran los experimentos realizados una vez fijada la función rectificadora lineal, pudiendo apreciar que la precisión máxima se alcanza para cada una de los grupos de características del experimento m-00, ejecutado sobre cada uno de los grupos de características, y logrando una precisión promedio de 98.56%. Los resultados de estas pruebas se detallan en la Tabla V.

Para los datos pertenecientes al Grupo 2 la función de activación que mejor resultado arroja es el rectificador lineal, con una precisión de 98.8%. Fijada esta función de activación, los experimentos relativos a los optimizadores arrojan que la precisión más elevada es la obtenida con el *Adam* (valor indicado previamente), seguido de RMSprop, con una precisión de 98.18%. La principal desventaja presentada por ésta última es que la precisión no se mantiene estable, sino que se produce desvanecimientos de su valor a lo largo del resto del experimento. En cuanto al optimizador Gradiente Descendente, los valores de precisión disminuyen aproximadamente un 12%.

Por su parte, los experimentos realizados con el optimizador *Adam* sobre los datos pertenecientes al Grupo 3, aquellos datos relacionados con una ventana temporal indican que el mejor desempeño se obtiene utilizando la función rectificadora lineal como función de activación (98.43% de precisión).

Por último, para los datos pertenecientes al Grupo 4, es decir, datos de tráfico caracterizados mediante en base a la dirección del tráfico, los valores de precisión son peores que para los Grupos 1, 2 y 3. La mejor combinación se alcanza con el rectificador lineal como función de activación y el optimizador *Adam*.

Comparando los datos obtenidos, podemos afirmar que los mejores resultados se obtienen utilizando el optimizador *Adam* y la función rectificadora lineal como función de activación, aplicando una función de normalización de tipo Z-Score y R_1.

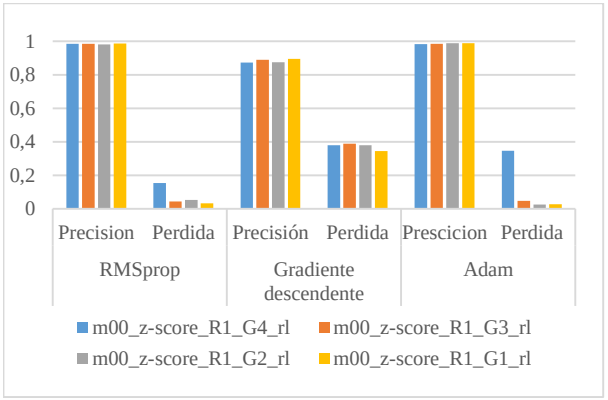


Fig. 4. Resultados en términos de perdida y precisión respecto a los optimizadores

B. *Red neuronal recurrente*

La red neuronal recurrente desarrollada está basada en una arquitectura LSTM. Se ha elegido este tipo de red debido a su capacidad de aprendizaje y los buenos resultados que han dado en otros proyectos similares. Esta red neuronal se compone de una neurona LSTM alimentada por un número determinado de conexiones. Dichas conexiones contienen, a su vez, un número variable de características. Las pruebas realizadas con esta red

se detallan en la Tabla VI, y los resultados correspondientes en la Tabla VII. Para el Grupo 1 se alcanza la precisión máxima de 98% usando el optimizador *Adam*; Los optimizadores que reducen el coste más rápidamente son *Adam* y RMSProp.

Tabla V

PRUEBAS REALIZADAS PARA CADA GRUPO DE CARACTERÍSTICAS USANDO LA RED NEURONAL MULTICAPA

Configuración	Precisión RMSprop	Precisión Gradiente descendente	Precisión Adam
m00_z-score_R1_G4_rl	0,987	0.895	0.988
m00_z-score_R1_G2_rl	0,981	0.873	0.988
m00_z-score_R1_G3_rl	0.985	0.889	0.9843
m00_z-score_R1_G4_rl	0.9838	0.8734	0.983

C. *Comparativa entre redes multicapa y recurrentes*

Otro de los objetivos de este trabajo es comparar el desempeño de las redes neuronales multicapa frente a las recurrentes. Para ello, se comparan las precisiones máximas obtenidas para cada grupo de datos.

En general no se un beneficio claro que justifique el uso de una red neuronal recurrente, pues las precisiones obtenidas son muy similares a las obtenidos mediante el uso de una red neuronal multicapa. Como se explica en [10], esto es debido a la complejidad inherente al entrenamiento correcto de una red neuronal recurrente. Sin embargo, el coste de computo ofrecido por las redes recurrentes es notablemente mayor, 9 veces mayor, al asociado a las redes multicapa.

Tabla VI

PRUEBAS REALIZADAS PARA CADA GRUPO DE CARACTERÍSTICAS USANDO LA RED NEURONAL RECURRENTE LSTM

Configuración	Optimizador
r-00	Adam
r-01	Descenso del gradiente
r-02	RMSProp

Los resultados de estos experimentos se muestran en la Tabla VII y Fig. 5. Para el grupo 1 de características se alcanza la precisión máxima usando el optimizador *Adam* (con un valor del 98 %); mientras para el grupo 2, 3 y 4 se alcanzan resultados similares Los optimizadores que reducen el coste más rápidamente son *Adam* y RMSProp.

Tabla VII

PRUEBAS REALIZADAS PARA CADA GRUPO DE CARACTERÍSTICAS USANDO LA RED NEURONAL RECURRENTE LSTM

Configuración	Grupo1	Grupo2	Grupo3	Grupo4
r00	0.98	0.983	0.984	0.984
r01	0,793	0.81	0.8723	0.798
r02	0.973	0.955	0.964	0.959

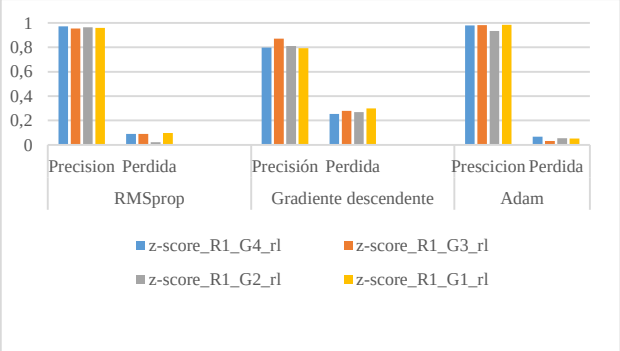


Fig. 6. Resultados en términos de perdida y precisión respecto a los optimizadores red neuronal recurrente

D. *Relevancia de los grupos de datos*

Tras realizar los experimentos descritos con la red neuronal multicapa sobre distintos grupos de datos, se observa que, utilizando únicamente un conjunto de características, es posible obtener buenos resultados de predicción. No obstante, en los experimentos realizados con el conjunto completo de datos de cada grupo de características, es posible observar un mejor desempeño y precisión con el grupo 1 y 2 con un valor del 99% mientras con los grupos 3 y 4 se obtiene una precisión aproximada del 98%. En cuanto a la red recurrente, los máximos resultados obtenidos de precisión en los experimentos son similares y cercanos a 98%.

E. *Arquitectura de la red neuronal y normalización de valores de entrada*

Cada red neuronal puede ofrecer diversos resultados dependiendo de su configuración, es decir, que estos resultados dependen de los nodos de su arquitectura, las funciones de optimización y la normalización de valores de entrada. En este caso, la regla que ha brindado una mayor precisión en cada uno de los grupos de datos expuestos es la R_1.

VII. CONCLUSIONES

En este trabajo se ha explorado la aplicación de las redes neuronales a la detección de intrusiones de ciberseguridad con dos objetivos principales. En primer lugar, la categorización de un dataset (UNSW-NB15), dividiendo sus características en: básicas, de contenido, estadísticas de tráfico y basadas en dirección; para analizar cuáles de estos grupos son los más relevantes para la detección de anomalías, aligerar el entrenamiento y reducir la pérdida de los modelos implementados [33]. El segundo objetivo se ha centrado en determinar qué red neuronal puede ofrecer un mejor desempeño según los datos que se disponga para su entrenamiento.

Los experimentos realizados, usando el dataset y la categorización propuesta, permiten extraer varias conclusiones. Por un lado, identificar los resultados óptimos para cada grupo de datos, según el tipo de red neuronal, la función de activación, la función de optimización y la arquitectura de red, como se detalla en las secciones VI-a y VI-b. Por otro lado, los resultados muestran que usando únicamente un grupo de datos se pueden obtener una predicción acertada del ataque, independientemente de la topología de red neuronal. Así, la configuración propuesta como *m00_z-score_R1_G1_rl* permite obtener una precisión

similar a la configuración *m00_z-score_R1_G4_rl*, aligerando la carga del algoritmo, en términos de rendimiento, pero con un menor número de características, como se detalla en VI-a.

En cuanto a la comparativa entre las diferentes arquitecturas de red neuronal analizada, no se observa una mejora sustancial al utilizar redes recurrentes en lugar de redes multicapa. Probablemente, esto se debe a la dificultad para entrenar una red recurrente.

Finalmente, como líneas futuras de investigación extraídas de este trabajo se propone, en primer lugar, extender la metodología propuesta a un dataset de ciberseguridad con mayor cantidad de información, como el propuesto por la Universidad de Granada en [26] (≈ 240M flujos de datos y tráfico real). Además, de cara a mejorar la métrica que estima el desempeño del algoritmo, es conveniente profundizar en el tipo de predicciones que se realizan y no obtener únicamente el porcentaje de la precisión.

REFERENCIAS

[1] B. Geluvaraj, P. M. Satwik, y T. A. Kumar, «The Future of Cybersecurity: Major Role of Artificial Intelligence, Machine Learning, and Deep Learning in Cyberspace», en *International Conference on Computer Networks and Communication Technologies*, 2019, pp. 739-747.

[2] A. Pannu, «Artificial intelligence and its application in different areas», *Artif. Intell.*, vol. 4, n.º 10, pp. 79-84, 2015.

[3] S. Dilek, H. Çakır, y M. Aydın, «Applications of artificial intelligence techniques to combating cyber crimes: A review», *ArXiv Prepr. ArXiv150203552*, 2015.

[4] S. Hochreiter y J. Schmidhuber, «Long Short-Term Memory», *Neural Comput.*, vol. 9, n.º 8, pp. 1735-1780, nov. 1997.

[5] D. S. Berman, A. L. Buczak, J. S. Chavis, y C. L. Corbett, «A Survey of Deep Learning Methods for Cyber Security», *Information*, vol. 10, n.º 4, p. 122, abr. 2019.

[6] L. B. Almeida, T. Langlois, J. D. Amaral, y A. Plakhov, «On-line Learning in Neural Networks», D. Saad, Ed. New York, NY, USA: Cambridge University Press, 1998, pp. 111–134.

[7] T. Tieleman y G. Hinton, «Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude», *COURSERA Neural Netw. Mach. Learn.*, vol. 4, n.º 2, pp. 26-31, 2012.

[8] D. P. Kingma y J. Ba, «Adam: A Method for Stochastic Optimization», *ArXiv14126980 Cs*, dic. 2014.

[9] Y. Bengio, P. Simard, y P. Frasconi, «Learning long-term dependencies with gradient descent is difficult», *IEEE Trans. Neural Netw.*, vol. 5, n.º 2, pp. 157-166, mar. 1994.

[10] R. Pascanu, T. Mikolov, y Y. Bengio, «On the difficulty of training recurrent neural networks», p. 9.

[11] N. Papernot, P. McDaniel, A. Swami, y R. Harang, «Crafting adversarial input sequences for recurrent neural networks», en *MILCOM 2016 - 2016 IEEE Military Communications Conference*, 2016, pp. 49-54.

[12] B. Subba, S. Biswas, y S. Karmakar, «A Neural Network based system for Intrusion Detection and attack classification», en *2016 Twenty Second National Conference on Communication (NCC)*, 2016, pp. 1-6.

[13] M. MORADI y M. ZULKERNINE, «A Neural Network Based System for Intrusion Detection and Classification of Attacks», p. 6.

[14] O. Al-Jarrah, «Network Intrusion Detection System Using Neural Network Classification of Attack Behavior», 2015.

[15] J. Z. Lei y A. Ghorbani, «Network intrusion detection using an improved competitive learning neural network», en *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.*, 2004, pp. 190-197.

[16] L. M. Ibrahim, «ANOMALY NETWORK INTRUSION DETECTION SYSTEM BASED ON DISTRIBUTED TIME-DELAY NEURAL NETWORK (DTDNN)», vol. 5, p. 15, 2010.

[17] T. Le, J. Kim, y H. Kim, «An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization», en *2017 International Conference on Platform Technology and Service (PlatCon)*, 2017, pp. 1-6.

[18] R. C. Staudemeyer, «Applying long short-term memory recurrent neural networks to intrusion detection», 01-jul-2015. [En línea]. Disponible en: https://www.ingentaconnect.com/content/sabinet/comp/2015/00000056/00000001/art00009. [Accedido: 05-mar-2019].

[19] J. Kim, J. Kim, H. L. T. Thu, y H. Kim, «Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection», en *2016 International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1-5.

[20] C. Yin, Y. Zhu, J. Fei, y X. He, «A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks», *IEEE Access*, vol. 5, pp. 21954-21961, 2017.

[21] T. Ma, F. Wang, J. Cheng, Y. Yu, y X. Chen, «A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks», *Sensors*, vol. 16, n.º 10, p. 1701, oct. 2016.

[22] M. H. Bhuyan, D. K. Bhattacharyya, y J. K. Kalita, «Network Anomaly Detection: Methods, Systems and Tools», *IEEE Commun. Surv. Tutor.*, vol. 16, n.º 1, pp. 303-336, First 2014.

[23] A. Shiravi, H. Shiravi, M. Tavallaee, y A. A. Ghorbani, «Toward developing a systematic approach to generate benchmark datasets for intrusion detection», *Comput. Secur.*, vol. 31, n.º 3, pp. 357-374, may 2012.

[24] S. García, M. Grill, J. Stiborek, y A. Zunino, «An empirical comparison of botnet detection methods», *Comput. Secur.*, vol. 45, pp. 100-123, sep. 2014.

[25] A. Carlin, D. P. Manson, y J. Zhu, «Developing the Cyber Defenders of Tomorrow With Regional Collegiate Cyber Defense Competitions (CCDC)», p. 10, 2010.

[26] G. M. Fernandez, J. Camacho, R. Magan-Carrion, P. Garcıa-Teodoro, y R. Theron, «UGR'16: A New Dataset for the Evaluation of Cyclostationarity-Based Network IDSs», p. 13.

[27] W. Lee y S. J. Stolfo, «A framework for constructing features and models for intrusion detection systems», *ACM Trans. Inf. Syst. Secur.*, vol. 3, n.º 4, pp. 227-261, nov. 2000.

[28] N. Moustafa y J. Slay, «UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)», en *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1-6.

[29] N. Moustafa y J. Slay, «The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set», *Inf. Secur. J. Glob. Perspect.*, vol. 25, n.º 1-3, pp. 18-31, abr. 2016.

[30] Akashdeep, I. Manzoor, y N. Kumar, «A feature reduced intrusion detection system using ANN classifier», *Expert Syst. Appl.*, vol. 88, pp. 249-257, dic. 2017.

[31] M. Cassel y F. Lima, «Evaluating one-hot encoding finite state machines for SEU reliability in SRAM-based FPGAs», en *12th IEEE International On-Line Testing Symposium (IOLTS'06)*, 2006, pp. 6 pp.-.

[32] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, y M. Rida, «A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection», *Comput. Secur.*, vol. 75, pp. 36-58, jun. 2018.

[33] and P. L. Bartlett y R. C. Williamson, «The importance of convexity in learning with squared loss», *IEEE Trans. Inf. Theory*, vol. 44, n.º 5, pp. 1974-1980, sep. 1998.

Investigación en Ciberseguridad: Una propuesta de innovación docente basada en el *role playing*

Noemí DeCastro-García
Dpto Matemáticas. Universidad de León
Campus de Vegazana s/n 24071 León
ncasg@unileon.es

Ángel Luis Muñoz Castañeda
Dpto Matemáticas. Universidad de León
Campus de Vegazana s/n 24071 León
amunc@unileon.es

Miguel V. Carriegos
Dpto Matemáticas. Universidad de León
Campus de Vegazana s/n 24071 León
miguel.carriegos@unileon.es

***Resumen*—El rápido crecimiento de las amenazas digitales, así como la aparición constante de retos tecnológicos, ponen de manifiesto la necesidad de incluir el desarrollo de competencias investigativas en los programas de formación en Ciberseguridad. De esta manera, el alumnado podrá adquirir habilidades que son fundamentales para mantener a la sociedad en la vanguardia del conocimiento.**

En este artículo se presenta una experiencia innovadora para la formación en investigación científica en el campo de Ciberseguridad. La metodología docente se basa en la creación de un entorno integral de investigación simulado y se ha diseñado en base a diferentes estándares internacionales potenciando la colaboración entre el sector académico y profesional. Además, la propuesta se ha implementado durante tres cursos escolares en el Master de Investigación en Ciberseguridad de la Universidad de León, tanto en modalidad *online* como presencial, presentando buenos resultados, así como posibilidades de mejora.

***Index Terms*—Competencias investigativas, Innovación, Ciberseguridad**

Tipo de contribución: *Formación e innovación educativa*

I. Introducción

La nueva sociedad del conocimiento requiere que la comunidad educativa tenga que reflexionar sobre qué procesos de enseñanza y aprendizaje son los más adecuados para desarrollar las capacidades que van a necesitar los futuros expertos del mundo digital. En particular, en el ámbito de la Ciberseguridad, la actualidad profesional se encuentra ligada a la aparición de problemas abiertos que implican retos enmarcados en contextos de Investigación, Desarrollo e Innovación (I+D+I), donde el conocimiento técnico es esencial, pero la adquisición de competencias relacionadas con la actividad investigadora es clave para poder resolver los problemas surgidos de una manera óptima y eficaz. Este panorama requiere que los nuevos talentos hayan desarrollado durante su formación competencias profesionales e investigativas que les hagan ser capaces de generar conocimiento siguiendo el quehacer actual de la comunidad científica y empresarial.

Actualmente existe una oferta abundante de cursos de formación en Ciberseguridad (programas reglados, cursos de extensión, cursos informales, etc), por lo que resulta necesario que la labor docente en el campo comience identificando qué contenidos y habilidades van a ser necesarios en las materias que se van a impartir basándose en el consenso, criterios y estándares de calidad que sientan las bases actuales de los programas nacionales e internacionales de formación en Ciberseguridad [1], [2]. Una vez se haya realizado este análisis, la hoja de ruta docente se ha de dirigir a proponer un currículo base, y a diseñar y llevar a cabo una propuesta metodológica adecuada.

En este contexto, el *Master de Investigación en Ciberseguridad* de la Universidad de León se caracteriza por ser un programa de especialización de orientación académica/mixta; es decir, con itinerario profesionalizador y de investigación. La titulación tiene una duración de dos cursos académicos e incluye en su segundo curso la asignatura *Investigación científica*. Esta materia tiene como finalidad potenciar el caracter investigador de la titulación (completando así la formación en Ciberseguridad que se imparte para que incluya todas las categorías de competencia establecidas en [1]), e integrar estos estudios de Máster en el Marco del Espacio Europeo de la Educación Superior del nivel de Posgrado (segundo ciclo de especialización) de acuerdo con el Programa de Bolonia.

Habitualmente, la aproximación metodológica a la formación en investigación aporta a los estudiantes conocimiento sobre metodologías de investigación que suelen estar centradas en el método científico. Sin embargo, las competencias investigativas necesarias en un campo de conocimiento dependen directamente de la naturaleza de la disciplina. Por ejemplo, los métodos de investigación son diferentes en el campo de las ciencias de la computación, en las ciencias sociales o naturales. El caso de la Ciberseguridad es aún más heterogéneo debido a la cantidad de disciplinas que están involucradas (matemáticas puras y aplicadas, desarrollo y análisis de software, ingeniería inversa, desarrollo y diseño de sistemas y redes, etc). Por lo tanto, tomando como referencia las competencias de investigación que queramos priorizar, se ha de determinar qué enfoques pedagógicos mantienen la coherencia entre el diseño, la metodología, la evaluación y los objetivos a alcanzar.

El objetivo principal de la metodología docente presentada en este artículo es desarrollar una experiencia de aprendizaje significativa e innovadora que fomente la adquisición de competencias de Investigación Científica en el campo de la Ciberseguridad.

Así mismo, se plantearon los siguientes objetivos específicos:

1. Identificar las competencias de investigación a desarrollar en un programa de formación en Ciberseguridad.
2. Diseñar y poner en práctica una metodología en la que se potencien competencias investigadoras que sean transversales a diversas disciplinas involucradas en el campo de la Ciberseguridad.
3. Establecer colaboraciones entre el sector académico y el sector empresarial.
4. Cumplir el mayor número posible de indicadores de in-

novación docente con la metodología puesta en práctica, [3].

La experiencia se ha realizado durante tres cursos escolares (2016/2017, 2017/2018 y 2018/2019) en la asignatura *Investigación Científica*, asignatura obligatoria de 6 ECTS del primer semestre del segundo curso del *Master de Investigación en Ciberseguridad* de la Universidad de León, tanto en la modalidad presencial, como en la opción *online*.

La experiencia de innovación ha introducido al alumnado en un entorno de investigación formativo integral enmarcado en una metodología de *Enseñanza orientada a la simulación de la investigación*. La implementación ha consistido en crear un entorno práctico de investigación para el alumnado de tal manera que este pueda adquirir y desarrollar diferentes competencias investigativas mediante un proceso de enseñanza y aprendizaje basado en el *role playing*.

Las competencias que se han priorizado son aquellas que se consideran clave en la profesión investigadora y que están relacionadas con la argumentación y el análisis, las habilidades en comunicación oral y escrita, la auto-regulación del aprendizaje (gestión del tiempo y de la información), y el desarrollo de competencias para la identificación y resolución de problemas.

El resto del artículo se organiza de la siguiente manera: en la Sección II se detallan los métodos utilizados en el estudio. A continuación, en la Sección III, se desarrollan los resultados. Finalmente, se incluyen las conclusiones y las referencias.

II. Métodos

II-A. Participantes

El colectivo de implementación ha constado de 40 estudiantes del segundo curso del *Master de Investigación en Ciberseguridad* de la Universidad de León, tanto en la modalidad presencial, como *online*. La experiencia se ha llevado a cabo en la asignatura *Investigación Científica* durante los cursos escolares 2016/2017, 2017/2018 y 2018/2019 (7, 9 y 23 estudiantes, respectivamente), con un 10 % de mujeres y un 90 % de hombres. Además, se tiene aproximadamente un 65 % de alumnado presencial y un 35 % de alumnado *online*. Cabe destacar que, aún habiendo más matriculados en la modalidad presencial del master, la mayoría de los estudiantes están trabajando y no acuden a todas las sesiones de clase.

II-B. Metodología

El estudio que se presenta se ha desarrollado en dos etapas. La fase inicial se ha llevado a cabo mediante una metodología exploratoria mixta en la que se han establecido las competencias y sub-competencias relevantes, así como el diseño del enfoque metodológico docente.

La segunda fase se ha realizado a través de una metodología mixta cuasi-experimental. La experiencia se ha desarrollado durante tres cursos escolares, lo que ha posibilitado poder realizar dos ciclos de investigación-acción, de tal forma que mediante los datos y sugerencias recogidos en las primeras intervenciones educativas, se han propuesto modificaciones que han supuesto una mejora en la calidad de la innovación. Finalmente, se ha analizado su implementación a través de diferentes indicadores.

II-C. Análisis

La experiencia se ha evaluado a través de diversos indicadores que implican diferentes instrumentos y participantes. Estos se incluyen en la Tabla I, y valoran diferentes dimensiones: rendimiento, innovación, aplicación en entornos virtuales y aspectos generales de la experiencia.

Tabla I
Indicadores para la evaluación del proceso de innovación

Dimensiones / Aspecto a evaluar	Instrumento
Tasa de rendimiento	
Calificaciones en la materia	Estadística descriptiva
Tasa de aprobados	Estadística descriptiva
Percepción de auto- adquisición de competencias investigadoras	Cuestionario
Proceso de innovación	
Coherencia entre diseño y objetivos de aprendizaje	Observación
Uso de tecnologías de la información y la comunicación.	Plataformas y TIC
Coordinación entre diferentes asignaturas e investigadores	Tasa de participación
Colaboración universidad-empresa	Tasa de participación
Duración de la experiencia	Número de intervenciones
Aplicación en entornos virtuales	
Comparativa de resultados modalidad presencial Vs *online*	Análisis inferencial
Aspectos generales de la experiencia	
La temporalización ha sido adecuada.	Observación
Los materiales han sido adecuados.	Observación

Se han calculado índices descriptivos básicos: análisis de frecuencias, medidas de tendencia central y de dispersión.Para analizar si las modalidades (presencial y *online*) son un factor que puede afectar a las calificaciones del colectivo de aplicación se han realizado pruebas inferenciales. En primer lugar, se ha aplicado la prueba de Kolmogorov-Smirnov ($N > 30$) que nos permite determinar la idoneidad de los contrastes utilizados. A continuación se ha empleado la prueba t de Student (contraste paramétrico) o la prueba U de Mann-Whitney (contraste no paramétrico) para 2 muestras independientes, dependiendo de la significación del contraste de normalidad. Todos los contrastes realizados se han medido con un nivel de significación del 5 %. El software utilizado ha sido IBM SPSS versión 24.

III. Resultados

III-A. Diseño de la escala de competencias de investigación en ciberseguridad

Las competencias que se incluyen en este estudio han sido seleccionadas a partir de una revisión sistemática cualitativa realizada de acuerdo a identificar cuáles son las competencias investigativas que pueden ser consideradas esenciales para la investigación Científica en Ciberseguridad, seleccionando aquellas que se repiten más frecuentemente en los documentos consultados.

La revisión ha comenzado con estudios y normativas sobre competencias básicas, generales o transverales en la educación superior ya que la asignatura está incluida en un programa oficial de Master Universitario ([4]–[6]). Así mismo, se han identificado las competencias que pueden ser consideradas como transversales en los estudios sobre formación investigativa y en investigación [7]–[11]. Se ha continuado con la revisión de trabajos centrados en las competencias profesionales (

[12]), así como con los documentos curriculares de la materia (guías docentes y plan de estudios).

Destacamos en este punto los siguientes estudios:

1. Descriptores de calidad de Dublin, *Joint Quality Initiative* (JQI, [13]). Este marco se desarrolló para establecer y cuantificar la calidad de la educación superior en el Plan Bolonia. Establece diferentes competencias cuya consecución supone la completación de cada uno de los tres ciclos establecidos (Grado, Master y Doctorado).
2. El Marco para el Desarrollo de las Habilidades del Investigador o *Research Skill Development (RSD*, [14]. En este documento podemos encontrar aquellas facetas que se consideran claves en la profesión de investigador. Consta de 6 aspectos concretos: curiosidad, determinación, crítica, organización, creatividad y persuasión. Además, consta de 7 niveles diferentes que se presuponen a conseguir desde la etapa de Educación Primaria hasta el ciclo de Doctorado.
3. El marco de calidad del desarrollo de competencias profesionales establecido por ENQA (European Association for Quality Assurance in Higher Eudaction), [15]. En este documento se detallan las competencias identificadas como esenciales en el sector profesional. Se dividen en tres categorías: conocimiento, competencias sistémicas o técnicas, y habilidades sociales. Así mismo, se desarrollan dos niveles de competencia: inicial y senior.

Finalmente, se ha realizado un análisis de las necesidades educativas relacionadas con la investigación que ha de tener un programa de formación en Ciberseguridad, y que se han establecido como indicadores de calidad y excelencia en *The NICE Cybersecurity Workforce Framework* [1]. El marco NICE proporciona una referencia para que los docentes del campo de la Ciberseguridad desarrollen programas de estudios, certificados o programas de grado, programas de capacitación, cursos, seminarios y ejercicios o desafíos que cubran los *KSAs (Knowledge, Skills and Abilities*) y las *Tasks (tareas*) requeridos en la actualidad para desarrollar labores profesionales enmarcadas en el campo de la Ciberseguridad. Este marco NICE incluye 31 áreas de especialidad repartidas en siete categorías. Cada área incluye diferentes roles de trabajo que tienen asignados grupos de *KSAs* y *Tasks.*

Aunque todas las categorías requieren del desarrollo de determinadas competencias de investigación, destacamos que en este marco la Investigación es una categoría en sí misma. La descripción de esta categoría incluye la investigación de eventos de ciberseguridad o delitos relacionados con sistemas, redes y evidencia digital de tecnologías de la información (TI) (p.11 [1]). A su vez, esta se divide en dos áreas de especialización: la *Ciber-Investigación* y el *Forense Digital*. Si nos centramos en el primer área, podemos encontrar su descripción en la tabla II.

Recopilando toda la revisión realizada, las competencias investigativas seleccionadas para desarrollar en la asignatura de *Investigación Científica* son las más frecuentes de los documentos consultados. Se muestran en la tabla III. Destacan por su carácter transversal e interdisciplinar con el objetivo de que la materia curricular aporte habilidades de investigación científica que puedan ser aplicadas en cualquier sub área de conocimiento del campo de la Ciberseguridad (análisis de riesgos, desarrollo de software, arquitectura de sistemas, desarrollo e investigación en tecnología, desarrollo de sistemas, análisis de ciber-defensa, gestión y evaluación de vulnerabilidades, etc).

Tabla II
ÁREA DE ESPECIALIZACIÓN: CIBER INVESTIGACIÓN. FUENTE: [1], P.121

Cyber- Investigation	
Work Role Name	Cyber Crime Investigator
Work Role ID	IN-INV-001
Speciality Area	Cyber Investigation (INV)
Category	Investigate (IN)
Work Role Description	Identifies, collects, examines, and preserves evidence using controlled and documented analytical and investigative techniques.
Tasks	T0031, T0059, T0096, T0103, T0104 T0110, T0112, T0113, T0114, T0120 T0193, T0225, T0241, T0343, T0346 T0360, T0386, T0423, T0430, T0433 T0453, T0471, T0479, T0523
Knowledge	K0001, K0002, K0003, K0004, K0005 K0006, K0046, K0070, K0107, K0110 K0114, K0118, K0123, K0125, K0128 K0144, K0155, K0156, K0168, K0209 K0231, K0244, K0251, K0351, K0624
Skills	S0047, S0068, S0072, S0086
Abilities	A0174, A0175

III-B. Diseño de la experiencia

Las competencias de investigación o investigativas engloban un proceso de formación dirigido a la integración de capacidades inherentes a la actividad científico-investigadora propias de una profesión. Una vez se han determinado las capacidades a desarrollar en el alumnado, es el momento de establecer el currículo de la materia y diseñar la aproximación metodológica que se va a utilizar.

Actualmente, existen dos aproximaciones metodológicas al desarrollo de competencias investigativas que difieren en el enfoque metodológico y los aspectos que se quieren priorizar. El término *formación para la investigación* parte de un enfoque desde el curriculo (p.190, [16]). Por otra parte, la aproximación desde la *investigación formativa* se basa en la utilización de la actividad investigadora como eje transversal del proceso de enseñanza y aprendizaje ([17]). Esta perspectiva se centra en factores relevantes en el éxito académico, especialmente en disciplinas STEM (Science, Techonology, Engineering and Mathematics), como la motivación hacia la investigación, el aprendizaje autónomo o la actitud positiva hacia la materia de estudio ([18]–[24]).En esta última categoría se encuentran metodologías como la *Investigación tutorizad*a, la *Enseñanza basada en la investigación* o la *Enseñanza orientada a la investigación* ([25], [26]). Esta última aproximación metodológica se basa en aportar al alumnado una experiencia de investigación global y práctica que suele llevarse a cabo mediante la participación de los estudiantes en proyectos de investigación reales que se lleven a cabo en las instituciones. En este caso, el currículo está enfocado a trabajar metodologías de investigación, así como a crear y validar nuevo conocimiento ([27], [28]). En todas las aproximaciones existentes se destaca el uso de actividades basadas en el aprendizaje basado en proyectos y problemas, que incluyan las tecnologías de la información y

Tabla III
COMPETENCIAS INVESTIGATIVAS SELECCIONADAS.

Competencia		Sub-Competencia: Que el estudiante sea capaz de ...	
C1	Resolución de problemas	C1.1.	Formular preguntas de investigación.
		C.1.2.	Identificar problemas de investigación en ámbientes multidisciplinares.
C2	Análisis de la información		
		C.2.1.	Buscar y recoger información con la metodología adecuada.
		C.2.2.	Gestionar la información/datos utilizando tecnologías de la información.
		C.2.3.	Evaluar, sintetizar e integrar la información consultada.
C3	Argumentación		
		C.3.1.	Formular juicios críticos y argumentados basados en información contrastada.
C4	Comunicación oral y escrita		
		C.4.1	Comunicar conclusiones y conocimiento a público especializado y no especializado de manera oral.
		C.4.2.	Comunicar conclusiones y conocimiento a público especializado y no especializado de manera escrita.
C5	Gestión del tiempo y organización		
		C.5.1.	Gestionar y coordinar el tiempo dedicado al trabajo.
		C.5.2.	Organizar las tareas a realizar.
C6	Habilidades de aprendizaje		
		C.6.1.	Estudiar de manera autónoma y auto dirigida.
		C.6.2.	Desarrollar capacidades de resiliencia ante el estrés, la presión y el fracaso.
		C.6.3.	Potenciar las actitudes proactivas ante el trabajo.

la comunicación, y prioricen el trabajo colaborativo

Aunque los trabajos citados muestran que el alumnado responde de manera positiva a las metodologías descritas, bien es cierto que la mayoría de éstos se han llevado a cabo a través de materias curriculares de *Metodologías de investigación* que suelen estar centradas en el aprendizaje del método científico para un campo concreto, habitualmente ciencias sociales y humanidades. Sin embargo, la aplicación de métodos de enseñanza que potencien una formación investigativa transdisciplinar, y no sólo centrada en el método científico, se alinea en mayor medida con la situación actual del campo de la Ciberseguridad en la que diferentes disciplinas científico-tecnológicas requieren diversas tipologías investigativas, los equipos de trabajo son multidisciplinares reuniendo especialistas de diversos campos, y la vanguardia de la generación del conocimiento ha completado el método científico hasta situarse en su cuarto paradigma mediante la ciencia intensiva de datos ([29], [30]). Por otra parte, la potenciación de la relación entre la enseñanza universitaria y el mundo laboral, o el desarrollo de metodologías docentes en entornos virtuales de enseñanza implican un reto complejo a la par que presente en la innovación universitaria actual ([3], [31]). En concreto, en lo que se refiere a la adquisición y desarrollo de competencias de investigación, los aspectos mencionados no han sido tenidos en cuenta en la mayoría de las metodologías docentes validadas existentes.

Por lo tanto, el enfoque pedagógico que se va a proponer para la materia de Investigación Científica en el campo de la Ciberseguridad ha de ser basarse en técnicas de enseñanza que proporcionen al alumnado la posibilidad de adquirir competencias investigativas de manera integradora, interdisciplinar y transversal, adecuándose a los métodos de trabajo propios de cada disciplina, pero aprendiendo aquellas habilidades y contenidos propios de la investigación como profesión. Así mismo, ha de ser una aproximación metodológica capaz de ser significativa en entornos virtuales y que potencie la colaboración entre el sector académico y profesional.

La metodología docente que se propone en este artículo se basa en *role playing*. Además, el diseño metodológico se ha planteado tomando como referentes pedagógicos el aprendizaje experimental, el aprendizaje significativo y el enfoque constructivo. A su vez, se ha potenciado la línea metodológica de "*aprender haciendo*".

El diseño curricular de la materia está centrado en el aprendizaje de contenidos especializados relacionados con la investigación como profesión. El alumnado ha de elegir un tópico del campo de la Ciberseguridad en el que desarrollará una investigación a lo largo de todo el semestre. Esta puede tener caracter de revisión o experimental. A su vez, se han de realizar diferentes tareas relacionadas en las que los avances de la investigación se van incluyendo. Además, se potencia que la elección esté relacionada con el tema de *Trabajo Fin de Master,* así como con la asignatura *Practicas en Empresas*. Se combinan clases magistrales de los tópicos curriculares seleccionados con sesiones de trabajo de aula. En el planteamiento de esta metodología docente resulta fundamental que el alumnado desarrolle competencias de aprendizaje autónomo y auto-regulación, debido a que estas habilidades son esenciales en el campo de la investigación. De esta manera, se plantea un 50 % de las horas lectivas para poder trabajar en el aula, consultar dudas, desarrollar la investigación, y solicitar correcciones y sugerencias al profesorado implicado.

La impartición de la materia suele contar con la colaboración de grupos de investigación que participan en la docencia del master, y que trabajen en los temas de investigación que haya elegido el alumnado. De esta manera, pueden aportar temáticas de investigación, así como referencias bibliográficas relevantes y diversidad de metodologías de investigación. Así mismo, estos grupos de investigadores, como profesionales del ámbito empresarial, colaboran en la experiencia mediante la revisión y participación activa en alguna de las tareas. Cabe destacar que la valoración de los investigadores involucrados en la asignatura se incluye de manera positiva en la calificación del alumnado.

Como hemos mencionado, una vez que el alumnado ha elegido el tema de investigación, ha de profundizar y realizar una pequeña investigación sobre éste a la vez que elabora

determinadas tareas. A su vez, se desarrolla la asignatura de Investigación Científica mediante una metodología docente mixta que combina clases magistrales con las sesiones libres de trabajo en el aula. El programa de la asignatura es el siguiente:

1. Tema 1: Proyectos de investigación: en este tema se introducen los proyectos de investigación que los investigadores establecen con entidades públicas o privadas. Se resuelven las siguientes cuestiones
   a) ¿Cómo escribir una memoria de proyecto de investigación?
   b) ¿Qué puntos debe incluir la memoria del proyecto de investigación?
   c) Preparación de Curriculum Vitae para diferentes convocatorias.
2. Tema 2: Revistas científicas. En este tema se describen las revistas científicas y cuál es su papel dentro de la investigación científica:
   a) ¿Cómo elegir una revista científica para enviar un artículo de investigación?
   b) ¿Qué es el índice de impacto?
   c) ¿Dónde puedo encontrar una lista con las revistas científicas más relevantes?
3. Tema 3: Artículos científicos. En este tema se desarrolla la información relativa a los artículos científicos.
   a) ¿Cómo escribir un artículo científico?
   b) ¿Qué puntos debe incluir un artículo científico?
   c) ¿Cómo funciona el proceso de publicación de un artículo científico?
4. Tema 4: Conferencias y congresos. En este tema se introducen los congresos y las conferencias.
   a) ¿Cómo elegir un congreso o conferencia?
   b) ¿Qué tipo de presentación se puede enviar a un congreso o conferencia?
   c) ¿Cómo funciona un congreso y cómo organizarlo?
5. Tema 5: Propiedad intelectual y modelos de utilidad. En este bloque se desarrolla la legislación y praxis en el campo de la propiedad intelectual, ética investigadora, patentes, licencias, modelos de utilidad y transferencia de la investigación a las empresas (innovación).

El alumnado dispone de todos los temas en la plataforma Moodle desde el primer día de curso. Además, también se incluyen abundantes referencias bibliográficas relacionadas con metodologías y tópicos de investigación del campo de la Ciberseguridad, documentos ejemplo, el programa de contenidos y evaluación de la asignatura, las fechas de entrega, una tabla con la temporalización de la materia, y las escalas de evaluación de todas las tareas. Asi mismo, se suben a la plataforma Youtube vídeos formativos de cada tema dado en el aula creados *ad hoc*. Por último, y con la finalidad de que la experiencia *online* sea lo más similar a la experiencia presencial, se ha desarrollado la asignatura mediante la plataforma AVIP, impartiendo la docencia en streaming.

Las actividades de aprendizaje constituyen a su vez las tareas que se van a utilizar para la evaluación. Se describen en la tabla IV

La evaluación de las actividades se realiza en base a las escalas de valoración de cada tarea, disponibles para el alumnado en la plataforma virtual. La actividad A1 cuenta con dos entregas debido a que se aportan correcciones de la primera versión a los estudiantes para que puedan realizar una segunda versión mejorada de la memoria del proyecto. La misma situación se tiene con la actividad A4 (versión 1), que se somete a una heteroevaluación que incluye evaluación por pares y co-evaluación por parte de un compañero. Cada trabajo es revisado por un investigador afín al área, un profesor de la asignatura y un estudiante de la materia con un tópico de investigación enmarcado en el mismo contexto científico. De esta manera, se potencia el desarrollo y adquisición de ciertas habilidades de investigación desde una perspectiva de análisis y evaluación, cambiando así el habitual rol del estudiante y asegurando una mejora en la calidad del aprendizaje. Además, se ha de enviar una nueva versión revisada A4 (versión 2) en base a las sugerencias y correcciones dadas por los evaluadores y los compañeros en la que se valora, principalmente, la realización de los cambios propuestos. Con respecto a A7, se ha de enviar un resumen y realizar una charla en el congreso *Cybersecurity Scientific Research Conference*. Este congreso simulado se celebra anualmente de manera virtual y presencial y sirve para que el alumnado exponga durante 15-20 minutos los resultados obtenidos en su investigación. El público está especializado en el campo de la ciberseguridad y proviene tanto del ámbito académico como del mundo empresarial. Se dispone de una rúbrica de evaluación, realizada mediante juicio de expertos, para que los asistentes puedan valorar a los ponentes. Para las actividades A4 y A7, el profesorado de la materia tiene en cuenta, de manera positiva, las evaluaciones que los investigadores y docentes que colaboran en la experiencia han emitido. Además, los estudiantes pueden guardar aquellas tareas en las que hayan llegado a la calificación mínima para la segunda convocatoria.

Tabla IV
ACTIVIDADES DE APRENDIZAJE Y PESO EN LA CALIFICACIÓN.

Actividad	Temporalización	Peso
A1 (versión 1)	Memoria de proyecto de investigación	10 %
A1 (versión 2)	Memoria de proyecto de investigación corregida	10 %
A2	Preparación de Curriculum Vitae	5 %
A3	Elección de una revista científica	10 %
A4 (versión 1)	Preparación y envío de un artículo científico.	20 %
A4 (versión 2)	Versión revisada de artículo científico	5 %
A5	Revisión de artículo de un compañero	5 %
A6	Envío de abstract a congreso simulado	5 %
A7	Charla en congreso simulado	15 %
A8	Trabajo sobre patentes y licencias	15 %

La temporalización llevada a cabo en el curso escolar 2018/2019 se desarrolla en la tabla V.

La entrega de los trabajos escritos (A4, A5 y A6), y de las revisiones, se realiza a través de la plataforma EasyChair (veáse Fig. 1), de la misma forma que se utiliza en gran cantidad de congresos y conferencias reales del campo de la ciberseguridad. Así mismo, también se utiliza la plataforma Moodle y los Formularios de Google. En el caso de las exposiciones virtuales, se ha utilizado el software Skype.

III-C. Evaluación de la experiencia

III-C1. Rendimiento: En la Fig. 2 podemos encontrar la calificación media obtenida por el alumnado en cada actividad

Tabla V
TEMPORALIZACIÓN

Actividad	Temporalización /Deadline
Explicación de la materia, de la evaluación y fechas	8 de octubre
Tema 1	9 y 15 de octubre
Tema 2	22 de octubre
Tema 3	5 de noviembre
Tema 4	19 y 27 de noviembre
Tema 5	12 y 13 de noviembre
A1 (versión 1)	29 de noviembre
Entrega de correcciones	10 de diciembre
A1 (versión 2)	19 de diciembre
A2	19 de diciembre
A3	21 de noviembre
A4 (v1)	8 de enero
A4 (v2)	29 de enero
A5	21 de enero
A6	21 de diciembre
A7	21- 29 de enero
A8	10 de diciembre

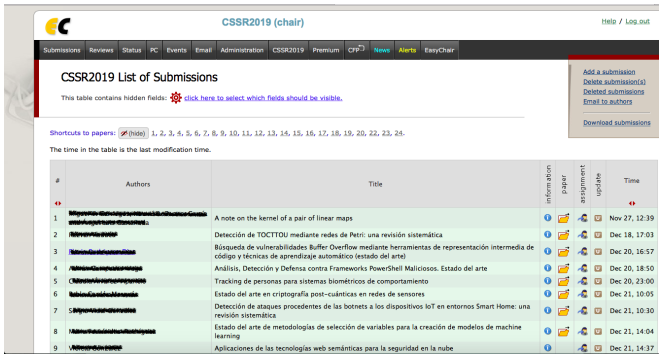


Figura 1. Imagen de Easychair

de aprendizaje.

En la Fig. 3 podemos encontrar la calificación media obtenida por el alumnado en la materia a lo largo de las tres intervenciones de los últimos cursos.

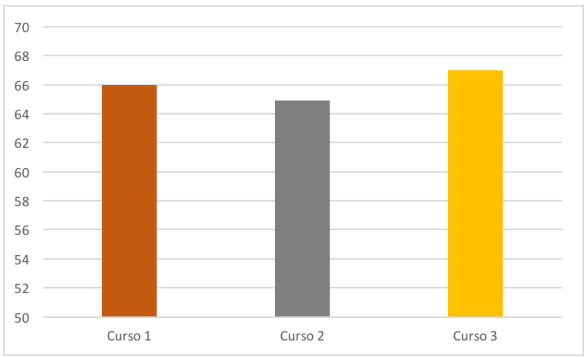


Figura 3. Calificación media obtenida en cada curso escolar.

Durante los tres cursos, ha habido un 37.5 % de alumnado que ha aprobado la materia en la segunda convocatoria, y un 5 % que no ha conseguido superar la asignatura. El 62 % restante ha superado la asignatura en su primera convocatoria. Cabe destacar que el alumnado de la modalidad *online* suele repartir el trabajo a entregar entre las dos convocatorias de la asignatura ya que la gran mayoría de estos estudiantes se encuentran cursando la modalidad *online* debido a que están trabajando al mismo tiempo.

Con respecto a la percepción que presenta el alumnado sobre si ha adquirido y en qué medida las competencias de investigación seleccionadas, el estudio realizado [32] arroja una percepción de auto-adquisición media de 3.76 sobre 5 (una vez se ha medido la fiabilidad y validez del instrumento de evaluación), destacando una alta percepción (mayor de 4 sobre 5) de auto-adquisición de la habilidad en la búsqueda de información sobre temáticas relacionadas, y la capacidad de emitir conclusiones y tener una opinión argumentada sobre textos científicos.

III-C2. Innovación: Dentro de los indicadores de la calidad de la metodología de innovación propuesta, lo primero que se analiza es la coordinación entre los objetivos de aprendizaje y las actividades realizadas. En la tabla VI puede encontrarse la relación existente entre los objetivos propuestos y las diferentes acciones llevadas a cabo en la experiencia.

Tabla VI
ALINEACIÓN DESARROLLO DE COMPETENCIAS INVESTIGATIVAS-ACTIVIDADES DE APRENDIZAJE

Competencia		Actividad/Tarea
C1	C1.1.	Elección de tópico
	C.1.2.	Elección de tópico
C2	C.2.1.	A3 y Realización de la investigación
	C.2.2.	A6 y Realización de la investigación
	C.2.3.	A2, A3, A4, A7
C3	C.3.1.	A3, A4, A5, A7
C4	C.4.1	A7
	C.4.2.	A4, A5 y A6
C5	C.5.1.	A1 y A7
	C.5.2.	A1
C6	C.6.1.	Todas
	C.6.2.	Todas
	C.6.3.	Todas

Con respecto al uso de las tecnologías de la información y la comunicación, todos los recursos y materiales se han incluido en Moodle. Además, se ha utilizado la plataforma EasyChair para la tarea A4 (v1 y v2), y los Formularios de Google oara cumplimentar las evaluaciones. Por otra parte, se han utilizado tecnologías participativas para la elaboración de vídeos formativos subidos a Youtube, la celebración de congreso virtual a través de Skype y AVIP y las tutorías a través de AVIP mediante videoconferencias.

En lo que se refiere a la coordinación entre la asignatura *Investigación Científica* y otras de la misma titulación, el 100 % de los Trabajos Fin de Master y más del 90 % de las memorias presentadas en la asignatura *Prácticas en Empresa* toman como base los trabajos de la asignatura *Investigación Científica*, bien como estado del arte o como investigación experimental exploratoria del tema que desarrollan.

Así mismo, se ha obtenido un alto grado de participación y colaboración entre diferentes grupos de investigación y el personal del ámbito empresarial. En relación al profesorado y grupos de investigación implicados, pueden consultarse en la tabla VII

III-C3. Aplicación en entornos virtuales: Para analizar si ha habido diferencias significativas entre la calificación obtenida en las actividades de aprendizaje entre el alumnado de la modalidad presencial y la modalidad *online* se ha realizado un contraste inferencial. Primero se ha comprobado si la variable a analizar seguía una distibrución normal para poder

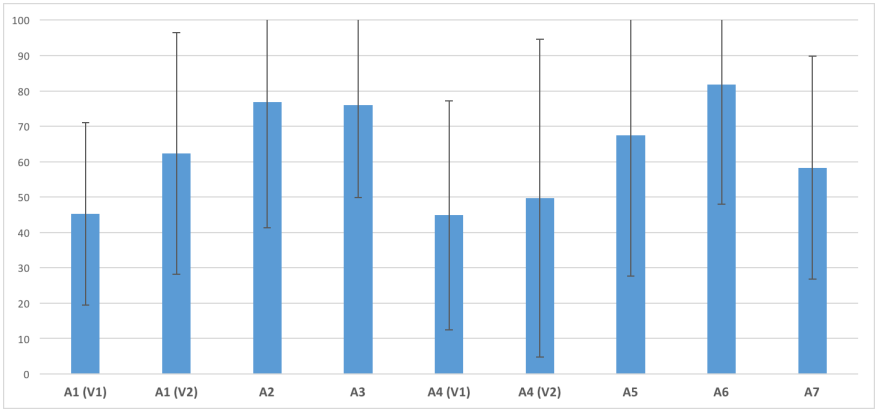


Figura 2. Media y desviación en cada actividad de aprendizaje.

Tabla VII
GRUPOS PARTICIPANTES EN LA EXPERIENCIA

Participantes	Número participantes	Actividades en las que participa
Profesorado de la materia	3	Todas
Grupo de Innovación Docente NARVIC	1-3	A4 y A7
Grupo de Robótica	1-5	A4 y A7
Grupo SECOMUCI	1-3	A4 y A7
Personal de RIASC	1-3	A4 y A7
Personal otras universidades	1	A4
Personal de INCIBE	2-7	A4 y A7

determinar la idoneidad de un contraste paramétrico o no. A continuación, se ha realizado el contraste de comparación de 2 muestras independientes (t de Student o U de Mann Whitney). Los resultados se incluyen en la tabla VIII.

Tabla VIII
ANÁLISIS INFERENCIAL

	K-S	Sig. Normalidad	t	Sig. Modalidad	Z	Sig. Modalidad
A1 (V1)	0.094	.20	2.444	**.019**		
A1 (V2)	0.208	.001(*)			-3.195	**.001**
A2	0.257	.001(*)			-1.2	.230
A3	0.181	.002			-0.664	.507
A4 (V1)	0.148	.032			-1.914	.056
A4 (V2)	0.225	.001(*)			-1.42	.154
A5	0.277	.001 (*)			-1.21	.225
A6	0.397	.001(*)			-1.106	0.269
A7	0.248	.001(*)			-2.109	**.035**

Las actividades en las que la modalidad aparece como un factor que influye en la calificación son A1 (tanto v1 como v2) y A7. En ambos grupos la versión revisada de A1 obtiene una mejor calificación. Sin embargo, y como podemos observar en la Fig. 4, la modalidad presencial ha obtenido mejores resultados tanto en A1 como en A7. Cabe destacar que se ha observado que el alumnado de la modalidad *online*, mayoritariamente trabajando a tiempo completo a la vez que realiza el máster, suele comenzar a realizar y organizar sus tareas más tarde en el cuatrimestre, lo que implica menos tiempo de trabajo y dedicación.

III-C4. Otros aspectos: Destacamos en este apartado aquellos aspectos cualitativos extraídos de las evaluaciones del alumnado y de los tres ciclos de investigación-acción que han de ser modificados para la mejora de la calidad educativa:

1. Mejoras de los ciclos de investigación-acción: en la última implementación se ha incluido más material bibliográfico, así como todas las escalas de evaluación de todas las actividades de aprendizaje.
2. El creciente número de estudiantes existentes en la modalidad *online* ha supuesto la introducción de nuevas plataformas para poder intercambiar experiencias y que el alumnado tuviera disponible toda la información. Este es el caso de la plataforma *Skype,* los vídeos formativos en *Youtube*, y los Formularios de Google.
3. Por otra parte, los materiales parecen adecuados, destacando que la mayoría del alumnado prefiere vídeos formativos de los contenidos ante la opción de la grabación de las clases en directo.
4. El ajuste de la temporalización será revisada en relación al distanciamiento entre algunas tareas.
5. En general, el alumnado ha presentado dificultades ante la labor investigadora y la realización del estado del arte de un tópico de investigación. Como observación cualitativa, podemos destacar que la mayoría no se ha sentido cómodo gestionando su aprendizaje, prefieren menos autonomía, y desconocen las implicaciones de la labor investigadora, así como sus exigencias en términos de plazos, normativas y calidad del trabajo.
6. Los errores más comunes se relacionan con el desconocimiento de las normativas sobre referencias bibliográficas y formatos, la planificación de la investigación, y la mala organización del tiempo a dedicar a cada tarea.

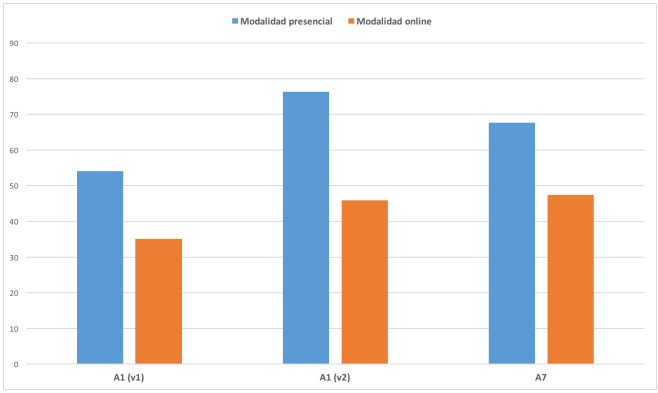


Figura 4. Calificación media en cada modalidad.

IV. Conclusiones

La educación basada en competencias supone un reto complejo para el colectivo docente debido a que las situaciones de aprendizaje han de desarrollar conocimiento especializado de una materia y la capacidad de aplicarlo en entornos prácticos. Además, la actual sociedad del conocimiento requiere que la formación universitaria aporte a sus estudiantes competencias de investigación como la habilidad de aprender de forma autónoma, dinámica y continua, y la capacidad de generar proyectos relevantes que aporten soluciones a los desafíos a los que la ciencia y la tecnología se enfrentan diariamente. En este escenario educativo se hace necesario establecer relaciones entre los métodos de enseñanza y la adquisición de competencias que tiene el alumnado que permitan diseñar situaciones de aprendizaje coherentes y significativas.

El presente trabajo muestra un marco metodológico innovador, *la Enseñanza Orientada a la Simulación de la Investigación*, que ha sido aplicado en la materia de Investigación científica en un programa de formación en Ciberseguridad. Este campo es especialmente heterogéneo en términos de metodologías de investigación utilizadas y, por ese motivo, las propuestas más tradicionales no resultan eficaces.

Como futuro trabajo resulta esencial poder identificar qué competencias de investigación resultan clave en cada sector de la Ciberseguridad, así como las posibles diferencias entre el mundo académico y profesional.

Agradecimientos

Los autores agradecen su participación a todos los investigadores y profesionales que han colaborado en esta experiencia a lo largo de tres cursos escolares.

Referencias

[1] W. Newhouse, S. Keith, B. Scribner and G. Witte, "National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework", NIST Special Publication 800-181. U.S. Department of Commerce, 2017 [Online] Disponible en https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-181.pdf.

[2] Estándares de calidad en los programas de formación en Ciberseguridad de la National Cyber Security Centre (NCSC) del Government Communications Headquarters (GCHQ) de Reino Unido dentro del UK Cyber Security Strategy 2016-2021.

[3] M.J. León Guerrero and M.C. López López, "Criterios para la Evaluación de los Proyectos de Innovación Docente Universitarios". *Estudios sobre Educación*,vol. 26, pp. 79-101, 2014.

[4] M. Gómez-Ruiz, G. Rodríguez-Gómez and M.S. Ibarra-Saiz, "COMPES: Autoinforme sobre las competencias básicas relacionadas con al evaluación de los estudiantes universitarios". *Estudios sobre educación*, vol.24, pp. 197-224, 2013.

[5] Real Decreto 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales. (BOE de 30 de octubre).

[6] Real Decreto 861/2010, de 2 de julio, por el que se modifica el Real Decreto 1393/2007, de 29 de octubre, por el que se establece la ordenación de las enseñanzas universitarias oficiales. (BOE, de 3 de Julio).

[7] J. Balbo, (2008), *Formación en competencias investigativas, un nuevo reto en las universidades* . Caracas: Universidad Central de Venezuela.

[8] O.Estrada, "Sistematización teórica sobre la competencia investigativa", *Revista Electrónica Educare*, vol. 18, n.2 , pp. 177-194, 2014. doi: http://dx.doi.org/10.15359/ree.18-2.9

[9] E.Ortega and A. Jaik, " Escala de evaluación de competencias investigativas", *Revista Electrónica Praxis Investigativa ReDIE*, vol.2, n. 3, pp. 72-80, 2010.

[10] D.Lopatto, "Survey of undergraduate Research Experiences (SURE): First Findings", *Cell Biology Education*, vol.3, pp.270-277, 2004.

[11] P. Sánchez and R. Tejeda, "El proceso de formación investigativa del profesional ingeniero y la(s) competencia(s) investigativa(s)", *Pedagogía Universitaria*, vol.15, n.4, pp. 37-47, 2010.

[12] J. Tejada Fernández and C. Ruiz Bueno, "Evaluación de competencias profesionales. Retos e implicaciones", *Educación XX1*, vol. 19, n.1, pp. 17-38, 2016. doi:http://10.5944/educXX1.12175

[13] Joint quality initiative: Shared 'Dublin' descriptors for the Bachelor's, Master's and Doctoral awards, 2004. [Online] Disponible en http://ecahe.eu/assets/uploads/2016/01/Joint-Quality-Initiative-the-origin-of-the-Dublin-descriptors-short-history.pdf

[14] J. Willison and K. O'Regan, "Research Skill Development, a conceptual framework for Primary school to PhD", 2006. The University of Adelaide.[Online]. Disponible en https://www.adelaide.edu.au/rsd/

[15] *ENQA Quality Assurance Professional Competencies Framework*, European Association for Quality Assurance in Higher Education, [Online]. Disponible en https://enqa.eu/indirme/papers-and-reports/occasional-papers/ENQA\%20Competencies\%20Framework.pdf

[16] M.E.Guerrero, "Formación de habilidades para la investigación desde el pregrado", *Acta Colombiana de Psicología*, vol.10, n.2, pp.190-192, 2007.

[17] J. M. Miyahira, "La investigación formativa y la formación para la investigación en el pregrado", *Revista Medica Herediana*, vol.20, n.3, pp. 119-122, 2009.

[18] M . Henri, M. D. Johnson and B. Nepal , "A Review of Competency-Based Learning: Tools, Assessments, and Recommendations" , *Journal of Engineering Education*, vol. 106, n. 4, pp. 607–638, 2017.

[19] B. Aeschlimann, W. Herzog and E. Makarova, "How to foster students' motivation in mathematics and science classes and promote students' STEM career choice. A study in Swiss high schools", *International Journal of Educational Research*, vol.79, pp.31–41, 2016.

[20] A. Anwar, "The Use of Students' Feedback for Effective Learning in Engineering Units", *The Int. J. Eng. Educ.*, vol.18, n.4, pp. 131-142, 2012.

[21] T. Brumm, L. F. Hanneman and S. K. Mickelson, "Assessing and developing program outcomes through workplace competencies", *International Journal of Engineering Education*, vol. 22, n.1, pp.123–129, 2006.

[22] D. Gerónimo, J. Serrat, A. M. López and R. Baldrich, "Traffic sign recognition for computer vision project-based learning", *IEEE Trans. Educ.*, vol. 56, no. 3, pp. 364-371, Aug. 2013.

[23] M.J. Rubio Hurtado, R. Vilá Baños and V. Berlanga Silvente, "La investigación formativa como metodología de aprendizaje en la mejora de competencias transversales", *Procedia - Social and Behavioral Sciences*, vol. 196, pp. 177- 182, 2015.

[24] E. J. H. Spelt, P.A. Luning, M. A. J. S. van Boekel and M. Mulder, "Constructively aligned teaching and learning in higher education in engineering: What do students perceive as contributing to the learning of interdisciplinary thinking?", *Eur J Eng Educ*, vol.40, n. 5, pp. 459-475, 2015.

[25] G. J. Visser-Wijnveen, R.M. van der Rijst and J.H. van Driel, "A questionnaire to capture students' perceptions of research integration in their courses", *High. Educ.*, vol.71, pp.473-488 ,2016. doi:10.1007/s10734–015–9918-2

[26] L. Smyth, F. Davila, T. Sloan, E. Rykers, S. Backwell and S.B. Jones, "How science really works: The student experience of research-led education", *High. Educ.*, 2015. doi:10.1007/s10734-015-9945-z

[27] F.Bottcher and F. Thiel, "Evaluating research-oriented teaching: a new instrument to assess university students' research competences", *High. Educ.*, vol. 75,n.1, pp 91–110, 2018, doi: https://doi.org/10.1007/s10734-017-0128-y

[28] F. Hauser, R. Reuter, H. Gruber and J. Mottok, "Research competence: Modification of a questionnaire to measure research competence at universities of applied sciences", *IEEE Global Engineering Education Conference (EDUCON)*, Tenerife, 2018, pp. 109-117. doi: 10.1109/EDUCON.2018.8363216 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8363216&isnumber=8363090

[29] C. Dede, *Data-intensive research in education: current work and next steps.* Computer Research Association, 2015.

[30] T. Hey, S. Tansley and K. Tolle, *The Fourth Paradigm. Data Intensive Scientific Discovery*. Microsoft Research, 2009.

[31] T. Mauri, C. Coll, and J. Onrubia, " La evaluación de la calidad de los procesos de innovación docente universitaria. Una perspectiva constructivista", *Revista de Docencia Universitaria*, 1, 2007

[32] N. DeCastro-García, A.L. Muñoz Castañeda and Miguel V. Carriegos, Relational model between teaching strategy and cross-disciplinary research competencies in higher engineering education. A machine learning approach", Enviado a *IEEE Transactions on Learning Technologies*, Mayo 2019.

Diseño de actividad lúdica orientada a la enseñanza de métodos y técnicas de OSINT

Miguel Paramo, Víctor A. Villagrá
Universidad Politécnica de Madrid (UPM), Avda. Complutense 30, 28040, Madrid
mparamo@alumnos.upm.es, victor.villagra@upm.es

En este documento se expone el diseño de una actividad docente aplicada a la formación en ciberseguridad que incorpora de manera innovadora componentes narrativos (*storytelling*) y aspectos lúdicos (*gamification*) en una modalidad de ejercicio de "captura la bandera" orientado al aprendizaje de técnicas y métodos transversales propios del campo de Inteligencia de Fuentes Abiertas (OSINT). La actividad propuso un caso de estudio criminalístico ficticio con el objetivo de que, de manera exploratoria y aplicada, el alumno entrene y adquiera dichas habilidades transversales bajo el marco de la asignatura "Gestión de Riesgos y Operaciones de Ciberseguridad" del Máster en Ciberseguridad de la Universidad Politécnica de Madrid. La actividad se desarrolló en el curso académico 2018-2019 con motivo del estudio de operaciones preventivas en el campo de la ciberseguridad; en concreto la ciber-inteligencia y la vigilancia digital.

***Index Terms*- OSINT, actividad académica, CTF**

Tipo de contribución: *Formación e innovación educativa*

I. Introducción

En este documento se expone el procedimiento de creación de una actividad docente en el campo de la ciberseguridad que surge de la iniciativa de explorar vertientes innovadoras en el planteamiento de ejercicios académicos que supongan una experiencia divertida tomando la inspiración de "*gymkanas*", o los populares "*escape rooms*" expresados en el formato de los tradicionales retos de ciberseguridad en la modalidad "*Capture the flag"* (CTFs) como los propuestos por plataformas como Atenea [1] o HackTheBox [2] que permiten practicar técnicas en un entorno seguro y controlado con objetivos concretos, motivos por el cual gozan de cierta popularidad y son positivamente valorados por profesionales del sector [3] y una vía de formación eficaz [4].

En materia de ciberseguridad, los conocimientos teóricos y capacidades prácticas se complementan sustentando una base de capacidades y habilidades específicas para el buen desempeño de la profesión en cualquiera de sus numerosas especializaciones [5]. Este conocimiento específico debe ser complementado con habilidades transversales que pueden considerarse complementarias; en concreto el área de la Inteligencia de Fuentes Abiertas, interpretada como la obtención de Inteligencia a partir del conocimiento público disponible para la resolución de problemas, toma de decisiones o elaboración de productos en el ámbito de la Ciberseguridad, es una disciplina aplicable a prácticamente cualquier vertiente profesional del campo de la seguridad informática; desde la Inteligencia de Amenazas más propia de un analista de malware o integrante de un "*Blue Team"* (Mitre, NVD, Virustotal…) hasta las capacidades de reconocimiento del objetivo de un integrante de un "*Red Team"* en una campaña de "*pentesting"* (Shodan, urlscan.io, dorking, exploración de DNSs, etc.).

Cuando se plantea el reto de enseñar y evaluar conocimientos acerca de técnicas basadas en OSINT, es necesario identificar que, por tratarse de una disciplina aplicada, tiene una carga teórica limitada y diferentes aproximaciones específicas bajo diversas materias de ciberseguridad especializadas. Se percibe como una disciplina candidata a ser trabajada de una manera práctica; pero el mero hecho de mostrar o emplear herramientas o hacer uso de fuentes de información de manera inconexa o sin un objetivo concreto podría percibirse como tedioso o poco estimulante por parte de los alumnos. Por ello el enfoque que se refleja en este documento parte de los probados beneficios en el aprendizaje de la Ciberseguridad por medio de CTFs [3][4] en conjunción con técnicas narrativas y de *gamificación* que confieran a la actividad un contexto, objetivo y percepción de progreso.

Una de las claves de la faceta innovadora del ejercicio es el formato y planteamiento de la práctica y su estructura, contexto y contenido de la historia en unos términos serios pero desenfadados que integran de manera implícita ciertos componentes lúdicos aplicando cierta psicología del entretenimiento de manera que se provoca la curiosidad a la vez que se estimula la exploración de información, concatenación de ideas y deseo de completitud de la historia [6].

En el ejercicio de la actividad docente existe el reto de inculcar conocimiento a los alumnos de una manera eficaz en la que se debe maximizar el aprendizaje del alumno estimulando, entre otras, el desarrollo de sus capacidades resolutivas, críticas o analíticas [7] y mediante una carga de trabajo equitativa. Por ello se ha pretendido fomentar citadas habilidades transversales más allá de las aproximaciones teóricas que se habrán trabajado a lo largo del plan de estudios. Dado que, como se ha mencionado previamente, algunas técnicas, herramientas o recursos abiertos relacionados con la disciplina OSINT para otras disciplinas son profundizadas en otras materias, se pretende no repetir en el ejercicio el manejo de citadas herramientas y fuentes de información. Con todas las consideraciones previas, se plantea una actividad con un enfoque que complemente al conocimiento previo e introduzca otro tipo de actividades sobre recursos y fuentes de información como SOCMINT (*Social Media Intelligence*), FININT (*Finantial Intelligence*) y GEOINT (*Geographical Intelligence*).

La actividad se centrará en la transformación de datos,

búsquedas pivote entre plataformas, la habilidad de reconocer y separar la información útil de la que no lo es y la aplicación de conocimientos de otras materias como la informática forense o la criptografía de manera lógica.

Indirectamente se hará reflexionar al alumno acerca de la huella digital y el alcance e implicaciones que puede suponer la exposición de información pública en la red y el grado de conocimiento que pueden alcanzar tan solo mediante el análisis de fuentes abiertas entidades adversarias contando no necesariamente con altísimas capacidades, pero si tomando ventaja de la exposición moderada de información que a priori pudiera no parecer del todo sensible.

II. MATERIALES Y MÉTODOS

En la asignatura de Gestión de Riesgos y Operaciones de Ciberseguridad del Máster en Ciberseguridad de la Universidad Politécnica de Madrid se imparten estrategias y operaciones proactivas y reactivas tanto a nivel teórico como práctico y se abordan conceptos como el de ciber-inteligencia o el de vigilancia digital a partir de los cuales se aborda también la disciplina OSINT.

Mediante la resolución del ejercicio, el alumno se familiarizará con técnicas de adquisición de información por medio del manejo de herramientas de búsqueda de diferentes tipos de datos y fuentes diversas que pondrán a prueba su capacidad resolutiva, pensamiento lateral e incluso perspicacia siempre de una manera lógica y dentro de la dificultad intrínseca al ejercicio que además se deberá de ingeniar de cara a ofrecer un reto didáctico, satisfactorio y asumible a todos los alumnos independientemente de la heterogeneidad de destrezas y perfiles.

A. Planteamiento del ejercicio

El ejercicio se diseñó de manera que no se limite a la mera aplicación de técnicas o análisis de información inconexos que hagan uso de diversas herramientas o fuentes; sino de manera que exista un hilo conductor narrativo que lleve a la consecución de un objetivo, en este caso, la detención de una organización criminal ficticia. Para ello se requirió la elaboración de una breve historia que dividiera al ejercicio en diferentes apartados concatenados y sirviera a su vez de enunciado y guía de la práctica. Se presenta al alumno el enunciado del ejercicio que incluye las actividades a desempeñar y una hoja de respuestas con 10 preguntas cuya respuesta responde al modelo de CTF (captura la bandera) de respuesta concisa y que permite la evaluación del ejercicio en la escala 0-10 de manera directa. El ejercicio se diseñó de manera que no es posible avanzar al siguiente punto sin haber resuelto el anterior de manera que se fuerce al razonamiento lógico y concatenación de técnicas y búsqueda en las fuentes de información preparadas para cada caso.

El ejercicio comienza con un breve reto forense a partir del cual se obtiene una pista inicial que permite desarrollar el resto del ejercicio, el cual se puede resolver a elección del alumno con un navegador o complementando con herramientas como *Recon-ng* o *Maltego* aunque no resulta necesario. De igual manera, la elección del sistema operativo sobre el que trabajar no afecta a la resolución de la práctica. En algunos apartados, e indicados en el enunciado se recomienda a su vez el uso del teléfono móvil del alumno por simplicidad o comodidad.

El objetivo del alumno es obtener toda la información posible por medio de, exclusivamente, técnicas OSINT y prosperar en la investigación por medio del conocimiento acumulado y la concatenación de pistas hasta desvelar la identidad de uno de los integrantes de la organización criminal ficticia, así como detalles personales de esta persona ficticia. Finalmente, en la conclusión de la práctica, habrá logrado determinar dónde y cuándo se va a producir la próxima entrega para proceder a la detención de los criminales. A medida que se avanza en el desarrollo de la práctica a través de cada uno de los diez puntos propuestos se requiere de la aplicación de una técnica diferente y se concluye con una reflexión o lección a aprender.

Se pretende que el alumno disfrute de su elaboración y le resulte divertida e incluso le enganche. Para reforzar la sensación de progreso y recompensa, se incorporan determinados componentes humorísticos a lo largo de la práctica que incluyen guiños sutiles a la promoción de alumnos.

La práctica, enmarcada en una asignatura de 6 créditos, se diseñó para que pueda ser resuelta de manera individual en un intervalo de entre 3 y 8 horas dependiendo de las habilidades de cada alumno.

B. Materiales

Para la elaboración del ejercicio se prepararon las fuentes de información requeridas y se hizo uso de información pública ya existente (como el registro contable público de Bitcoin o localizaciones reales). Se crearon una serie de perfiles falsos en redes sociales (Facebook, Twitter, Instagram y Linkedin) así como dos perfiles en plataformas de mensajería instantánea (Whatsapp y Telegram) vinculados a un teléfono móvil real de una línea telefónica no utilizada y, para finalizar, se creó un anuncio también falso en Vibbo (antes Segundamano) y se registró una clave PGP en varios repositorios públicos (como el de Rediris) asociada a un nombre de usuario. Se planificó la concatenación de pistas siguiendo la historia de forma lógica y se enriqueció el contenido de cada una de las fuentes de información. En paralelo se redactó el documento guía de la actividad, así como la hoja de respuestas y el material de la práctica que incluía el volcado forense de una memoria microSD y un *leak* de datos modificado de la plataforma Taobao del año 2012 que se emplea en un determinado apartado de la práctica.

La preparación de material incluyó la creación de una cuenta de correo para poder registrar todas las anteriores cuentas en las respectivas plataformas, verificación por medio del móvil real de alguna de ellas (caso de Twitter) y la incorporación de imágenes propias o con licencia permisiva en los perfiles falsos.

Por último, se revisó repetidas veces el hilo conductor de la práctica y su resolución, comprobando la persistencia de los perfiles en las redes sociales (pudieran requerir de verificación o ser cancelados) y demás fuentes de información. Con todo el material listo se readaptó el guion narrativo del enunciado aportando más o menos pistas en cada apartado a resolver de manera que el esfuerzo requerido en tiempo y la dificultad de cada apartado quedará dentro de los márgenes deseables.

III. DESAROLLO DE LA ACTIVIDAD

El ejercicio se expresa por medio de una breve historia criminalística en la que el alumno adopta el rol de un profesional criminólogo de delitos telemáticos al que se le facilita una imagen en formato *EnCase* de una tarjeta microSD vinculada a una investigación en curso que persigue a organización criminal dedicada al tráfico de osos panda.

Fase 1

Se requiere recuperar varios documentos de la imagen suministrada (fichero evidence.E01) (que además han sido borrados), uno de ellos es un fichero .7z protegido por contraseña (que habrá que adivinar más adelante) y otro un fichero .docx alterado que denominamos "pista". Habrá que hacer uso de herramientas como FTK Imager o Autopsy para extraer ambos ficheros.

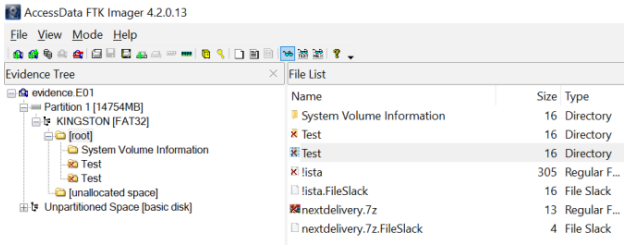


Fig. 1. Contenido de la tarjeta de memoria recuperado por FTK Imager.

Por medio de la herramienta *file* o un editor hexadecimal (reconociendo el número mágico) se concluye que el fichero *pista* se trata de un archivo .docx. Se procede a añadir la extensión al mismo y abrirlo. Dentro se encuentra un texto en coreano que habrá de traducir con, por ejemplo, Google Translator.

Fig. 2. Contenido parcial del fichero recuperado.

En este documento se detallan instrucciones de la organización criminal para llevar a cabo sus planes. Los criminales informan a sus clientes que de ahora en adelante las comunicaciones irán firmadas por precaución con una clave pública PGP cuyo ID se facilita. Además, se invita a extraer toda la metainformación (datos Exif) del fichero. Si bien el documento no cuenta con información Exif, es posible extraer información de la imagen que se aprecia en la figura 2 por medio de diversas técnicas, (una de ellas el cambio de la extensión .docx a .zip en el que se localiza la imagen inalterada en el directorio *media*). Consultando la metainformación con, por ejemplo, Exiftool o FOCA se extrae una pista acerca de la identidad del presunto autor ("jcarlos").

Por otro lado, realizando una búsqueda de la clave PGP en repositorios de claves (como el de Rediris o pgp.circl.lu) se puede encontrar información asociada a la misma, en concreto el *uid:josch226*.

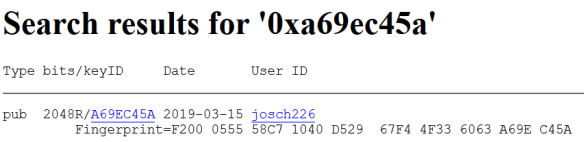


Fig. 3. Búsqueda del UID del ID de clave PGP pública.

De esta fase previa, podemos resaltar las siguientes lecciones aprendidas por los estudiantes:

- Refuerzo del uso de herramientas variadas tanto forenses como otras utilidades.
- En un análisis forense, es necesario extraer toda la metainformación de manera metódica, no se debe pasar por alto la que pudiera contener la imagen.
- Un fichero .docx se puede renombrar a .zip y acceder a su estructura e imágenes que contiene de manera directa.
- Búsqueda de claves PGP públicas en diferentes repositorios.

Fase 2

A partir de los datos anteriores ("jcarlos" y "josch226") se puede efectuar una búsqueda de lo que parece ser un nombre de usuario y el nombre del sujeto a investigar. Existen coincidencias no intencionadas con perfiles en redes sociales reales que no forman parte del ejercicio y que pueden llevar a resultados erróneos. La pista del nombre "jcarlos" se estableció para que fuese lo suficientemente genérica como para que no sea sencillo identificar al sujeto de no haber obtenido el *uid* de la clave PGP.

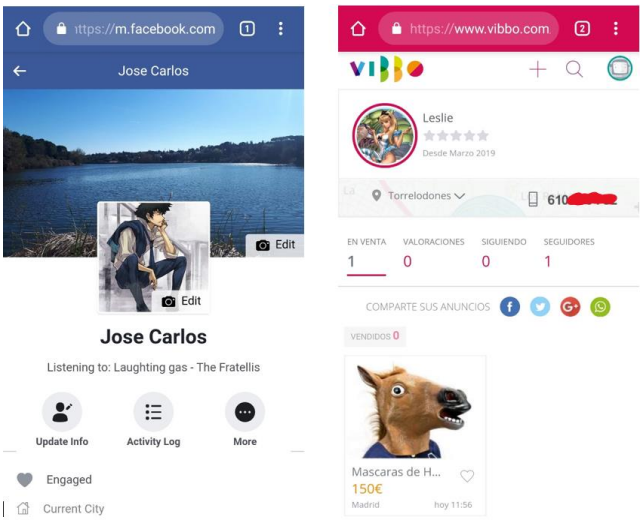


Fig. 4. Perfil falso en Facebook y anuncio en Vibbo.

Con ambos datos es trivial dar con un perfil de un sujeto llamado "José Carlos" en Facebook y comprobar que el sujeto expone ciertamente muy poca información que permita desarrollar el caso. No obstante, parece haber compartido un anuncio en Vibbo de un producto que vende su pareja.

Lecciones aprendidas por los estudiantes en esta fase:

- Búsqueda de personas en redes sociales no indexadas por buscadores ya sea mediante el uso de herramientas como Namechk, empleando el motor de búsqueda de las propias plataformas o mediante la modificación de la URL para dar con el perfil de usuario buscado.

Fase 3

Del anuncio de Vibbo, obtenemos un número de teléfono que una vez añadido a la agenda de un teléfono móvil que tenga instalado Whatsapp o Telegram, permite revisar el perfil público (estado y foto) en ambas plataformas.

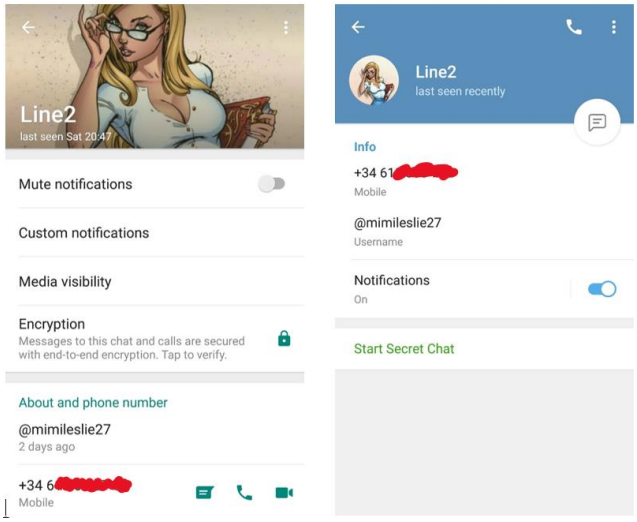


Fig. 5. Perfil en Whatsapp y Telegram.

Esto conduce a un identificador de usuario ("@mimileslie27") que con cierta agilidad se puede comprobar en Twitter o Instagram, siendo esta última la plataforma donde se encuentra un perfil de usuario sospechoso.

Lecciones aprendidas en esta fase por los estudiantes:

- Aunque una persona desee proteger su intimidad y minimizar su huella digital, su exposición pública también depende de su entorno cercano.
- Búsqueda pivote por nombre de usuario y teléfono en plataformas de mensajería y redes sociales populares.

Fase 4

Una vez el alumno llega a este punto, se requiere aplicar técnicas de GEOINT. Por el contenido del perfil de Facebook, en el que el sujeto investigado revela que vive cerca de un lago; y por el contenido del perfil de Instagram correspondiente a su pareja, del que se deduce que viven juntos y en Torrelodones, se invita al alumno a explorar cualquier mapa para localizar el único lago que hay en citada población.

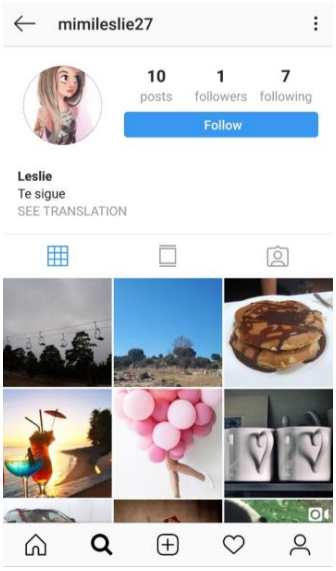


Fig. 6. Perfil en Instagram de la pareja del sujeto investigado.

Existen otros muy cercanos en otras poblaciones de la comunidad de Madrid (Galapagar y Las Rozas) con lo que es deber del alumno analizar con detenimiento la región.

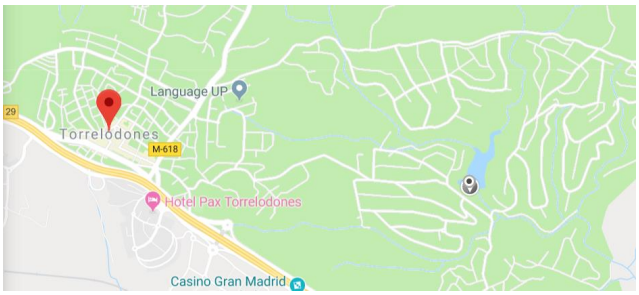


Fig. 7. Región geográfica en la que el investigado reside.

A partir de ahí, el alumno puede efectuar una búsqueda de información geoposicionada en Twitter y dará con el perfil de un amigo del investigado que acudió a una fiesta en su casa. Inocentemente, esta persona ha geoposicionado un *tweet* en la dirección exacta. Además, dado que el criminal debe ganar grandes cantidades de dinero y dispone de un coche de lujo, su amigo fascinado habrá capturado y publicado un video del vehículo en el que se puede percibir la matrícula. Por último, en ese mismo *tweet* se expone el apellido del sujeto con lo que su nombre y apellidos quedan finalmente revelados como "Jose Carlos Herranz". Con ello, podemos buscar en una red profesional como Linkedin y obtener el perfil del investigado y de éste, su dirección de correo electrónico.

Lecciones aprendidas en esta fase por los estudiantes:

- Técnicas de GEOINT, manejo de cartografía búsquedas de referencias geográficas.
- Búsqueda de información geoposicionada, en este caso en y mediante Twitter.
- La exposición de información personal se maximiza cuanto mayor es el entorno social del sujeto.
- Datos y acciones a priori inocentes pueden ser comprometedoras, incluso para un tercero.

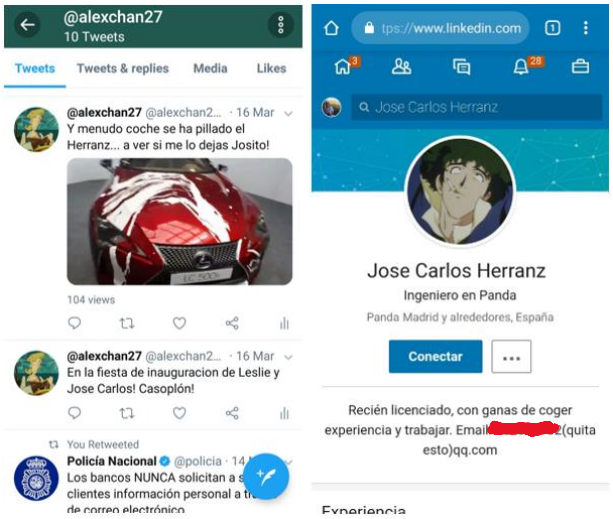


Fig. 8. Perfiles del investigado en Linkedin y de su amigo en Twitter.

Fase 5

Una vez el alumno se halla en posesión del correo electrónico del investigado, se le invita a que averigüe si éste figura en algún *leak* de datos, por ejemplo, mediante el uso de *HaveIBeenPwned*, que ofrece un positivo. Como se detallaba previamente en el apartado de materiales de este artículo, se suministra al alumno (por medio de un enlace de descarga en el enunciado de la actividad) una versión modificada de dicho leak. Para evitar que el alumno pueda avanzar hasta este ejercicio sin haber completado los anteriores, el fichero se encuentra protegido por contraseña, en este caso, el nombre de la plataforma que sufrió el *leak* "Taobao". Se le sugiere al alumno que tal vez el sujeto investigado ponga contraseñas similares. Una vez accedido al *leak*, compuesto por varios ficheros de texto de gran tamaño, una simple búsqueda del correo objetivo mediante los comandos *find* o *grep* permite dar con una contraseña que sigue un claro patrón y permite deducir de manera sencilla el password del fichero protegido por contraseña que se extrajo de la imagen EnCase al comienzo del ejercicio.

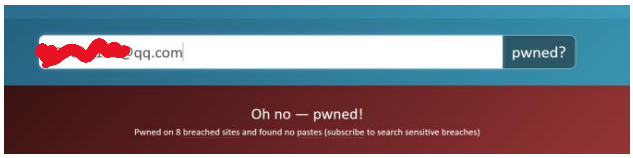


Fig. 9. Positivo del correo del investigado en *HaveIBeenPwned*.

Lecciones aprendidas en esta fase por los estudiantes:

- Manejo de leaks de credenciales y exploración de cuentas de correo comprometidas.
- Los usuarios suelen establecer contraseñas similares y siguiendo patrones predecibles.
- Entrenamiento en el *guessing* de dichos patrones.

En el fichero final, se detalla el plan de operaciones de la próxima venta de la organización criminal. En el documento se expone el método de pago, vinculado a una dirección de Bitcoin arbitraria. El alumno debe consular el saldo actual y ciertas gráficas y movimientos de la cuenta para establecer deducciones acerca de previas operaciones de la banda basándose en las transacciones del registro o "*ledger*" público.

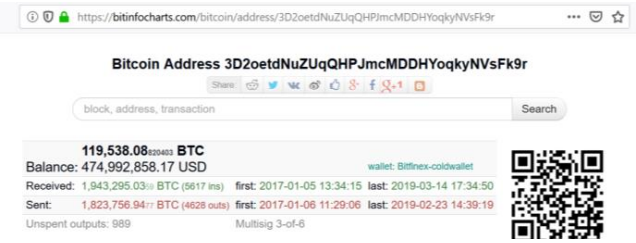


Fig. 10. Detalles de la dirección de Bitcoin vinculada a la investigación.

En concreto, la dirección empleada cuenta con una transacción de entrada de importe 1337 BTC que se utiliza como guiño final del ejercicio y cierre de la actividad.

Lecciones aprendidas en esta última fase por los estudiantes:

- Consulta del registro público de transacciones de criptomonedas.
- Introducción a la *Finantial Intelligence*.

IV. RESULTADOS

La práctica se propuso como actividad docente evaluable a un grupo de 29 estudiantes del Máster de Ciberseguridad de la Universidad Politécnica de Madrid. Una vez vencido el plazo de la entrega, se procedió a elaborar una encuesta voluntaria y anónima a los alumnos que permitió evaluar la dedicación, opinión sobre el formato de la práctica y su grado de aprendizaje y satisfacción del ejercicio.

La encuesta consistió en una serie de preguntas para ser evaluadas en una escala en el intervalo [0, 5] donde un cero equivale a "nada de acuerdo" y un cinco expresa "muy de acuerdo". Todas las preguntas se formularon de manera que una mayor puntuación es siempre una valoración positiva. Por otro lado, el cuestionario también recoge datos acerca del nivel de dificultad percibido, el desempeño en horas del alumno para su resolución y si requirió o no de ayuda. Finalmente, el cuestionario incluye un campo abierto para que los alumnos que lo deseen sean libres de expresar otras observaciones.

La encuesta fue respondida por 26 de los alumnos, los cuales componen la muestra a partir de la cual se han podido recoger los resultados expuestos a continuación que reflejan una valoración formal acerca del ejercicio, su formato y su enfoque:

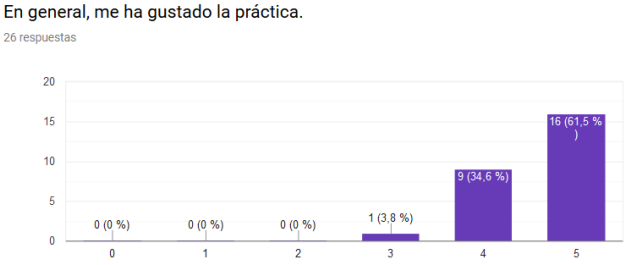


Fig. 11. Resultados obtenidos sobre la afirmación "Me ha gustado la práctica".

Los alumnos consideran que el formato de la práctica es innovador (min: 3, moda: 4, media: 4.34); indican que la práctica les ha resultado muy interesante (min: 4, moda: 5, media: 4.65) y por unanimidad, expresan que les ha gustado la práctica (min: 3, moda: 5, media: 4.57).

De cara a evaluar las técnicas de *gamificación* y *storytelling* incorporadas, los alumnos valoran positivamente que la práctica esté dividida por apartados de manera que se percibe el punto en el que se encuentran y consideran que además esta sensación de progreso invita a seguir adelante hasta terminarla (min: 3, moda: 5, media: 4.2); consideran que la práctica les ha enganchado hasta completarla (min: 2, moda: 5, media: 4.30) y además la califican como muy divertida/entretenida (min: 3, moda: 4, media: 4.38).

Aunque se ha valorado positivamente el planteamiento de la estructura narrativa; existe una mayor divergencia de opiniones acerca de si prefieren (mayor puntuación denota mayor preferencia) una práctica enlazada por medio de una estructura narrativa frente a exactamente los mismos ejercicios expresados de manera inconexa, sin hilo conductual (min: 2, moda: 5, media: 4.1).

Acerca de la dificultad del ejercicio, consideran que es asequible (min: 2, moda: 3, media: 3.11).

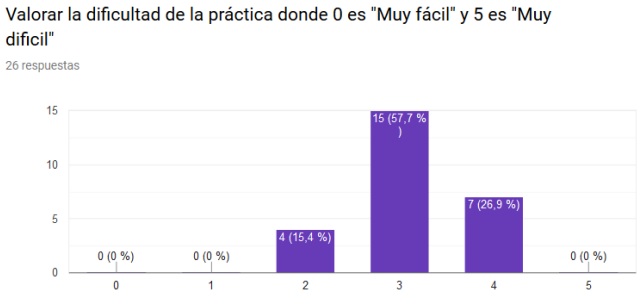


Fig. 12. Evaluación de la dificultad percibida.

A su vez, expresan que de media dedicaron a su resolución aproximadamente 5 horas si tomamos el punto medio de cada intervalo indicado en la figura 13:

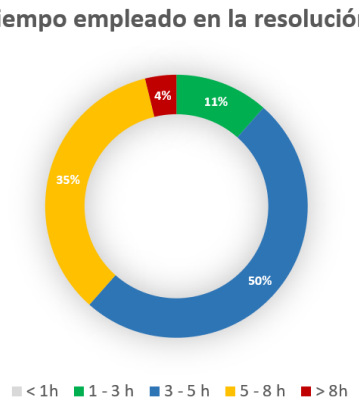


Fig. 13. Tiempo dedicado a la resolución del ejercicio.

Y a la vez están de acuerdo y perciben que la extensión del ejercicio es razonable (min: 2, moda: 4 y 5, media: 4.38).

A la pregunta de si han necesitado ayuda (si o no) para resolver el ejercicio, se les ofreció la posibilidad de indicar como tercera opción si requirieron de pistas pero únicamente como vía de ahorro de tiempo más que por no lograr avanzar; 7 alumnos se decantaron por esta opción. Los resultados de esta pregunta son significativos dado que tan sólo 5 alumnos lograron resolver la práctica sin ayuda, habiendo quedado bloqueados en algún punto de la resolución 14 de ellos.

Por último, a nivel de aprendizaje, se formula la pregunta de si han aprendido algo con el ejercicio; manifestando el 100% de los alumnos haber aprendido técnicas, conceptos, lecciones o herramientas nuevas aparte de poner en práctica algunas ya conocidas. En contrapartida, es menos valorado el interés suscitado por la disciplina trabajada (min: 0, moda: 5, media: 3.84):

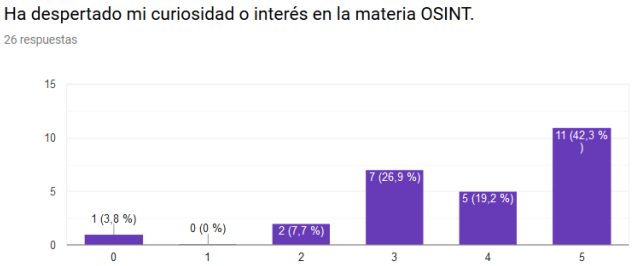


Fig. 14. Interés en la disciplina OSINT suscitado tras la práctica.

Por último, en las opiniones cualitativas vertidas, ya sea por medio del cuestionario de manera verbal, existen de manera generalizada opiniones muy positivas entre las que se expresa que el ejercicio está muy trabajado y ha resultado muy entretenido. Se vierten opiniones que establecen la práctica como un ejercicio que engancha y son varios alumnos los que manifestaron no haber "soltado" la práctica hasta haberla resuelto incluso quedándose hasta altas horas de la noche en su empeño. Es destacable que varios alumnos manifestaron haber compartido el ejercicio con su pareja o familiares y que les sirvió tanto a ellos mismos como a sus allegados de reflexión acerca de la privacidad y la huella digital. Varios alumnos contactaron con los autores del ejercicio preguntando dónde podían encontrar más ejercicios de OSINT y recomendación de libros con los que profundizar la materia.

V. CONCLUSIONES Y LÍNEAS FUTURAS

Las conclusiones extraídas del ejercicio es que, en el marco de la disciplina OSINT, se validan los citados trabajos previos que concluyen que las técnicas de *gamification* y en concreto los ejercicios CTF son una manera entretenida y estimulante para el aprendizaje de ciberseguridad. El enfoque de *storytelling* permite dotar de contexto y sentido al objetivo de la práctica en pos de la consecución de sucesivas metas (flags) pero con una sucesión lógica en la que se concatena la solución de cada apartado como punto de partida para el siguiente; resultando en una sensación de recompensa y progreso que los alumnos han considerado satisfactorios.

Por otro lado es conveniente destacar que el formato del ejercicio permite graduar su dificultad y longitud en el enunciado por medio de pistas; si bien en este caso el resultado ha sido que la resolución ha requerido una mayor

dedicación de la prevista. Puede ser considerada como contrapartida de la modalidad de ejercicios CTF la estimación de tiempo requerido por parte de los autores de los ejercicios, por lo general inferior a la real. A su vez, otra contrapartida de este formato orientado al ámbito académico es el posible bloqueo que pueden experimentar algunos alumnos durante la resolución, dando como lugar el intercambio de soluciones de algunos apartados y que conllevaría una necesidad de examen para revalidar el desempeño y así descartar "fusilamientos" del ejercicio. No obstante es destacable que no existe abandono ni renuncio del ejercicio.

Como líneas futuras, dado que la actividad ha sido evaluada de manera satisfactoria, se considerará su reutilización en posteriores cursos académicos teniendo en cuenta las opiniones vertidas por los alumnos; por ello es posible que se efectúen reajustes en la dificultad disminuyendo la ambigüedad de algunos puntos del enunciado que pudieran dar lugar a confusión y dificultar su resolución a la vez que se reduce la duración del ejercicio. La digitalización del ejercicio para su acceso online es otro factor a considerar dado que aparte de facilitar la evaluación y la introducción de *flags* por parte del alumno por medio de su comprobación automática, abriría numerosas y nuevas posibilidades para enriquecer los componentes lúdicos y narrativos.

AGRADECIMIENTOS

Los autores de este artículo quieren expresar su sincero agradecimiento a todos y cada uno de los alumnos, compañeros y amigos de la promoción de 2018-2019 del Máster de Ciberseguridad de la UPM y a todos sus profesores, en especial a Juan Alberto de Frutos por marcar la diferencia con su excelente material docente y a Jorge Dávila Muro y Socorro Bernardos por su paciencia inagotable y su siempre enriquecedora visión y consejo.

REFERENCIAS

[1] Plataforma de retos CTF Atenea del CCN-CERT. https://atenea.ccn-cert.cni.es/

[2] Plataforma de retos CTF HackTheBox. https://www.hackthebox.eu/

[3] Gavas, E., Memon, N., & Britton, D. (2012). Winning cybersecurity one challenge at a time. *IEEE Security & Privacy*, *10*(4), 75-79.

[4] Chothia, T., & Novakovic, C. (2015). An offline capture the flag-style virtual machine and an assessment of its value for cybersecurity education. In *2015 {USENIX} Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*.

[5] Caulkins, B. D., Badillo-Urquiola, K., Bockelman, P., & Leis, R. (2016, October). Cyber workforce development using a behavioral cybersecurity paradigm. In *2016 International Conference on Cyber Conflict (CyCon US)* (pp. 1-6). IEEE.

[6] Radoff, J. (2011). *Game on: Energize your business with social media games* (pp. 24-32). Hoboken. NJ: Wiley.

[7] Kember, D., & Leung, D. Y. (2005). The influence of the teaching and learning environment on the development of generic capabilities needed for a knowledge-based society. *Learning Environments Research*, 8(3), 245.

MOOC "Investigación en Informática Forense y Ciberderecho", experiencia y resultados

Andrés Caro Lindo[1]
José Carlos Sancho Núñez[2]
Universidad de Extremadura
Escuela Politécnica
[1,2]{andresc, jcsanchon@unex.es

Mar
Ávila Vegas
Universidad de Extremadura
Escuela Politécnica
jcsanchon@unex.es

Miguel
Sánchez Cabrera
Cátedra ViewNext-UEx
Escuela Politécnica
mscabrera@unex.es

Resumen- **Las universidades utilizan los MOOC como escaparate para atraer alumnos a su oferta de títulos académicos. Sin embargo, su alto índice de abandono y la gran cantidad de recursos necesarios para su puesta en marcha, ponen en duda el beneficio de este tipo de cursos masivos. Esta contribución pretende acercar la motivación, experiencia y resultados del MOOC "Investigación en Informática Forense y Ciberderecho" celebrado por la Universidad de Extremadura en la plataforma *Miríada X*, entre los meses de octubre y diciembre del año 2018. Se exponen las claves que han llevado a este MOOC a tener una tasa de finalización del 25,65% sobre el total de matriculados, un dato muy superior a la media de finalización con éxito que, generalmente, consiguen esta tipología de cursos gratuitos.**

***Index Terms*- MOOC, formación online, ciberseguridad, informática forense, ciberderecho, innovación formativa.**

Tipo de contribución: *Formación innovación*

I. Introducción

La informática forense, por un lado, y el ciberderecho, por otro, han venido despertando gran interés en los últimos años. Sin embargo, más allá del curso de Experto Profesional [1] que ofrece la Universidad de Extremadura (UEx) desde hace 5 años, no existen muchas otras ofertas formativas donde se *mezclen* estos conocimientos. El interés por estas temáticas suscitó la posibilidad de organizar una formación introductoria en formato MOOC (curso abierto, masivo y on-line), pese a conocer que el índice de abandono de los MOOC es muy elevado y su éxito de finalización ronda entre un 5% y un 10% con respecto al total de usuarios inscritos.

La UEx convocó el primer proyecto piloto en junio de 2016 para ofertar MOOCs, con el objetivo de promover este tipo de oferta formativa. Los cursos seleccionados se ofrecerían a través de la plataforma *Miríada X* [2].

Nuestro grupo de investigación decidió presentar una propuesta multidisciplinar, muy cercana al mundo real, impartida por 2 profesores de la UEx y 7 profesionales. Del total de 9 profesores, 3 son especialistas en informática forense, uno en *hacking* ético, 3 en investigación policial y 2 en ciberderecho. Siendo esta composición una de las principales causas y claves del éxito.

Los primeros problemas surgieron a la hora de grabar los vídeos del MOOC: la UEx dispone de estudios de grabación y profesionales en sus propias instalaciones. Habiendo 7 profesores externos a la UEx, y con residencias en Madrid, Zaragoza, Barcelona y Salamanca, el tema resultaba complicado, sobre todo *económicamente*.

Este artículo comparte la experiencia y resultados de la realización de este proyecto MOOC, que, se entiende, puede resultar de interés para otras Universidades.

II. Planificación del proyecto

Para facilitar la grabación, se convocó en Madrid a los 7 profesores externos, donde un equipo de nuestra Universidad se desplazó durante 2 días y desplegó un pequeño estudio de grabación en las oficinas de una multinacional colaboradora del proyecto. Las grabaciones, montaje y adecuación de todo el material audiovisual se realizaron a lo largo de 2017, quedando el MOOC listo para su oferta en el curso 2018.

Como indican González y Carabantes en [3] el 59% de usuarios que hacen un MOOC estarían interesados en obtener certificados oficiales de pago. Por lo que una vez revisado el material generado, e identificada la calidad del mismo, desde la UEx se sugirió que este MOOC se ofertase en formato *freemium*.

Esta modalidad permite al estudiante optar de manera opcional a un "Certificado de Aprovechamiento" emitido por la Universidad, incorporando un módulo final con acceso de pago. Para su obtención los estudiantes deberían superar las actividades obligatorias de los módulos de carácter académico (abiertos y gratuitos) y, además, matricularse y superar el módulo final con reconocimiento biométrico. Como ventaja, se reconocen los créditos ECTS del MOOC.

El curso cuenta con un total de 20 horas divididas en 4 módulos de contenidos de extensión similar y un sistema de evaluación basado en cuestionarios de tipo test.

Finalmente, el MOOC se presenta en *Miríada X* entre el 22/10/2018 y el 08/12/2018. La *Fig. 1* muestra un fragmento del vídeo de presentación de los docentes.

Fig. 1. Fragmento del vídeo de presentación de los docentes.

III. Resultados

La tabla I muestra los resultados totales y porcentajes de inscritos al curso, los alumnos que lo inician y finalizan y el porcentaje de finalización sobre los que inician.

Los porcentajes sitúan al MOOC como el que mejor tasa de inicio y finalización tiene de todos los ofertados por la UEx. El dato más relevante que aporta validez a la calidad del MOOC es que finalizan con éxito un 25,65% de usuarios sobre el total de matriculados. Un dato muy positivo y diferenciador que anima a los docentes a plantearse la realización de una segunda edición, ya que como se indica en [3] la tasa de éxito que habitualmente barajan los cursos abiertos masivos está entre un 5% y 10% de alumnos que finalizan todo el itinerario formativo sobre los matriculados.

Tabla I
INSCRITOS AL MOOC VS INICIAN VS FINALIZAN

Nº inscritos	Inician	Finalizan	Finalizan vs inician
3739	2593 (69,35%)	959 (25,65%)	36,96%

En *Fig. 2*, se muestra el perfil de edad de los participantes, mostrando interés la temática del curso casi por igual a grupos de personas de 18 a 55 años. En cuanto al sexo, del 73,9% es masculino, el 25,18% femenino y el 0,94% prefiere no contestar.

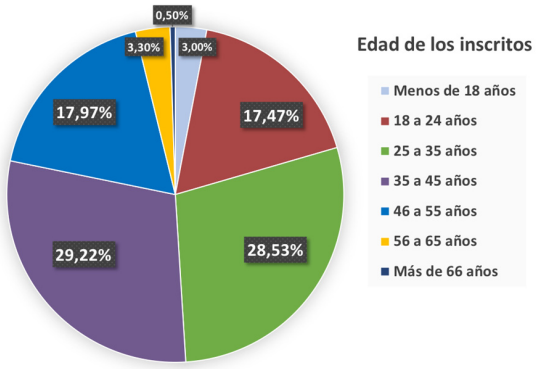


Fig. 2. Rango de edad de los inscritos.

En cuanto al nivel de estudios alcanzado por los participantes, observamos en *Fig. 3*, que la mayoría ha finalizado los estudios universitarios (36,63%), siendo también importante la presencia de estudiantes universitarios y titulados de posgrado.

La mayoría de los inscritos procedían de fuera de España, un (52,81%) frente a un 47,19% de procedencia española.

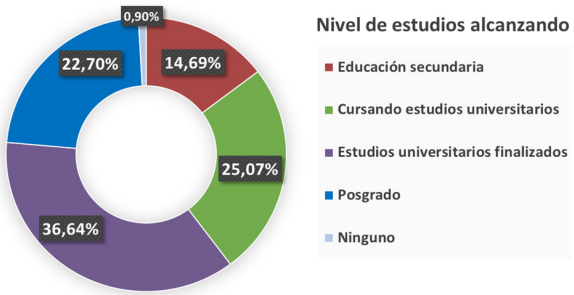


Fig. 3. Nivel de estudios alcanzados.

Las razones para inscribirse en este MOOC han sido varias como se observa en *Fig. 4*. Destacando que un 32,38% de los inscritos denota interés general por curso y la temática expuesta, el 25,27% manifiesta que la temática tiene relación con su trabajo, mientras que el 19,79% indica que la relación es con sus estudios.

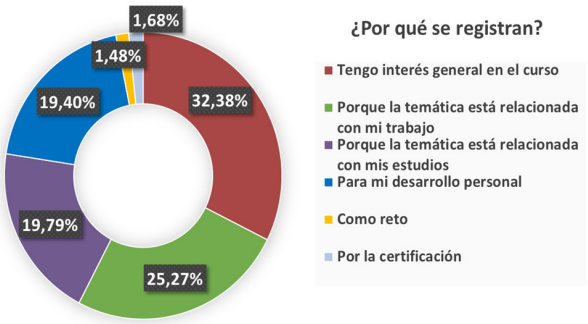


Fig. 4. Razones para hacer el MOOC.

En la encuesta final, destacan los buenos resultados en cuanto a expectativas cumplidas, nivel de satisfacción general y aplicabilidad de contenidos aprendidos a nivel profesional. Además, el 97,47% de los participantes recomiendan el curso y el 97,44% se inscribiría en un MOOC de la misma temática, pero de contenidos más avanzados.

Tabla II
RESPUESTAS DE LA ENCUESTA DE LOS PARTICIPANTES

Pregunta	%
¿Ha cumplido el MOOC tus expectativas?	
Ha cumplido mis expectativas	45,45%
Ha sido lo que esperaba	46,75%
No las ha cumplido	7,79%
¿Cuál es el nivel de satisfacción con este MOOC?	
Totalmente satisfecho	50%
Satisfecho	46,15%
Insatisfecho	
Usuarios que afirman poder aplicar los contenidos aprendidos a nivel profesional	85,71%
Usuarios que recomiendan este MOOC	97,47%
Usuarios que se inscribirían en otro MOOC	97,44%

IV. Conclusiones

Los resultados demuestran una gran acogida en formación que mezcla ciberseguridad y ciberderecho, impartida por profesores de universidad y profesionales de primera línea y habiendo generado contenido de calidad.

Agradecimientos

Los autores agradecen la financiación recibida por parte de la Junta de Extremadura (Fondo Europeo de Desarrollo Regional), Consejería de Economía e Infraestructuras (Proyecto GR18138).

Referencias

[1] "Curso de Experto Profesional en Derecho Tecnológico e Informática Forense." [Online]. Disponible: https://dtif.unex.es/.

[2] "MiríadaX." [Online]. Disponible: https://miriadax.net.

[3] Á. González de la Fuente and D. Carabantes Alarcón, "MOOC: medición de satisfacción, fidelización, éxito y certificación de la educación digital," *RIED. Revista Iberoamericana de Educación a Distancia*, vol. 20, no. 1, p. 105, 2016.

Design and Development of a Translation and Enforcement Module for Cybersecurity Policies

Fernando Monje, Víctor A. Villagrá
Universidad Politécnica de Madrid (UPM), ETSI Telecomunicación. Avda. Complutense 30, 28040, Madrid
f.monjer@alumnos.upm.es, victor.villagra@upm.es

Abstract- **Nowadays, cyber attacks constitute a bigger threat to organizations than before, given the higher sophistication of those attacks, their growing propagation velocity and the increase of their destructive capabilities. This problem requires solutions capable of answering in real time and automatically. The proposed solution is the development of a system capable of translating a high-level security policy designed by an organization into another low level policy, so that it can be interpreted by the elements of the network in charge of the security. In such a manner, it is possible to design the security policy independently of the network elements. The policy is implemented in real time accordingly to the dynamic risk of the organization. This risk calculation will be carried out using the data obtained by an Intrusion Detection System (IDS) monitoring the organization's network. System efficiency will be validated with two virtualized scenarios using different network topologies.**

***Index Terms*- Cyber Attack, Risk, Cyber Security, Iptables, Firewall, Automatic Response System, Security Policies.**

Tipo de contribución: Investigación en desarrollo

I. INTRODUCTION

The increasing complexity of implementing security measures in organization´s network demands for new approaches. The aim of this work is to present a way for easing the process of designing, maintaining and enforcing the cybersecurity policy of an organization and to allow organizations to automatically respond to cyber threats in real time. Most approaches to the design of security policies, either lack the connection to devices in charge of detecting threats, in order to react to the new context, or the design of the policies becomes very complex. Those are the main areas this work attempts to improve.

To do so, the approach taken in this project aims to let the security administrator write the policies in a high-level language, independent from the network 's topology and from the technology used by the security enforcement elements. Even though there have been efforts in the scientific community in this regard, this approach has two main advantages compared to the solutions already in the market:

- It allows the use of already installed equipment, just with routing capabilities, as the system core has been agnostic from the devices, to allow any low-level implementations, therefore it use may be cheaper than solutions requiring specific software or hardware.
- Many solutions, as XACML reference architecture [1] rely on a Policy Decision Point to decide and several Policy Enforcement Point to enforce the policies. The approach in this paper implements the decision layer at each enforcement point, creating a decentralized solution, therefore improving the reliability of the system.

The solution proposed, detailed in section III, uses an RBAC model to configure the security policy, then it maps the organization network to translate the high-level policy to a low level, which in this paper is Iptables. Finally, an integration with an IDS is proposed, to provide the system the necessary cyber-situational awareness to adapt the organization policy in real time.

In order to accomplish the aforementioned objectives, the process can be divided into four goals:

- The first goal is to design a high-level policy definition system.
- The second goal is to design a system able to understand the high-level policies and transform them into a language understandable by the security enforcement elements.
- The third goal is to upload the translated policies from where they are generated to the concerned security elements automatically. It has to be done without human interaction and in a secure way.
- The fourth and last goal is to design a communication and decision system which will allow an IDS to communicate with the device in charge of translating the policies. Hence, allowing the inference of the risk level of the organization and applying the adequate security policy.

The paper includes a review of the state of the art in the area, in order to evaluate its applicability to the analyzed problem, followed by a description of the proposed system and its validation in two scenarios.

II. STATE OF THE ART

There have been several efforts in creating security policies languages, which are key in managing big organizations. In today's networks, there are various types of elements from several vendors with different OSs, such as Linux based distributions, or Windows, and each OS allowing different configurations. The design of a security policy that takes into account all the previously mentioned possibilities is very challenging. That is the reason why there have been

efforts in making the security policies independent from the component in charge of enforcing it. Such is the approach taken in [2], where authors present an Or-BAC model (Organization Based Access Model). The policies are written using this model, then translated into an intermediate level of abstraction, still agnostic from the component where the rules are going to be deployed. This approach is the same taken in [3], where this abstract level is closer to the real organization network's topology. There are more ways to specify a Security Policy, and their benefits and limitations are discussed in [4] where the authors present the most used policy languages. Some of the most distinguished are:

Ponder: It is an object-oriented and declarative policy language [5]. It uses the RBAC model (Role-Based Access Control). The interaction between roles are defined as relationships. Ponder is not XML based but it can be translated into an XML representation if needed. The element in charge of enforcing the policies is called PEP (Policies Enforcement Point) and is written in Java [6].

EPAL: XML-based language proposed by IBM [7]. It is focused on privacy policies and unlike other policy languages it's not very easy to implement correctly. It uses a purpose-based access control, where the authorization decision is evaluated by the subject's purpose of using the requested service. This access control mechanism is easier than RBAC but requires a well-structured purpose element. IBM has considered this problem and thus palliated this limitation by providing tools to facilitate the implementation of this policy language [8]

VALID: (Virtualization Assurance Language for Isolation and Deployment). Language to express high-level security goals especially useful in virtualized infrastructures hosted in the cloud. It is based on IF (Intermediate Format) [9], which helps building a formal foundation to facilitate automated reasoning. It can be used to write access control rules, so it is able to deploy them automatically and then validate them.

XACML: (eXtensible Access Control Markup Language) It is the most used one. XACML is a declarative language, based on XML. One of the main features it provides is the policy combination procedure, used when more than one policy is applied, using several algorithms [10] to avoid conflict. It also has the possibility of adding additional algorithms. A tool called Sun's XACML Open Source Implementation [11] is commonly used for the implementation and enforcement of policies written in XACML. It only provides limited support for the RBAC model, but a more in-depth integration has been done in xfACL (eXtensible Functional Language for Access Control) [12].

However, in recent years the need of having dynamic policies enforcement has become clear. The process of changing them manually has been the main procedure but, nowadays, it is too slow to be effective against threads and prompt to human mistakes and misconfigurations, due to the often high complexity level of the policies.

So, in order to detect and automate the response to cyber threads, AIRSs (Automatic Intrusion Response Systems) were defined. These systems rely on IDS (Intrusion Detection System), which are in charge of detecting the intrusion and notifying it to the AIRS. In order to work, AIRS rely on metrics to choose the optimal response in each scenario depending on the context and other parameters, such as the importance of the network element being compromised, the cost of the possible response, the IDS confidence and several more. Various AIRSs have been proposed in the recent years. Some of the most important ones are:

Stakhanova's IRS [13]: This system is mostly based on two different metrics.

- Damage reduction metric: It compares the damage the intrusion could cause and the damage the response could originate and chooses the least harmful response which mitigates the attack.
- Maximum benefit at the lower risk metric: This metric also considers the negative impact that the response could have as well, but it also ponders the success of the responses already taken in the past.

AAIRS [14]: It considers three metrics:

- IDS confidence metric: This AIRS measure the rate of false positives the IDS have.
- Attack identification metric: The system detects the type of the intrusion and reacts accordingly.
- Response success metric: Also considers past responses and their achievement level.

IDAM&IRS [15]: It is one of the newest AIRS. It has a set of responses associated with a certain risk level. Whenever an intrusion is detected, a risk level is calculated, and if this risk is greater than the one associated with a possible response, the response is deployed. It is a similar approach to the one proposed in this work.

The main problem that occurs today lies within the connection between the security policies languages and the AIRSs. Either the policies are too complex to be managed by the AIRSs, which then have to function as an independent device, or they have to be configured manually according to the policy. The aim of this work is to reduce the complexity, by automating the process and making it simpler for human comprehension.

III. DEVELOPMENT

The development of the system has been divided in two main features: The translation of the security policies functionality and the automatic intrusion response feature.

In order to achieve the first functionality, the systems must be able to translate the security policies from a high level to a low level. Therefore, reducing the complexity of the process by making it independent of the network 's security enforcement elements, such as firewalls.
The process has been divided in several steps as shown in Fig. 7

A. Configuration

The high-level security policy is based in the RBAC model and requires four input files written in XML format in which the policy will be defined.
However, a GUI (Graphical User Interface) has been developed to avoid an administrator having to know XML syntax. It automatically displays the configuration in the files and can change it under request. It is able to manage all four files. Fig. 1 shows the main screen of presented to the administrator:

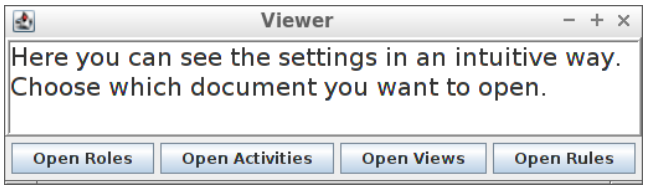


Fig. 1. Main screen of the GUI.

Roles
It is the document where the roles are written. A role represents a network element or a set of them and it consists of:

- **name:** It is the name of the role and it can be any set of alphanumeric characters with three exceptions. If it is a firewall, it must follow this pattern: FW_«name of the firewall»_«interface». It is required that the different interfaces of the same firewall share the same «name of the firewall». Another exception is that the role considered Internet has to be called «Internet». The last one is that the name of the administrator role has to be «Admin».
- **addr and mask:** These fields are the IP address and the network mask of the role. There can be any combination between them.

There is an optional field called «hostExclusion», which is made of one or more roles. This field can be used if you want to exclude a certain role whose address and mask are contained in the ones defined in your role. An example can be found in Fig. 2.

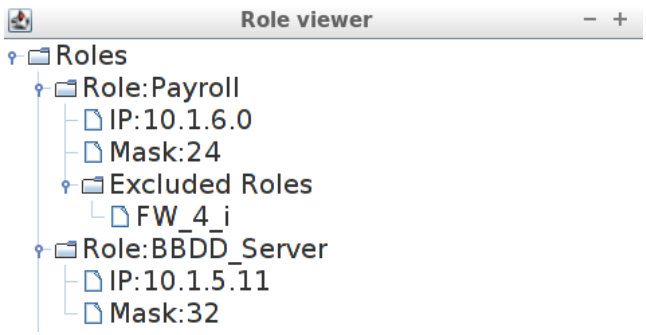


Fig. 2. Expanded role screen of the GUI.

Activities
The activities are the group of possible actions that can be made in an organization. They have three fields:

- **relevantActivity:** In the attribute «name» of this tag is where the name of the activity has to be written, it can be any combination of alphanumeric characters.
- **protocol:** In the attribute «type» of this tag is where the protocol of the activity is defined. It can be «tcp», «udp», or «icmp».
- **destPort:** This field is needed in the case that the protocol is «tcp» or «udp». It can have one or more singlePort fields inside. Each singlePort field is filled with the number of the port that shall be allowed.

An example of an activity is provided in Fig. 3.

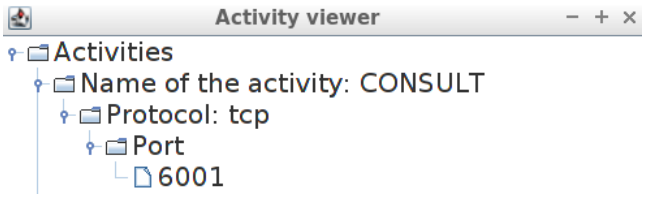


Fig. 3. Expanded activities screen of the GUI.

Views
The views are the possible roles to which access can be granted. Each one has two fields:

- **relevantView:** In the attribute «name» it has to be specified the name of the view, it does not have any restrictions.
- **toTarget:** In the attribute «roleName» is where the name of the role that can be reached has to be written.

An example of a view can be seen in Fig. 4.

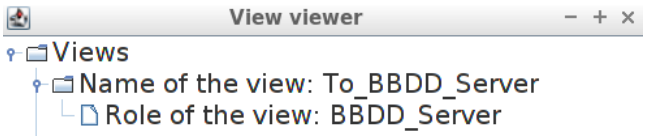


Fig. 4. Expanded views screen of the GUI.

Permissions
This is the file where the security policy is written. The permissions are classified in risk levels from 0 to 10, being 0 the lowest risk and 10 the highest. This file has two main sections:

Info: This section is made of three XML tags:

- **companyNetwork:** In this tag the organization's IP and mask has to be written in CIDR notation, which has the following format, «xxx.xxx.xxx.xxx/yy» being the «x» the numbers corresponding to the Internet address and «y» to the mask.
- **initialCompanyRiskLevel:** The input is a number from 0 to 10. It specifies the risk level the organization starts at when the module is executed for the first time.
- **automaticRulesUpload:** This tag accepts two options: «yes» or «no». If the affirmative option is selected the rules will be uploaded automatically to the firewalls. Nonetheless, if the choice is «no», the rules will only be written in plain text format inside a file allocated in the same folder as the executable program is at. Below, an example of an Info section containing the three tags can be found:

```
<Info>
    <automaticRulesUpload>yes</automaticRulesUpload>
    <initialCompanyRiskLevel>0</initialCompanyRiskLevel>
    <companyNetwork>10.1.0.0/16</companyNetwork>
</Info>
```

Rules: In this section is where the permissions are written. Each permission is a rule. They are classified in ten risk levels, each risk level holding one or more permissions. An example of a permission is given in Fig. 5. A permission has the following attributes:

- **roleName:** It is the name of the role from where the resource is going to be accessed.
- **activityName:** It is the activity's name that is going to be performed over the view by the role.
- **viewName:** It is the name of the view that is going to be accessed by the role. Below is an example of the risk level 0 with two permissions inside:

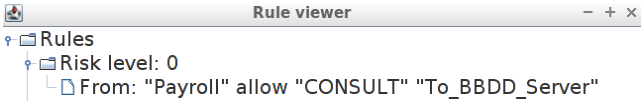


Fig. 5. One permission at risk level 0.

As said before, all the files can be edited also inside the GUI, as example of the creation a rule is shown in Fig. 6.

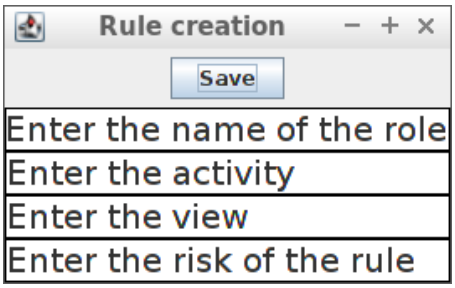


Fig. 6. Rule creation with added excluded roles fields.

B. Translation

Once the high-level policy has been defined, the system has to translate it into an abstract level, still agnostic from the language of the network devices but that takes into account the network's graph of the organization. The process is the following:

Translation to an abstract level

The first step is discovering which roles are relevant to which firewalls. A role is relevant for a given firewall when the firewall can reach it in one hop. To obtain this information, an algorithm has been developed to check the IP and mask of each role subnet and see if it contains an IP interface of a firewall. If it does, then it can reach it in one hop as they are in the same subnet. It also alerts of wrong IP and mask allocation as for example, 10.1.0.128/24.

Next, the program will start translating the set of permissions corresponding to the risk at which the organization starts, as defined in the «initialCompanyRiskLevel» tag of the Permissions file. So, for each rule of the set, there will be the role which wants to access the resource (from now on subjectRole), the type of access the permission allows, defined by its protocol and ports if needed, and lastly the role trying to be accessed (from now on viewRole).

The system has to obtain the firewall or firewalls that are between the subjectRole and the viewRole. In order to know in which ones, the rules will have to be written to allow the network traffic.

To obtain the firewalls, another algorithm has been developed, which queries the routing elements (in this case, all routing elements are also firewalls). It starts by looking in the relevant roles list, obtaining the list of firewalls that have as relevant role the subjectRole and the ones that have the viewRole. Doing so, we now have two lists, the first one represents the firewall candidates to be the first firewall to encounter the traffic, and in the second list the candidate of being the last firewall to manage the traffic before it reaches its destination. So, it takes the firewall address of the candidates of the first list, and executes the following command:

```
ssh root@"firewall address" ip route get "destination subnet"
```

Then, it analyzes the response. If the petition of the destination address is not reachable in one hop by the asked firewall, the response is going to look like this:

```
"destination address" via "next hop address" dev eth0 ...
```

being the important information the field «next hop address», which is the next routing element used to reach the «destination address». The other possibility is having the destination at one hop, in this case the response would look like this:

```
"destination address" dev eth0 ...
```

meaning that there are no more network routing elements involved in this traffic.

Using all this information and considering all the possibilities, the affected firewalls by the permission are calculated. The approach has been a restrictive one, not allowing anything except what is permitted. With this information, the abstract rules can be generated, specifying for each network element, in this case firewalls, the rules to implement. An example for FW_1 can be seen:

```
<Firewall IP="10.1.2.1/32" name="FW_1">
    <sourceIP>10.1.1.0</sourceIP>
    <sourceMask>21</sourceMask>
    <destinationIP>internet</destinationIP>
    <destinationMask>0</destinationMask>
    <protocol>udp</protocol>
    <destinationPort>8000</destinationPort>
    <excludedViewRole
        roleName="FW_1_e">10.8.1.1/32</excludedViewRole>
</Firewall>
```

It is important to remark that not every firewall has the same amount of information. For example, in this case, «FW_1», has the «excludedViewRole» tag. This has been done because that firewall is the first one the traffic would go through, so, if it is the strictest one, the rest of the firewalls do not need that

information, thus optimizing the system. This advantage will become more noticeable when the concrete rules are generated as each firewall will only have the strictly necessary rules, avoiding unnecessary workload for them.

Translation to a concrete level

With these abstract rules, the system can translate them to any firewall language. In this work the one chosen has been Iptables. For each permission a chain of Iptables is created. One simple example of a chain is the following:

```
iptables -N FW_1-SSH-To_Admin
iptables -A FORWARD -s 111.222.100.1/32 -p tcp --dport 22 -j FW_1-SSH-
    To_Admin
iptables -A FW_1-SSH-To_Admin -d 111.222.2.20/32 -j ACCEPT
```

For all the chains this rule is also added:

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

It allows traffic in both directions, matching the replies with the allowed packets and letting them through. If it is not written, then the traffic can only go in one direction, which is not very useful, as for example, TCP protocol has feedback with ACKs messages.

As a summary, this functionality allows an administrator to write a security policy in a way that is independent of the type of the security enforcement elements the organization has.

C. *Automatic Intrusion Response*

The second functionality, which is the automatic intrusion response uses the potential of the first one. With few additions, the system is able to read alert messages sent by an IDS (Intrusion Detection System) and update all the security policy of the organization accordingly. When an intrusion is detected, an alert is sent via TCP protocol to the administrator‘s PC, where a socket is reading the incoming information. Depending on the type of intrusion, the system changes the risk of the organization, therefore reading a new set of permissions and doing the high to low level translation as explained in the former section.

Automatic Deployment

Another addition is to have a Configuration module, which configures the SSH to avoid asking for human interaction by helping the administrator adding the firewalls to the list of known hosts of the administrator´s PC. Then sends the public key of the administrator to all the firewalls to avoid being asked for the password when an SSH access is requested. Consequently, allowing the translation functionality to run uninterrupted when discovering the firewalls involved by the new set of rules.

Finally, an Automatic Rules Uploader has been added which takes the rules written for each firewall and sends them via SSH. If the Configuration module was used this process is done without any human interaction, adapting the organization‘s network security to the cyber threats in real time.

A diagram of all the systems can be appreciated in Fig. 7.

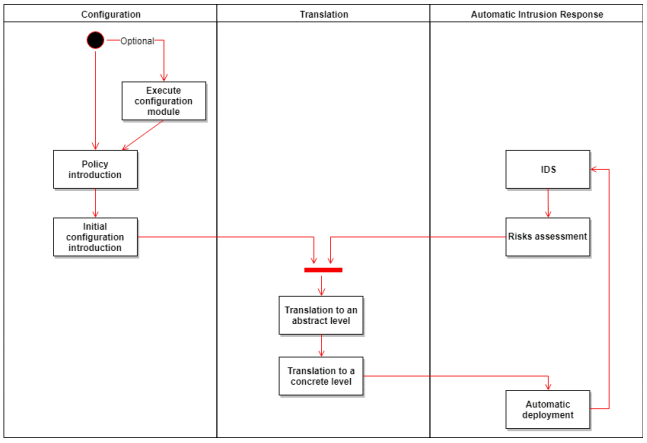


Fig. 7. System flow diagram.

IV. VALIDATION

The system has been tested in two scenarios, a simpler one, Fig. 8, and a more complex one. Both scenarios have been created using the virtualization software VNX [16].

A. *Internet Access*

The network of the first one is shown in Fig. 8.

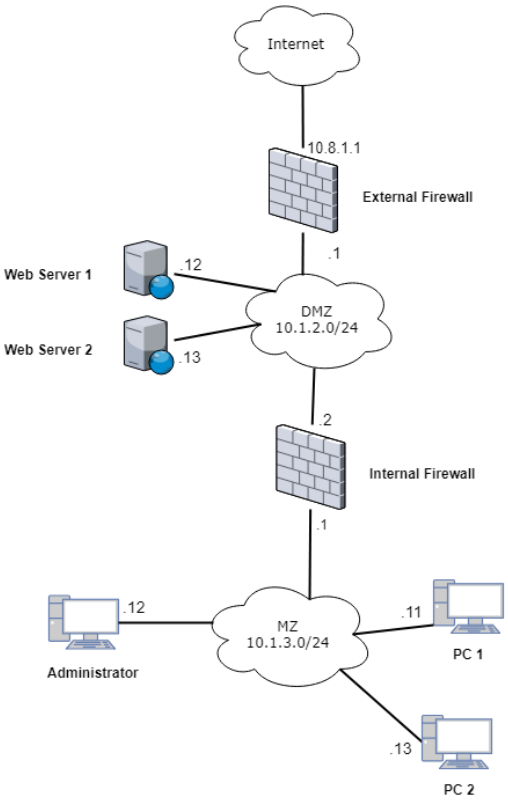


Fig. 8. First scenario network `s topology.

This topology is a common one when the company wants to offer services accessible from the Internet, like Web services in this case. Thus, DMZ (Demilitarized Zone) can be accessed through certain protocols and ports allowed in the External Firewall. Nonetheless, the MZ has very important assets and therefore is a stricter area where access is much more restricted and is monitored by the Internal Firewall.

For this example, the initial risk is going to be 0, which is the lowest possible. At this risk there are going to be two permissions written:

- The first one allows PC1 making a ping to Web Server 1 to check if the machine is up (by allowing ICMP transmissions).
- The second one allows the Internal network (MZ) to access the Internet using the ports 80, 8080 and 443 in order to enable HTTP and HTTPS communication to external web services, except the Administrator, since it would not be safe praxis.

The policy in the Permissions file can be seen using the GUI in Fig. 9.

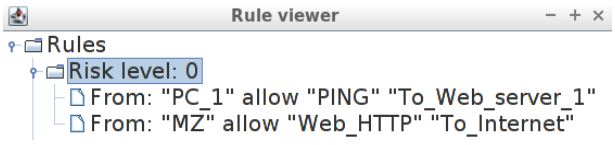


Fig. 9. Permissions of the simpler test case.

And the roles defined for the example are shown Fig. 10.

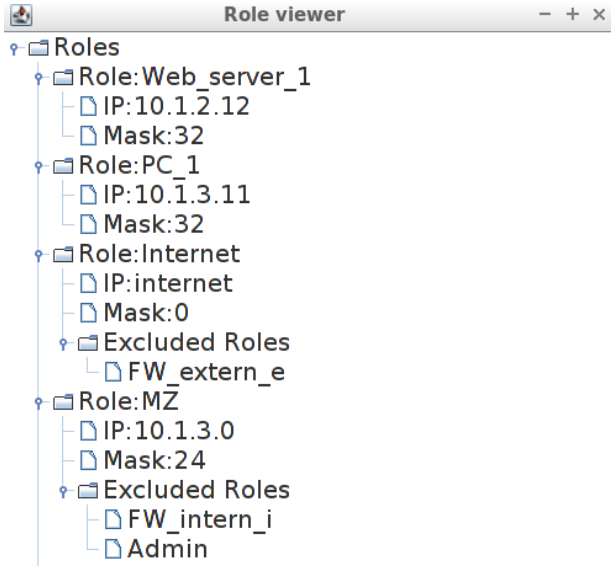


Fig. 10. Roles of the simpler test case.

The activities as described before, can be seen in Fig. 11.

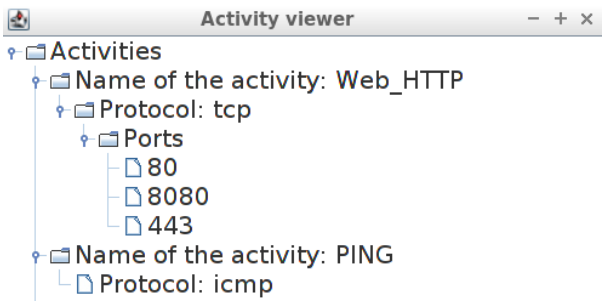


Fig. 11. Activities of the simpler test case.

The views are just the roles that are going to be accessed, in this case the Web Server and the Internet.

Finally, the system needs three more inputs in the info section at the permissions file. They are the initial risk of the organization, its network subnet and the automatic rules uploader option. So, after all the calculations described in the previous section, below it is shown how the AbstractionRules file looks like for the first permission:

```
<Firewalls>
    <Info>
        <subjectRole>PC_1</subjectRole>
        <activity>PING</activity>
        <view>To_Web_server_1</view>
        <companyNetwork>10.1.0.0/22</companyNetwork>
        <automaticRulesUpload>yes</automaticRulesUpload>
        <adminIP>10.1.3.12/32</adminIP>
    </Info>
    <Firewall IP="10.1.2.2/32" name="FW_intern">
        <sourceIP>10.1.3.11</sourceIP>
        <sourceMask>32</sourceMask>
        <destinationIP>10.1.2.12</destinationIP>
        <destinationMask>32</destinationMask>
        <protocol>icmp</protocol>
    </Firewall>
</Firewalls>
```

Also, the two rules created due to both permissions that affect the Internal Firewall are shown:

```
iptables -N PC_1-PING-To_Web_server_1
iptables -A FORWARD -s 10.1.3.11/32 -p icmp -j PC_1-PING-
    To_Web_server_1
iptables -A PC_1-PING-To_Web_server_1 -d 10.1.2.12/32 -j ACCEPT

iptables -N MZ-Web_HTTP-To_Internet
iptables -A FORWARD -s 10.1.3.0/24 -p tcp --match multiport --dports
80,8080,443 -j MZ-Web_HTTP-To_Internet
iptables -A MZ-Web_HTTP-To_Internet -s 10.1.3.1/32 -j RETURN
iptables -A MZ-Web_HTTP-To_Internet -s 10.1.3.12/32 -j RETURN
iptables -A MZ-Web_HTTP-To_Internet -d 10.8.1.1/32 -j RETURN
iptables -A MZ-Web_HTTP-To_Internet -d 10.1.0.0/22 -j RETURN
iptables -A MZ-Web_HTTP-To_Internet -j ACCEPT
```

However, in the External Firewall only one rule was written:

```
iptables -N MZ-Web_HTTP-To_Internet
iptables -A FORWARD -s 10.1.3.0/24 -p tcp --match multiport --dports
80,8080,443 -j MZ-Web_HTTP-To_Internet
iptables -A MZ-Web_HTTP-To_Internet -d 10.1.0.0/22 -j RETURN
iptables -A MZ-Web_HTTP-To_Internet -j ACCEPT
```

As it can be seen, the rules are only inferred for the concerned firewalls. The chain «PC_1-PING-To_Web_server_1» does not affect the External Firewall and therefore is not written in it.

B. Quarantine

A more complex environment has been designed to see how the system performs. The new network topology is displayed in Fig .8.

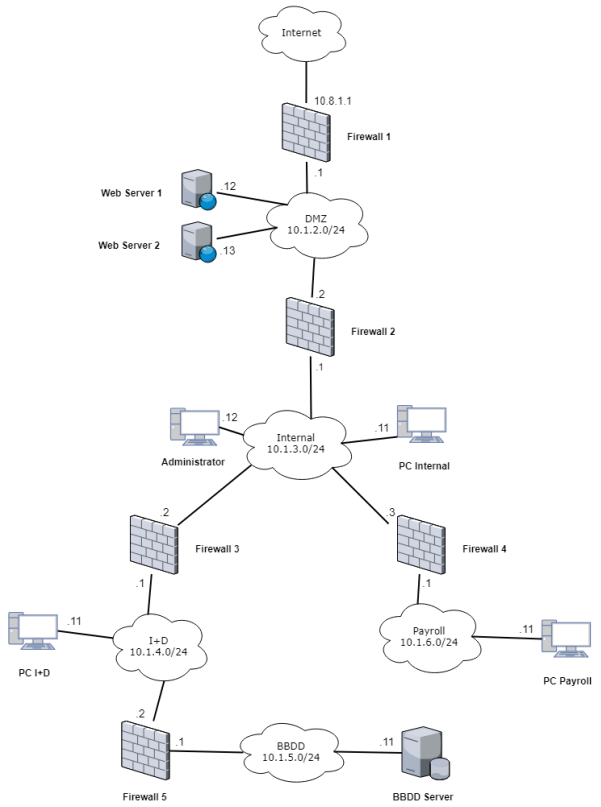


Fig. 12. Second scenario network topology.

In the following example the AIRS is going to be evaluated.

The test is going to be a network quarantine: the detection of an employee from the Internal department trying to access a malicious site and getting infected. To prevent any further damage, the department is going to be quarantined. This attack will occur when the «PC Internal» visits a web site and gets infected with malware, which is going to try to leak information using via FTP using the TCP port 20 to an unknown IP. The IDS is going to detect that IP as suspicious and will send an alert.

The initial state of the organization is having the risk at level 0. When the alert is triggered, it will be analyzed and the risk will be increased to level 2, isolating the internal department, as it can be seen in the permissions of Fig. 13.

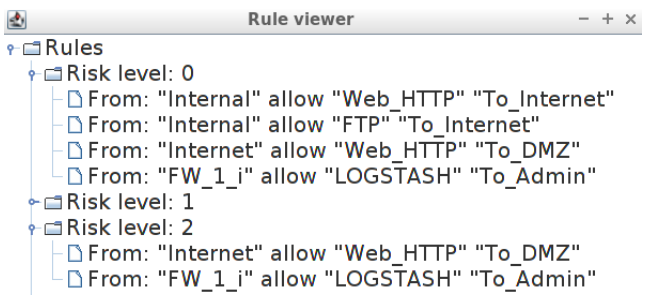


Fig. 13. Permissions of Quarantine test case.

The organization's network will start at a level 0 of risk. In Fig. 14, the Iptables rules of the Firewall 2 can be seen:

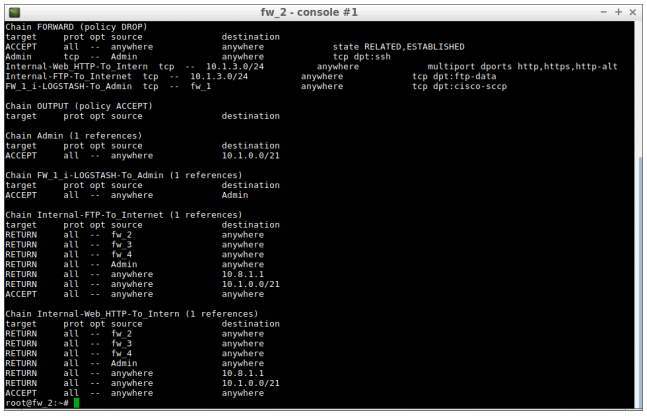


Fig. 14. Firewall 2 Iptables rules before the infection in Quarantine test case.

To detect the intrusion, an IDS is deployed in Firewall 1 monitoring all the traffic entering and going out of the organization. Checking malicious IPs sites has created a rule which looks like this:

```
alert tcp 10.1.3.0/24 any -> 83.223.12.3 20 (msg:"Alert, FTP to suspicious IP, level 2"; classtype:policy-violation; sid:2; rev:2;)
```

This rule tells the IDS to generate an alert when any traffic goes out with the default port of FTP with the malicious IP as destination. The alert will be saved in a log file called fast.log Then, the module Logstash, which is a tool able to manage and transport logs, is going to send that alert via TCP from the IDS to the port 2000 of the Administrator's PC, where it will be analyzed by the system to determine if the risk level has to be changed or not. The process is shown in Fig. 15.

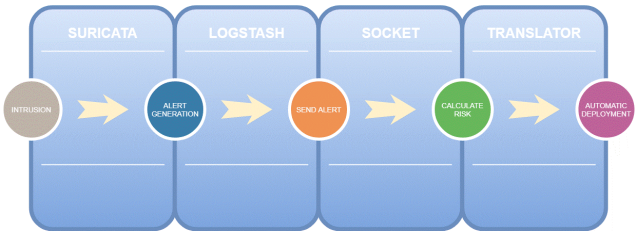


Fig. 15. Overview of the AIRS functionality.

The IDS rule is going to be triggered executing the following command in «PC Internal», which emulates a petition from a PC located at the Internal network, to a suspicious IP using the tool iperf from the command-line interface:

```
iperf -c 83.223.12.3 20 -p 20
```

Finally, analyzing the potential thread, the system is going to increase the risk level to two, loading the new permissions, and therefore, isolating the Internal department. After all the refinements, the new rules uploaded in firewall 2 can be seen in Fig. 16.

Fig. 16. Firewall 2 Iptables rules after the infection in Quarantine test case.

As it can be seen, all the rules allowing traffic to and from the Internal department have been deleted, thus, enforcing the quarntine.

V. CONCLUSIONS

The system proposed in this work, aims to make organizations as secure as possible with the highest efficiency. The focus has been put into designing it to operate for any organization, with an arbitrary type of network topology and with any kind of security enforcement elements.

An important objective of this project was making the system as user friendly and automatic as possible, hence the creation of a GUI to visualize and modify the security policies, and also the Configuration module, which sets up the SSH communications automatically.

In addition, taking advantage of SSH connections already configured, the system is able to send the rules inferred the concrete rules generator, autonomously. Furthermore, the approach taken during the development allows the system to be used by any type of firewall with the correct concrete rules' generator, making it compatible with most network's topologies.

The last main effort has been put on making the system able to connect to an IDS so that, with information of the threads or strange behavior detected, a risk level can be inferred, and, ultimately the security of the organization can be adapted in real time.

REFERENCES

[1] Lazouski A., Martinelli F., Mori P. *"A Prototype for Enforcing Usage Control Policies Based on XACML"*. In: Fischer-Hübner S., Katsikas S., Quirchmayr G. (eds) Trust, Privacy and Security in Digital Business. TrustBus 2012. Lecture Notes in Computer Science, vol 7449. Springer, Berlin, Heidelberg, pp. 79-92, 2012.

[2] Cuppens F., Cuppens-Boulahia N., Sans T., Miège A. *"A Formal Approach to Specify and Deploy a Network Security Policy"*. In: Dimitrakos T., Martinelli F. (eds) Formal Aspects in Security and Trust. IFIP WCC TC1 2004. IFIP International Federation for Information Processing, vol 173. Springer, Boston, MA, pp. 203–218, 2005.

[3] A. A. Hassan and W. M. Bahgat, *"A framework for translating a high level security policy into low level security mechanisms,"* 2009 IEEE/ACS International Conference on Computer Systems and Applications, pp. 504-511, 2009.

[4] W. Han and C. Lei, *"A survey on policy languages in network and security management,"* Computer Networks, pp. 477–489, 2012.

[5] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, *"The ponder policy specification language,"* Proceedings of Policy 2001: Workshop on Policies for Distributed Systems and Networks, pp. 18–39, 2001.

[6] N. Dulay, E. Lupu, M. Sloman, and N. Damianou, *"A policy deployment model for the ponder language,"* Proceedings IEEE/IFIP International Symposium on Integrated Network Management, pp. 14–18, 2001.

[7] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, *"Enterprise privacy authorization language (epal 1.2),"* W3C Member Submission, 2003.

[8] M. Backes, B. Pfitzmann, and M. Schunter, *"A toolkit for managing enterprise privacy policies,"* Proceedings of ESORICS, pp. 162–180, 2003.

[9] *"AVISPA, The intermediate format, Automated Validation of Internet,"* http://www.avispa-project.org/delivs/2.3/d2-3.pdf, 2003, accessed: 2019-04-01.

[10] N. Li, Q.Wang,W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin, *"Access control policy combining: theory meets practice,"* The ACM Symposium on Access Control Models and Technologies (SACMAT 2009), p. 135–144, 2009.

[11] Sun, *"Sun's XACML open source implementation,"* http://www.oasis-open.org/committees/xacml/, 2011, accessed: 2019-04-01.

[12] Q. Ni and E. Bertino, *"An extensible functional language for access control"* The ACM Symposium on Access Control Models and Technologies (SACMAT 2011), 2011.

[13] N. Stakhanova, S. Basu, and J. Wong, *"A cost-sensitive model for preemptive intrusion response systems."* Proceedings of the 21st international conference on advanced networking and applications. AINA' 07. IEEE Computer Society, p. 428–435, 2007.

[14] C. Carver, J. Hill, and U. Pooch, *"Limiting uncertainty in intrusion response."* Proceedings of the 2001 IEEE workshop on information assurance and security, 2001.

[15] C. Mu and Y. Li, *"An intrusion response decision-making model based on hierarchical task network planning,"* Expert Syst Appl, 37, p. 2465–2472, 2010.

[16] D. Fernández, F. J. Ruiz, L. Bellido, E. Pastor, O. Walid, and V. Mateos, *"Enhancing learning experience in computer networking through a virtualization based laboratory model,"* International Journal of Engineering Education, vol. 32, no. 6, pp. 2569–2584, 2016.

CyberSPL: Plataforma para la verificación del cumplimiento de políticas de ciberseguridad en configuraciones de sistemas usando modelos de caracterısticas

A. J. Varela-Vaca, Rafael M. Gasca, Rafael Ceballos, y Pedro Bernáldez Torres
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla, Spain
{ajvarela, gasca, ceball, pedbertor}@.us.es

***Resumen*—Los ataques de ciberseguridad se han convertido en un factor muy relevante que pueden contravenir el cumplimiento de las políticas de ciberseguridad de las empresas y organizaciones. Dichos ataques pueden estar provocados en gran medida por una ausencia de configuraciones de seguridad o de valores por defecto en la configuración de productos y sistemas. La complejidad en la configuración de productos y sistemas es un reto en la industria del software. En este artículo proponemos una plataforma, Cybersecurity Software Product Line (CyberSPL), basado en la metodología de diseño de líneas de productos de tal manera que a través de la definición de modelos de características podamos agrupar patrones de configuraciones de aplicaciones y sistemas relacionados con la ciberseguridad. Mediante el análisis automatizado de estos modelos permitiríamos la diagnosis de los posibles problemas en las configuraciones de seguridad y por tanto evitarlos. Como soporte para dicha plataforma se ha implementado una solución multiusario y multiplataforma que permite definir un catálogo de modelos de características público o privado. Además se han integrado mecanismos para determinar todas las configuraciones de un modelo, detectar si una configuración es correcta o no, además de diagnosticar las causas de fallos dada una configuración determinada. Para validar la propuesta se usará un escenario real donde se plantea la configuración de un canal seguro de transmisión mediante el protocolo SSL/TLS, aplicado a un servidor de aplicaciones. En dicho escenario se analizaran dos modelos de características, se validarán diferentes configuraciones, y se diagnosticarán varias configuraciones con problemas.**

***Index Terms*—configuraciones de seguridad, configurabilidad, cumplimiento, políticas de seguridad, modelos de características, automatización**

Tipo de contribución: *Investigación original*

I. Introducción

El notable aumento de ataques en ciberseguridad ha conducido a los investigadores a utilizar técnicas más eficaces para mitigar el impacto de dichos ataques. En muchos casos, estos ataques pueden estar provocados por una ausencia de configuraciones de seguridad, o también por unos valores por defecto en la configuración de productos y sistemas que no están de acuerdo con las políticas de ciberseguridad establecidas. La complejidad en la configuración de productos y sistemas es un reto en la industria del software. Un ejemplo de herramienta de configuración es el KConfig [10] donde los desarrolladores pueden seleccionar entre más de 12.000 opciones de configuración del Kernel de Linux. Una configuración errónea o inadecuada puede desencadenar en problemas de seguridad como por ejemplo ataques por debilidades de la propia configuración.

Entre las técnicas que se utilizan para detectar ataques de ciberseguridad podemos destacar las relacionadas con líneas de productos (Software Product Lines) [6], más concretamente, el uso de técnicas de análisis de variabilidad o análisis de características (Feature-oriented Domain Analysis) [7]. Otros trabajos relacionados con la ciberseguridad son la extracción y selección de características de ficheros de logs [1] para detectar amenazas y ataques de ciberseguridad mediante machine learning, o la verificación de seguridad de las configuraciones de aplicaciones móviles mediante la técnica de model checking parcial [2], y la aproximación orientada a características para la construcción modular de árboles de fallos, que posibilita la reutilización de las estructuras de caminos de propagación de fallos [3].

En nuestro caso, dado los beneficios probados que han tenido las técnicas orientadas a modelos de características en el desarrollo de líneas de productos software, en este artículo proponemos su uso para favorecer el cumplimiento de políticas de ciberseguridad, comprobando las adecuadas configuraciones de los sistemas y productos que la gestionan. También la podríamos considerar muy convenientes como una aproximación de diseño de los productos y sistemas de ciberseguridad basada en modelos (Model-based design) de acuerdo a los requisitos de ciberseguridad exigidos en las políticas de diseño seguro de los mismos.

Los modelos de características (feature models) y las características (features) [7] son el principal concepto de la descomposición funcional de la aproximación de líneas de producto, por tanto, podemos considerar estos modelos para representar los parámetros de configuración de acuerdo a las políticas establecidas. Además, estos modelos permiten expresar restricciones y atributos entre las características de las configuraciones, tal como se ha realizado en un trabajo previo [4]. Permitiendo una mayor expresividad de las dependencias y relaciones entre las diferentes características de las configuraciones de sistemas y productos relacionados con la ciberseguridad.

Una vez las configuraciones correctas se han hecho explíci-

tas en un modelo de características, podemos aplicar técnicas automáticas de verificación formal [8]. Estos mecanismos de verificación formal nos ayudarán a determinar el cumplimiento o no de las políticas o diseños de productos (*Detección de fallos en configuración*) y también identificar los posibles incumplimientos o configuraciones de los sistemas y productos (*Diagnosis de fallos de configuración*) que no cumplen las políticas.

También debemos tener en cuenta que la naturaleza de los modelos de ciberseguridad son altamente dependientes del contexto [11]. Por tanto, los modelos de características deberán ser adaptados en función de los objetivos y los contextos a aplicar para las políticas de cumplimiento [12].

Derivado de todo esto, en este artículo se presenta una propuesta de plataforma que cubre los siguientes objetivos:

- **OBJ1.** Facilitar la definición por parte de los usuarios de un catálogo de políticas de ciberseguridad a través de modelos de características asociados a las diferentes contextos de productos y sistemas relacionados con la ciberseguridad.
- **OBJ2.** Automatizar la derivación de propiedades de los modelos de características. Por ejemplo una propiedad puede ser comprobar la corrección del modelo, es decir, indicar si a partir del modelo se puede extraer algún producto. Otro ejemplo podría ser la extracción de todas las configuraciones de productos o sistemas admitidos por la política de ciberseguridad.
- **OBJ3.** Automatizar la validación del cumplimiento de políticas de ciberseguridad a través de la descripción de características de los diferentes contextos de ciberseguridad que se disponen.
- **OBJ4.** Automatizar el diagnóstico de configuraciones para determinar la causa o causas del incumplimiento de políticas de ciberseguridad a través de la identificación de los fallos en las configuraciones establecidas.
- **OBJ5.** Validación de la propuesta mediante casos de uso complejos. Como por ejemplo la configuración de mecanismos de ciberseguridad de un servidor de aplicaciones web.

El artículo se ha dividido en varias secciones de acuerdo con todo lo anteriormente expuesto. En la sección II se hace una pequeña introducción a los modelos de características y del análisis de estos. En la sección III se presenta nuestra propuesta, incluyendo la arquitectura seguida, el flujo de trabajo establecido, y una descripción del funcionamiento en un escenario real. En la sección IV se expondrá un caso de uso, y se detallarán los resultados obtenidos de la experimentación con dicho caso. En la sección V daremos una perspectiva general de trabajos relacionados. Se finalizará con las conclusiones y los trabajos futuros.

II. Modelos de características

Los modelos de características son el método más común para el análisis de líneas de productos software. Un modelo de características generalmente se representa gráficamente mediante diagramas que definen características y sus relaciones, tal como se puede ver en el ejemplo de la Fig. 1. Este modelo de características puede representar todas las configuraciones de un producto o sistema que cumplan una determinada

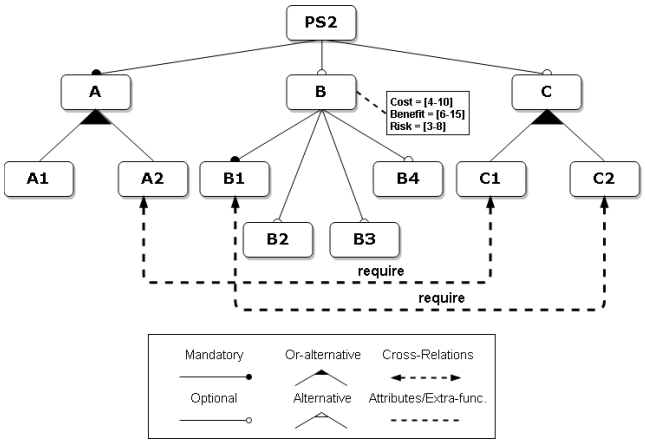


Figura 1: Modelo de características de ejemplo.

política de ciberseguridad. En este ejemplo podemos ver como la características $PS2$ y A siempre deberán aparecer en nuestros productos o sistemas, ya que el elemento raíz siempre debe aparecer y la característica A es obligatoria siempre que aparece $PS2$. Por otro lado, B y C son opcionales, es decir, podrán o no aparecer en las configuraciones. También podemos expresar opcionalidad, como por ejemplo, si está la característica A en nuestras configuraciones podrán aparecer $A1$ ó $A2$ ó ambas. Existen otras dependencias que expresan restricciones cruzadas entre características, por ejemplo, si aparece la característica $A2$ implica que debe aparecer también la característica $C1$.

Además estos modelos pueden ser extendidos con propiedades y funciones extra que aportan información adicional al modelo. Por ejemplo, como se puede observar en Fig. 1, el modelo tiene asociado ciertos atributos a la característica B, que son el $Cost$, $Benefit$, y $Risk$. Estos atributos se han definido de tipo entero con un dominio como el que se indica en la imagen. Esto quiere decir que si aparece la característica B en alguna configuración, dicha característica puede tener asociado estos atributos con algún valor de entre los que se indican en el modelo.

Estos modelos son la base del análisis de una línea de productos e intentan representar el espacio de posibles soluciones. Con estos modelos podemos inferir cierta información, como por ejemplo, el número total de posibles productos/sistemas válidos, si un producto/sistema concreto es válido o no, o si cierta configuración de productos puede ser válida o no. A modo de ejemplo, a continuación mostramos todas las configuraciones obtenidas del modelo de la Fig. 1:

```
Configuracion1 = "PS2;A;A1;",
Configuracion2 = "PS2;C;C2;A;A1;",
Configuracion3 = "PS2;C;C1;A;A1;",
Configuracion4 = "PS2;C;C1;C2;A;A1;",
Configuracion5 = "PS2;C;C1;A;A2;",
Configuracion6 = "PS2;C;C1;A;A1;A2;",
Configuracion7 = "PS2;C;C1;C2;A;A2;",
Configuracion8 = "PS2;C;C1;C2;A;A1;A2;"
```

Por cada configuración se indican las características seleccionadas, por ejemplo, la $Configuracion1$ se han seleccionado las características $PS1$, A, y $A1$. Esto quiere decir que el producto resultante está compuesto de estas características.

Para el análisis automatizado de estos modelos se suelen

utilizar métodos formales [8] basados en lógica proposicional, lógica descriptiva o programación con restricciones. En la literatura se han definido herramientas como FAMA tool [13] o [4] que hacen transformaciones directas de los modelos de característica a modelos de satisfacción de restricciones (Constraint Satisfaction Problem, CSP), y modelos de optimización con restricciones (Constraint Optimization Problem, COP). También, para estos problemas, si los dominios de las variables son de tipo boolean (verdadero o falso), se pueden aplicar resolutores del tipo SAT Solver que facilitan el modelado y la eficiencia en la resolución de los mismos.

El principal objetivo de los CSP para modelos de características es encontrar la satisfacción del modelo, o calcular el número de productos válidos. Mientras que los COP persiguen la optimización de ciertas funciones. En el código 1 podemos ver un ejemplo de modelo CSP válido para la herramienta ChocoSolver [25] y obtenido para el ejemplo de la Fig. 1.

```
==== VARIABLES ====
B1 [0, 1], A [0, 1], B2 [0, 1], A2 [0, 1], C1 [0, 1]
PS2 [0, 1], B [0, 1], B.cost [0, 10], B.risk [0, 8],
B.benefit [0, 15], B3 [0, 1], A1 [0, 1], B4 [0, 1]
C [0, 1], C2 [0, 1], rel-4_card [1, 2], rel-9_card [1, 2]
S-B1 [0, 1], S-B2 [0, 1], D-A [0, 1], S-A2 [0, 1],
S-C1 [0, 1], S-B3 [0, 1], S-B [0, 1], S-B.cost [0, 1],
S-B.risk [0, 1], S-B.benefit [0, 1], S-PS2 [0, 1],
S-A1 [0, 1], S-B4 [0, 1],S-C [0, 1],S-C2 [0, 1]
==== CONSTRAINTS ====
ifonlyif({PS2[0,1],cst[1],A[0,1],cst[1]})
implies({B[0,1],cst[0],B2[0,1],cst[0]})
ifonlyif({S-C1[0,1],cst[1],C1[0,1],cst[1]})
ifthenelse({C[0,1],cst[0],C1[0,1]C2[0,1],rel-9_card[1,2],
   C1[0,1]C2[0,1],cst[0]})
ifthenelse({B[0,1],cst[1],B.risk[0,8],cst[1090519040],
   B.risk[0,8],cst[1077936128],B.risk[0,8],cst[0]})
ifonlyif({S-B.benefit[0,1],cst[1],B.benefit[0,15],cst[0]})
ifonlyif({S-B1[0,1],cst[1],B1[0,1],cst[1]})
ifonlyif({S-C2[0,1],cst[1],C2[0,1],cst[1]})
...
```

Código 1: Trozo de código del modelo de CSP de ChocoSolver.

En Tab. I presentamos los datos más relevantes del modelo de ejemplo de la Fig. 1. Se indica el número de características y la cantidad de relaciones de cada tipo. También aparece el resultado de aplicar dos operaciones sobre el modelo: la comprobación de si el modelo es válido (símbolo ✓), y el cálculo de todas las configuraciones posibles del modelo.

Tabla I: Datos extraído del modelo de ejemplo.

Numero de características	12
Mandatory	2
Optional	5
OR	2
XOR	0
Attributes	1
Cross-Relations	2
Válido	✓
Número de configuraciones	8

III. CyberSPL: Cybersecurity Software Product Line

CyberSPL está enfocado a la verificación del cumplimiento de las políticas de ciberseguridad. El flujo de trabajo de esta plataforma se basa en la ejecución del proceso de negocio que se describe en la Fig. 2.

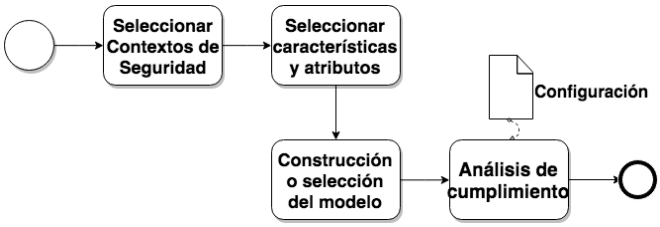


Figura 2: Proceso de verificación del cumplimiento de las políticas.

Este proceso director de CyberSPL nos permitirá evaluar los diferentes contextos de ciberseguridad de una organización. En primer lugar, se *seleccionará el contexto de ciberseguridad* a analizar y se *seleccionarán las características y atributos* de las mismas que son permitidos según la política de ciberseguridad. A continuación, a partir de ellos se construirán o se seleccionarán un catálogo de modelos de características y se procederá a realizar los correspondientes análisis de los productos o servicios que se pretenden verificar de acuerdo a la política establecida, esto se representa mediante la tarea *análisis de cumplimiento*. Es muy importante en la ejecución de este proceso de análisis que los modelos de características que se utilicen estén totalmente actualizados con los últimos detalles de la política de ciberseguridad establecida por la organización.

La *selección de los contextos de seguridad* y la *selección de características y atributos* es algo que tendrá que establecerse en la política de ciberseguridad y será necesario un trabajo de análisis [12] previo a la definición de los modelos de características. Por ejemplo, podemos seleccionar un contexto de despliegue de servicios de una organización que estará basado en un servidor de aplicaciones. El servidor podría ser *Apache Tomcat*, y nuestra política de seguridad indicaría que las comunicaciones hacia el servidor deben ser totalmente seguras a través de la configuración de canales basados en Secure Socket Layer (SSL) [14] y/o Transport Layer Security (TLS) [15][16].

Como se ha indicado en el proceso tendremos que, o bien construir un modelo si no existe, o seleccionar un modelo de características de los existentes en un catálogo. CyberSPL se ha provisto de mecanismos para tanto la construcción de modelos de características nuevos en formato FAMA [13], como para la selección de modelos de un catálogo de modelos de características, de tal manera que el esfuerzo de desarrollo y análisis se facilite a los gestores. En un trabajo previo [4] fue definido un catálogo de modelos de características para la configuración de controles de seguridad para un motor de procesos. Actualmente, CyberSPL no sólo nos proporciona mecanismos de modelado para la construcción de modelos de características, sino que también facilita la persistencia y actualización de los mismos mediante un catálogo de modelos. Dicho catalogo se puede compartir, permitiendo así reutilizar el conocimiento recogido en los modelos.

En la Fig. 3 se muestra el flujo de trabajo de CyberSPL usando el catálogo de modelos. En este caso, seleccionaríamos un modelo dependiendo del contexto y de la política indicada, y en un paso posterior se podría reajustar el modelo con las características y atributos adecuados. Este modelo reajustado

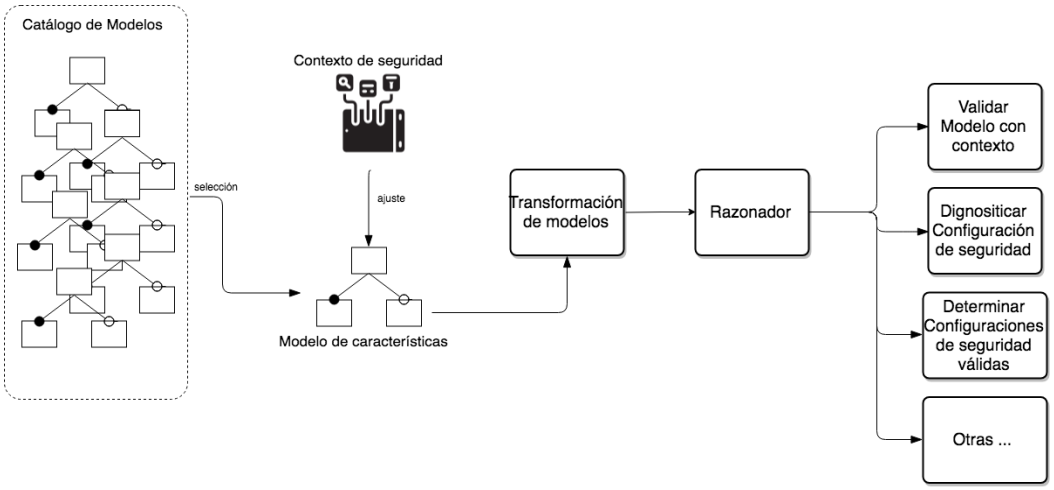


Figura 3: Flujo de trabajo de CyberSPL con el catálogo de modelos.

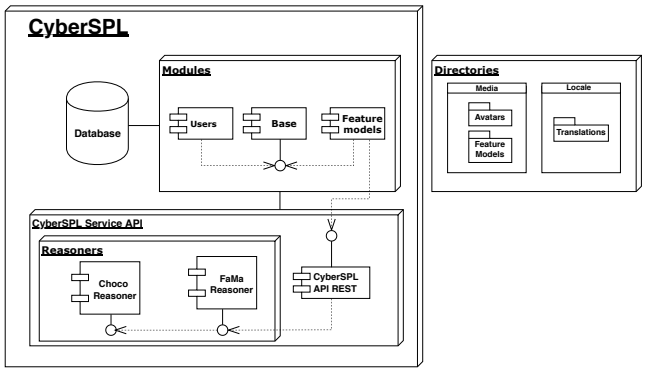


Figura 4: Arquitectura de CyberSPL.

pasaría por una transformación en la que se generaría un modelo formal basado en restricciones, tal como se indicó en la sección II. Usando un razonador ajustado al modelo obtenido podemos aplicar el *análisis de cumplimiento* que necesitemos. Como por ejemplo validar el modelo según el contexto, diagnosticar una configuración de seguridad determinada, determinar todas las configuraciones válidas, u otras operaciones.

CyberSPL ha sido implementado siguiendo la arquitectura de la Fig. 4. Dicha implementación es modular donde se integran un módulo de gestión de usuarios, un módulo base de gestión de la plataforma, y un módulo para gestión de modelos de características (feature models). CyberSPL se ha integrado con un API REST que le permiten consumir FAMA tools junto con el razonador de ChocoSolver, para realizar el análisis y razonamiento de los modelos que se vayan creando.

CyberSPL está pensado para ser una solución web multiplataforma y multiusuario. Es decir, cualquier usuario puede registrarse desde un navegador web y utilizarla. Cualquier usuario puede ver los modelos públicos disponibles en CyberSPL y de manera interna a través de su perfil puede definir su propio catálogo de modelos tal y como se muestra en la Fig. 5. En la misma figura podemos ver como es la edición de un modelo de CyberSPL. Los modelos que cada usuario genere se pueden indicar como privados, para uso exclusivo del usuario, o como públicos, para ser consultados y usados por cualquier otro usuario.

Sobre cada modelo se puede actuar editándolo, borrándolo, ó realizando un análisis. La sección de análisis de cada modelo provee al usuario de tres opciones principales como se puede observar en la Fig. 6 (aparece difuminado en el fondo): (i) validar modelo, (ii) diagnosticar una configuración que responde a una política de ciberseguridad establecida, y (iii) obtener todas las configuraciones válidas para una política de ciberseguridad establecida. En la Fig. 6 se ha seleccionado la opción de diagnosticar una configuración (resaltado con un modal), donde CyberSPL da al gestor de ciberseguridad la oportunidad de especificar cierta configuración para ser cruzada con el modelo, que en este caso se corresponde con el modelo que ya fue presentado previamente en la Fig. 1. En la figura también se puede observar que en dicha sección tendremos disponible una consola donde se irán mostrando al usuario los resultados de las diferentes operaciones realizadas sobre el modelo. En la Fig. 6 se observa el listado de las configuraciones obtenidas para el modelo. Además CyberSPL da la opción de guardar dichas configuraciones de manera externa en formato JSON, usando la opción *Exportar configuraciones* que se encuentra debajo de la opción de obtener todas las configuraciones.

IV. Caso de uso y datos de experimentación

Para poder analizar las ventajas de aplicar y usar CyberSPL vamos a utilizar el caso de uso planteado en la Fig. 7. Dicho caso de uso representa un contexto de ciberseguridad real, donde una organización tiene un conjunto de usuarios o empleados que usan los servicios de la red corporativa de la organización, a través de un servidor de aplicaciones *Apache* que actúa de proxy. En la política de ciberseguridad se establece que las conexiones a la red corporativa deben hacerse de manera segura. Por tanto, en dicho servidor se pueden establecer ciertas configuraciones de seguridad para asegurar el canal a través del uso del protocolo *SSL/TLS*. En la Fig. 7 podemos observar una parte de la configuración de un servidor *Apache*, donde se han establecido ciertas características de la comunicación segura tales como versiones del protocolo o incluso tamaños de claves permitidos.

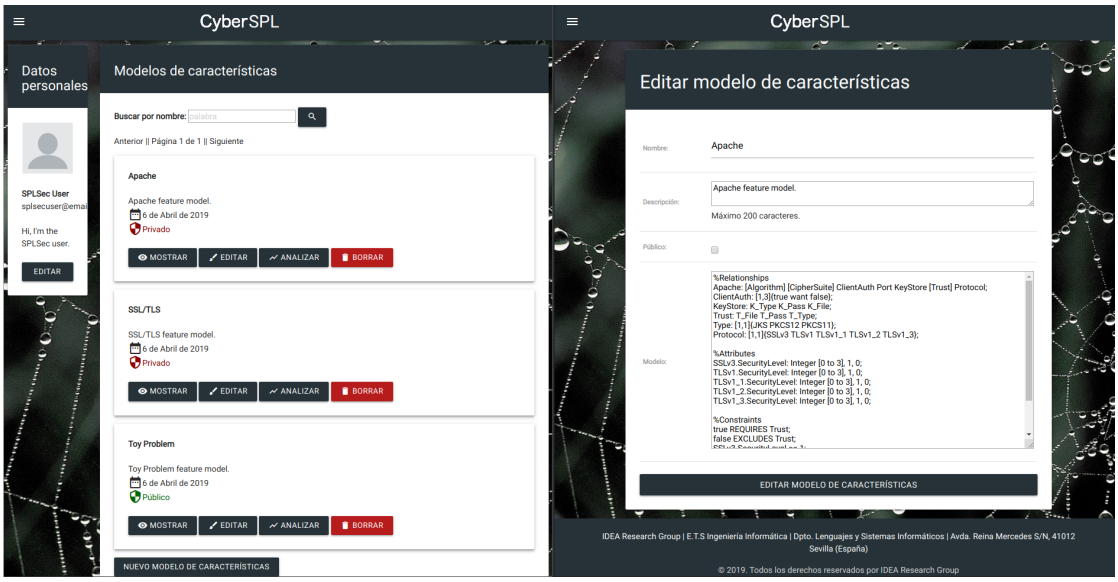


Figura 5: Catálogo de modelos de un usuario.

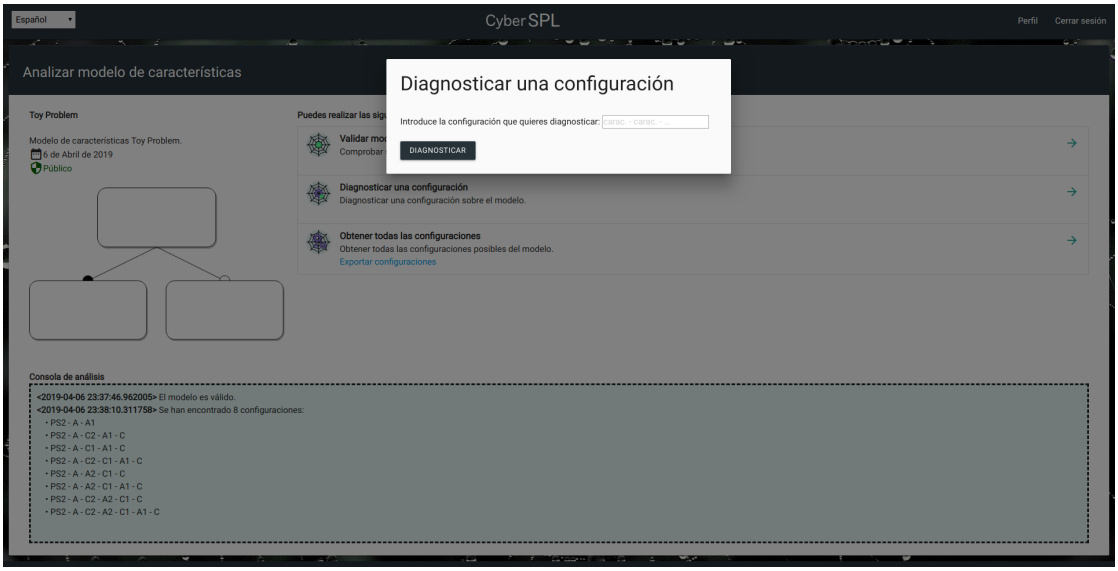


Figura 6: Sección de análisis de modelos.

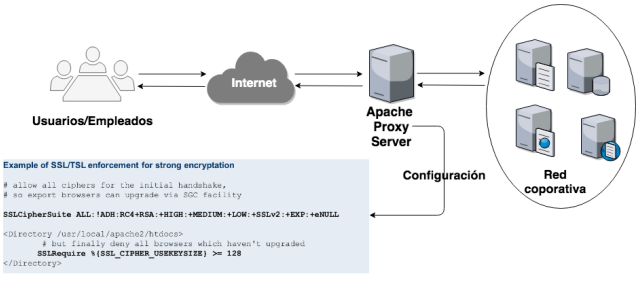


Figura 7: Caso de uso.

IV-A. Descripción de contexto y modelos de característica

Para analizar el contexto de ciberseguridad vamos a usar dos modelos características: (1) modelo de *Apache*, y (2) modelo de *SSL/TLS*. Ambos modelos son una evolución actualizada de los presentados en [4], donde se han eliminado las características consideradas inseguras y se han añadido otros parámetros como los tamaños de claves en el caso del modelo de SSL/TLS.

El primer modelo que usaremos es el presentado en la Fig. 8, que muestra características configurables de seguridad de un servidor *Apache* con respecto a *SSL/TLS*. El protocolo *SSL/TLS* se basa en el handshake cuya secuencia de pasos son: (1) Negociación del *Cipher Suite* a usar durante la transferencia, y generación e intercambio de un número aleatorio (master key); (2) Establecer e intercambiar un identificador de sesión entre cliente y servidor; (3) Autenticar al servidor frente al cliente; (4) Autenticar al cliente frente al servidor. Este handshake se ha simplificado en la nueva versión *TLSv1.3*. SSL/TL permite autenticar tanto cliente como servidor, y la comunicación anónima. La autenticación se hace a través de firmas digitales como certificados o claves. En el caso

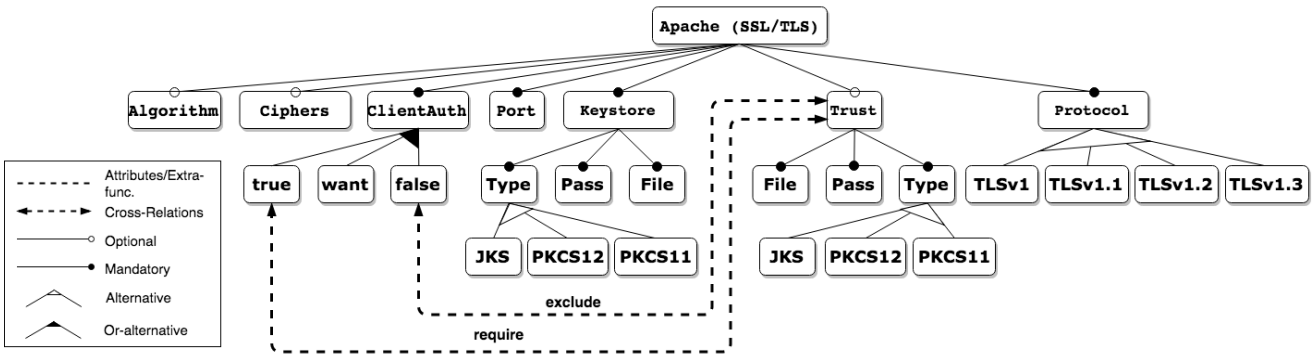


Figura 8: Modelo de características de Apache.

de certificados estos siempre se cruzan con autoridades de certificación (CA). Por otra parte, estos protocolos también permiten la autenticación anónima usando intercambio de claves *Diffie-Hellman* en los protocolos *SSLv3* y *TLSv1.0*.

En el modelo podemos observar como hay ciertos parámetros como qué versiones del protocolo (*Protocol*) se pueden configurar. En este caso hemos descartado aquellas versiones de protocolo que ya están desaconsejadas como es el caso de SSLv3. Hemos querido mantener versiones débiles como *TLSv1* ó *TLSv1.1*, porque aún se siguen usando y porque su cese está programado para el año 2020, mientras tanto hay proveedores y clientes que soporten dichos protocolos. Por otro lado, si usamos autenticación en cliente (*ClientAuth*) esto va a requerir que usemos cierta infraestructura como almacenes (*Keystore* y *Trust*) y certificados. Además existirá la posibilidad de establecer para ciertos protocolos las *Ciphers* aceptadas. La configuración de los *Ciphers* nos llevo a realizar un análisis en profundidad de todas las características de los cipher suites seguras.

El segundo modelo que usaremos es el mostrado en la Fig. 9, que corresponde con un modelo completo de *SSL/TLS* donde se presentan todas las características para la configuración de los cipher suites para cualquier producto o sistema basado en las versiones de *TLS1.2,* y *TLS1.3*. En principio todas las versiones anteriores se han considerado inseguras y por tanto se han descartado. Este modelo se puede enlazar con el de *Apache* de la Fig. 8 por la característica de *Ciphers*.

Queremos destacar que el modelo de *SSL/TLS* es simplificado, ya que no se han representado las restricciones cruzadas. Representar todas estas restricciones implicaría tener un modelo que sería visualmente muy complejo de interpretar. Sin embargo, se han dejado anotados en la caja de texto *Cross Relations* de la Fig. 9. Por ejemplo, *AES_1238_CBC* no se recomienda para la versión del protocolo *TSLv1.3*, por lo que se ha añadido una relación cruzada de exclusión. Por otro lado, el uso de *RSA* implica que el uso de tamaño de claves (*KeySize*) debe ser de *2048*.

IV-B. Análisis de los modelos de característica y diagnosis de configuraciones

A continuación, vamos a realizar un análisis de los modelos anteriores que describen nuestro contexto y se adecuan a una política de ciberseguridad. El análisis aplicado se centra en describir los modelos, validarlos, y ver cuántas son las configuraciones que pueden estar disponibles. Los resultados se pueden ver en Tab. II.

Tabla II: Datos extraído del modelo de ejemplo.

Modelo de característica	*Apache*	*SSL/TLS*
Número de características	27	48
Mandatory	10	8
Optional	3	0
OR	1	1
XOR	3	9
Cross-Relations	2	12
Válido	✓	✓
Número de configuraciones	96	1482

Podemos destacar que ambos modelos son válidos, por lo que de ambos modelos obtendríamos al menos una configuración válida. Con respecto a las configuraciones, podemos decir que el número de configuraciones posibles para *Apache* son 96, mientras que las de *SSL/TLS* son 1482. Estos datos nos dan una perspectiva de la complejidad que plantea la configuración de un sistema como *Apache* y *SSL/TLS*. Aún acotando las configuraciones a pocos parámetros, podemos ver como el dominio del problema sería seleccionar o analizar una configuración de entre miles, lo que clasifica a este problema como inabarcable si se realizara de manera manual por un humano. La complejidad aumentaría si nos planteáramos combinar ambos sistemas, ya que el número de configuraciones se multiplicaría significativamente, dando lugar a una explosión combinatoria en el número de posibles configuraciones, que haría el problema aún más intratable.

Uno de los puntos fuertes de nuestra propuesta, CyberSPL, es la posibilidad de hacer diagnosis de configuraciones. La diagnosis va más allá del mero hecho de determinar si una configuración es válida o no. La diagnosis pretende dar respuesta a el porqué una configuración es no válida de acuerdo a la política de ciberseguridad establecida. A continuación, vamos a presentar varios ejemplos de configuraciones de sistemas que necesitan ser verificadas, es decir, comprobar si están de acuerdo con la política representada en los modelos de características. En el caso de que las configuraciones no sean válidas, aplicaremos CyberSPL para establecer el diagnóstico, es decir, determinar cuáles son los posibles fallos en las mismas.

Como podemos observar en ambos casos en Tab. III y Tab. IV, se han probado cuatro configuraciones de las cuáles dos fueron no válidas, en cuyo caso se ha proporcionado

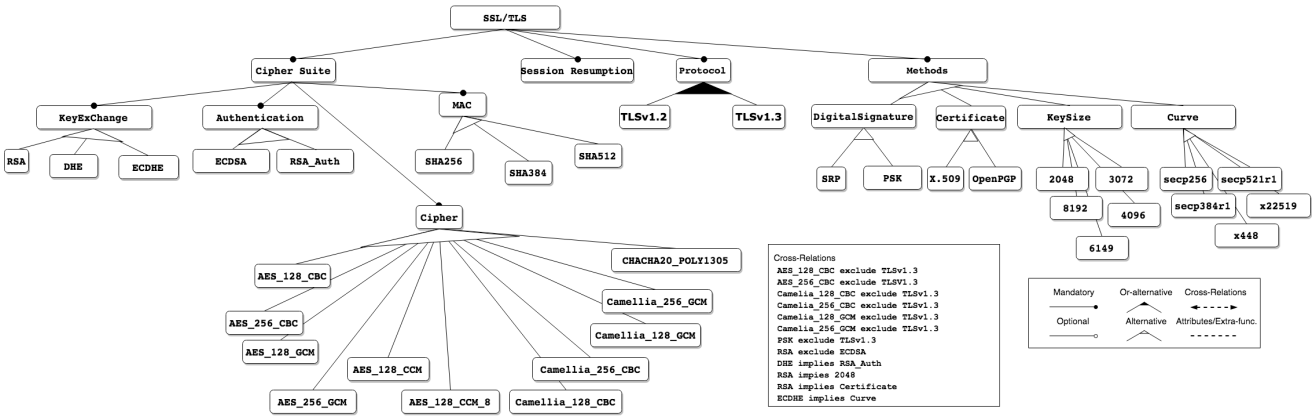


Figura 9: Modelo de características de SSL/TLS.

Tabla III: Diagnóstico de configuraciones en *Apache*

Configuración	Resultado	Diagnosis
Apache, Protocol, TLSv1.1, KeyStore, File, Pass, Type ClientAuth, false, Port	**Válida**	-
Apache, Protocol, TLSv1.2, KeyStore, File, Pass, Type Ciphers, Algorithm, ClientAuth, want, Port	**Válida**	-
Apache, Protocol, KeyStore, File, Pass, Type ClientAuth, false, Port, Algorithm, Ciphers	**No Válida**	**Seleccionar**: TLSv1, TLSv1.2, TLSv1.1, TLSv1.3
Apache, Protocol, TLSv1.2, KeyStore, File, Pass, Type, ClientAuth, Port, Algorithm, Ciphers	**No Válida**	**Seleccionar**: want, false

Tabla IV: Diagnóstico de configuraciones en *SSL/TLS*

Configuración	Resultado	Diagnosis
SSL/TLS, Protocol, TLSv1.2, KeyExchange, DHE, CipherSuite, Cipher, AES_128_GCM, MAC, SHA256, Authentication, ECDSA, Methods, KeySize, 3072, SessionResumption	**Válida**	-
SSL/TLS, Protocol, TLSv1.2, CipherSuite, Cipher, AES_128_CCM, Authentication, ECDSA, KeyExchange, DHE, MAC, SHA512, Methods, KeySize, 3072, SessionResumption	**Válida**	,
SSL/TLSS, Protocol, TLSv1.2, CipherSuite, Cipher, AES_128_CCM, Authentication, KeyExchange, ECDHE, MAC, SHA512, Methods, KeySize, 3072, SessionResumption	**No Válida**	**Seleccionar**: RSA_Auth, DHE **Deseleccionar**: ECDHE
SSL/TLS, Protocol, CipherSuite, MAC, SHA384, Authentication, ECDSA, Cipher, AES_256_GCM, KeyExchange, DHE, Methods, KeySize , 4096, SessionResumption	**No Válida**	**Seleccionar**: TLSv1.2, TLSv1.3

el diagnóstico. Para el caso de las configuraciones inválidas de *Apache* (Tab. III) podemos ver que la primera configuración fallida es porque no se ha especificado la versión del protocolo, aquí el diagnóstico sugiere la selección de una versión del protocolo concreta. Mientras que en el segundo caso se ha configurado todo pero no se ha indicado qué tipo de *ClientAuth*. En este caso no se puede seleccionar *true* porque implicaría la selección de la característica de *Trust*, por lo que el diagnóstico, nos sugiere que debemos seleccionar entre las opciones de *want* ó *false*. Para el caso de *SSL/TLS* (Tab. IV) tenemos también dos configuraciones inválidas. Para el diagnóstico de la primera configuración inválida, se observa que para la característica de *Authentication* no se ha seleccionado ningún mecanismo, que podría ser *ECDSA* o *RSA_Auth*. Por otro lado se ha seleccionado *ECDHE* como característica de *KeyExchange* que requeriría una curva elíptica (*Curve*) pero en nuestra configuración nos dice que estamos usando claves con *KeySize* de 3072. Por todo esto, la diagnosis más factible nos indica que debemos no seleccionar *ECDHE* y que podríamos seleccionar las características *RSA_Auth* y *DHE*, ya que seleccionando estas dos características se cumpliría con el requisito de una clave (*KeySize*) y un tamaño determinado (3072).

V. Estado del arte

Tras hacer una revisión bibliográfica del estado del arte al respecto, no se han encontrado referencia alguna sobre el uso de modelos de características aplicados en el ámbito del cumplimiento de políticas de ciberseguridad. Por tanto, a la vista de ello podemos decir que este trabajo es pionero en tratar esta problemática aplicando técnicas basadas en modelos de características. Dado que se integran dos áreas de investigación hasta ahora no integrados, los trabajos relacionados los hemos dividido en estas dos principales perspectivas que aborda el artículo:

Análisis y diagnosis sobre modelos de características

El análisis automatizado de modelos de características es algo conocido y aplicado desde hace décadas en el área de la líneas de productos software [8][22][24]. El análisis automatizado persigue extraer o inferir ciertas propiedades de los modelos. En ciertos trabajos el análisis se centraba en determinar, analizar, o diagnosticar errores en modelos de características ya sea en diseño [21] o en etapas de reconfiguración [23].

Existen otras aproximaciones donde el análisis de modelos de características se aplica a otros ámbitos. En [17], se propone una extensión de un método basado en objetivos (KAOS) para generar modelos de requisitos adaptables a partir

de modelos de variabilidad. En [19], se utilizan modelos de características para analizar los requisitos de variabilidad y, en consecuencia, transformar estos modelo de característica para generar un modelo arquitectónico. En [20], se utiliza el análisis de modelos de características para proporcionar sistemas auto-adaptables mediante la determinación dinámica de las mejores variantes adaptadas a los requisitos específicos de QoS.

Análisis o aplicación de mecanismos de verificación de configuraciones en el ámbito de la ciberseguridad

En este apartado, podemos considerar el trabajo [18], donde se propone un enfoque que facilita el desarrollo de líneas de productos de software seguro (SPLs) y sus productos derivados.

También debemos reseñar en el contexto de verificación de configuraciones de seguridad, que dado que los datos de configuración de los sistemas Internet-of-Things (IoT) son no estructurados, las técnicas tradicionales no pueden tratar de forma automática las configuraciones específicas del IoT tales como la seguridad. Es por ello que se ha propuesto la plataforma *IoTChecker* [9] para el análisis de seguridad en productos IoT. Finalmente, también debemos reseñar para el cumplimiento de políticas Bring your own device (BYOD) se ha propuesto en [2] una técnica para la verificación automática de seguridad en aplicaciones móviles.

VI. Conclusiones y trabajos futuros

En este artículo se ha presentando la plataforma CyberSPL y una herramienta para su implementación que nos permite la verificación de cumplimiento de políticas de ciberseguridad a través del análisis de configuraciones de productos, aplicaciones y servicios que participan en ella. Los resultados obtenidos con el uso de dicha propuesta nos llevan a pensar que puede representar un avance importante para facilitar el trabajo en la gestión automatizada del cumplimiento de políticas por parte de los gestores de la ciberseguridad o en el desarrollo operacional (DevOps), donde asegurar y comprobar que una configuración es acorde con la política supondría un notable incremento del alineamiento entre la capa de desarrollo y la operacional.

Como trabajos futuros se han identificado dos líneas principales: (1) la necesidad de realizar la actualización automática de los modelos de características de acuerdo con los avances tecnológicos o vulnerabilidades que se produzcan a lo largo del tiempo; y (2) el mantenimiento de un histórico de la evolución de los modelos de características para cada uno de los contextos de tal manera que nos permita alertar de forma temprana a los usuarios cuando se produzcan cambios en las políticas de ciberseguridad establecidas.

Agradecimientos

Este artículo ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología de España a través del proyecto ECLIPSE (RTI2018-094283-B-C33), de la Junta de Andalucía vía los proyectos PIRAMIDE y METAMORFOSIS, los fondos FEDER. Los autores quieren agradecer a la Cátedra de Telefónica "Inteligencia en la Red" de la Universidad de Sevilla por su apoyo en el desarrollo de este trabajo, y a José A. Galindo y David Benavides por el soporte de la herramienta FAMA.

Referencias

[1] D. Sisiaridis and O. Markowitch, "Automating Feature Extraction and Feature Selection in Big Data Security Analytics," in Artificial Intelligence and Soft Computing, Springer International Publishing, 2018, pp. 423–432.
[2] G. Costa, A. Merlo, L. Verderame, and A. Armando, "Automatic security verification of mobile app configurations," Future Generation Computer Systems, vol. 80, pp. 519–536, Mar. 2018.
[3] B. Behringer, M. Lehser, and S. Rothkugel, "Towards Feature-Oriented Fault Tree Analysis," in 2014 IEEE 38th International Computer Software and Applications Conference Workshops, 2014.
[4] A. J. Varela-Vaca and R. M. Gasca, "Towards the automatic and optimal selection of risk treatments for business processes using a constraint programming approach," Information and Software Technology, vol. 55, no. 11, pp. 1948–1973, Nov. 2013.
[5] M. Schumacher, Security Engineering with Patterns. Springer Berlin Heidelberg, 2003.
[6] Kang. Kyo, Cohen. Sholom, Hess. James, Novak. William, and Peterson. A., "Feature-Oriented Domain Analysis (FODA) Feasibility Study,"Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-90-TR-021 , 1990. http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=11231
[7] D. Batory, "Feature Models, Grammars, and Propositional Formulas," in Software Product Lines, Springer Berlin Heidelberg, 2005, pp. 7–20.
[8] D. Benavides, S. Segura, and A. Ruiz-Cortés, "Automated analysis of feature models 20 years later: A literature review," Information Systems, vol. 35, no. 6, pp. 615–636, Sep. 2010.
[9] M. Mohsin, Z. Anwar, F. Zaman, and E. Al-Shaer, "IoTChecker: A data-driven framework for security analytics of Internet of Things configurations," Computers & Security, vol. 70, pp. 199–223, Sep. 2017.
[10] R. Lotufo, S. She, T. Berger, K. Czarnecki, and A. Wąsowski, .Evolution of the Linux Kernel Variability Model,ïn Software Product Lines: Going Beyond, Springer Berlin Heidelberg, 2010, pp. 136-150.
[11] M. Schumacher, Security Engineering with Patterns. Springer Berlin Heidelberg, 2003.
[12] A. J. Varela-Vaca and R. M. Gasca, "Formalization of security patterns as a means to infer security controls in business processes," Logic Journal of IGPL, vol. 23, no. 1, pp. 57–72, Dec. 2014.
[13] D. Benavides, P. Trinidad, A. Ruiz Cortés, and S Segura. "FaMa". Springer Berlin Heidelberg, 2013, Chapter FaMa, 163–171. DOI:http://dx.doi.org/10.1007/978-3-642-36583-6-11.
[14] K. Hickman, The SSL Protocol. "Netscape Communications Corp.", 1995.
[15] T. Dierks, E. Rescorla, "The TLS Protocol Version 1.2". RFC 5246, 2008.
[16] E. Rescorla, "The TLS Protocol Version 1.3". RFC 8446, 2018.
[17] F. Semmak, C. Gnaho, and R. Laleau, "Extended KAOS Method to Model Variability in Requirements," in Communications in Computer and Information Science, Springer Berlin Heidelberg, 2010, pp. 193–205.
[18] D. Mellado, E. Fernández-Medina, and M. Piattini, "Security requirements engineering framework for software product lines," Information and Software Technology, vol. 52, no. 10, pp. 1094–1117, Oct. 2010.
[19] J. Pérez, M. A. Laguna, Y. C. González-Carvajal, and B. González-Baixauli, "Requirements Variability Support Through MDA™ and Graph Transformation," Electronic Notes in Theoretical Computer Science, vol. 152, pp. 161–173, Mar. 2006.
[20] P. Sawyer, R. Mazo, D. Diaz, C. Salinesi, and D. Hughes, "Using Constraint Programming to Manage Configurations in Self-Adaptive Systems," Computer, vol. 45, no. 10, pp. 56–63, Oct. 2012.
[21] P. Trinidad, D. Benavides, A. Durán, A. Ruiz-Cortés, and M. Toro, "Automated error analysis for the agilization of feature modeling," Journal of Systems and Software, vol. 81, no. 6, pp. 883–896, Jun. 2008.
[22] D. Benavides and J. A. Galindo, "Automated analysis of feature models," in Proceeedings of the 22nd International Conference on Systems and Software Product Line - SPLC '18, 2018.
[23] A. Felfernig et al., "Anytime diagnosis for reconfiguration," Journal of Intelligent Information Systems, vol. 51, no. 1, pp. 161–182, Jan. 2018.
[24] J. A. Galindo, D. Benavides, P. Trinidad, A.-M. Gutiérrez-Fernández, and A. Ruiz-Cortés, "Automated analysis of feature models: Quo vadis?," Computing, Aug. 2018.
[25] C. Prud'homme, J.-G. Fages, X. Lorca, Choco Documentation, http://www.choco-solver.org, 2017.

Modelo Emergente Preventivo para producir software seguro

José Carlos Sancho Núñez
Cátedra ViewNext-UEx
Escuela Politécnica
Av. Universidad s/n - Cáceres
jcsanchon@unex.es

[1]Andrés Caro Lindo
[2]Pablo García Rodríguez
Universidad de Extremadura
Escuela Politécnica
Av. Universidad s/n - Cáceres
[1,2]{andresc, pablogr}@unex.es

José Andrés Félix de Sande
ViewNext S.A.
CENIT Cáceres (PCTEx)
Av. Universidad s/n - Cáceres
jafelix@viewnext.com

Resumen- **La previsión del incremento de ciberataques y de su sofisticación podrían poner en jaque a sistemas e infraestructuras críticas de consumo humano. Por este motivo, se considera necesario introducir nuevos modelos emergentes que desarrollen software de forma segura por defecto. Esta contribución realiza un experimento comercial llevado a cabo en una empresa de desarrollo. Acerca de forma novedosa una comparativa de resultados entre dos escenarios de desarrollo, uno clásico cuyo enfoque de la seguridad es reactivo y otro, emergente y preventivo que aplica la seguridad por defecto durante todas las fases del ciclo de vida del software. La reducción de un 66% de vulnerabilidades y la minimización del impacto temporal en la resolución de los fallos de seguridad encontrados, son las claves que evidencian que la propuesta presentada construye un software más seguro por defecto que el realizado utilizando modelos clásicos.**

***Index Terms*- Modelo Emergente, Software Seguro, Ingeniería del Software, Reducción de Vulnerabilidades, Enfoque Preventivo, Experimento Comercial.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Cada año se incrementa el número de ciberataques, así como su sofisticación, la gravedad de sus consecuencias y su audacia [1]. El escenario se presenta alarmante para la sociedad. Sobre todo, cuando los ataques se dirigen a sistemas críticos que afectan al consumo humano, desconfiguración de infraestructuras críticas o hacia sistemas SCADA encargados de controlar y/o supervisar procesos industriales. También son preocupantes los ataques en campos más genéricos como el sector bancario o el IoT (Internet of Things).

Las posibles consecuencias de estos tipos de ataques son la alteración de parámetros de la composición y depuración del agua, la manipulación de redes eléctricas, la alteración de objetos cotidianos que se encuentran interconectados en hogares y ciudades, así como las graves consecuencias en empresas, bancos u otras instituciones financieras.

La aparición de vulnerabilidades en el software que controla y supervisa estos sistemas ha hecho que la seguridad en todos los procesos de desarrollo se convierta en un gran desafío que no puede pasar desapercibido.

Para evitar que los ataques evolucionen y adapten sus técnicas a un ritmo más rápido que el desarrollo de contramedidas es preciso romper con los modelos clásicos de desarrollo de software, en los que la seguridad adopta un papel claramente reactivo. La búsqueda insistente de *nuevos modelos de software seguro*, capaces de cumplir sus objetivos funcionales y, a la vez, detectar los problemas de seguridad por anticipado se presenta como un gran reto. Queda patente la necesidad de disponer de nuevos modelos de software en los que el producto final sea el resultado de un proceso de desarrollo que integre una funcionalidad precisa unida a una respuesta preventiva que se anticipe a cualquier problema de seguridad. De este modo, Tran y otros [2] avanzan hacia la detección temprana de vulnerabilidades, proponiendo un modelo que detecte y combata ataques zero-day. Y Murtuza y otros [3] analizan las tendencias y estudian patrones de vulnerabilidades en el software. Pese a ello, se trata de planteamientos reactivos que actúan para solventar fallos de seguridad demasiado tarde.

Por contra, en este trabajo se considera que la mejora en los sistemas software pasa por la propuesta de nuevos modelos de software seguro, desarrollados en base a enfoques preventivos. Estos modelos emergentes deben tener en cuenta la seguridad en el proceso de desarrollo del software desde las fases iniciales. En este sentido, hay trabajos que hace tiempo evidencian la necesidad de considerar la seguridad en todas las tareas de las fases del ciclo de vida del software [4]. Y otros como Mellado [5] dan un paso más profundo al aplicar la seguridad en el proceso de ingeniería de requisitos para minimizar ataques maliciosos.

Sin embargo, la realidad es que las empresas desarrolladoras de software generan productos que cumplen los requisitos funcionales olvidando las cuestiones de seguridad. Es en la fase de pruebas donde determinan la calidad del software e identifican y solventan los problemas de seguridad. Este modelo de desarrollo conlleva que los

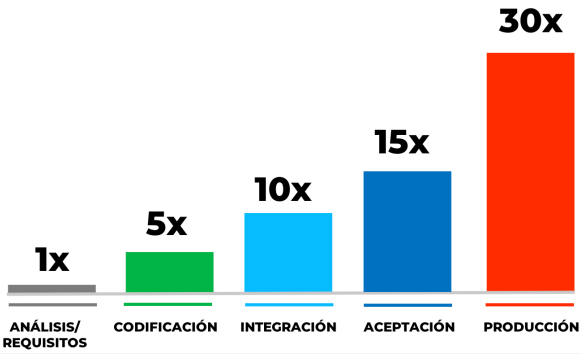


Fig. 1. Coste relativo de resolución de fallos de seguridad en función de la fase del SDLC donde se detecta.

fallos de seguridad de software repercutan a nivel económico, en la reputación corporativa de la empresa e incluso en incumplimientos normativos como protección de datos de clientes y sus graves consecuencias económicas.

En este sentido, es preciso valorar la diferencia entre la utilización de modelos clásicos como es el reactivo, o modelos emergentes como es un modelo preventivo que se anticipe a los posibles fallos de seguridad. En Fig. 1 el Instituto Nacional de Estándares y Tecnología (NIST) cuantifica en [6] que es 30 veces más costoso solucionar una vulnerabilidad durante la fase de post-producción que durante la etapa de diseño, la toma de requisitos y la definición de la arquitectura.

En *Fig. 1* se observa que esperar hasta la etapa de pruebas para detectar y solventar los fallos de seguridad aumenta exponencialmente el coste, por lo que las medidas reactivas ayudan a la detección de vulnerabilidades, pero no a una adecuada protección de los sistemas ni a la minimización de los costes de su resolución.

Además, identificar vulnerabilidades desconociendo su origen ocasiona un rediseño y excesivos cambios en la implementación del software. Por lo que, coincidiendo con Solinas y otros [7] se considera que la seguridad en el software deja de ser opcional para pasar a ser obligatoria.

Con respecto a la estimación del esfuerzo se puede afirmar que el ahorro en el tiempo de ejecución de los proyectos y, por ende, una mejora en la productividad para las empresas de desarrollo se convierte en ganancias económicas. Por este motivo, la estimación de esfuerzo destinado a realizar un desarrollo de software seguro es de vital importancia. Yang y otros [8] establecen un modelo de estimación de esfuerzo para el desarrollo seguro de software de sistemas operativos en China; Sodiya y otros [9] discuten los problemas de conseguir productos de software seguros.

Por lo tanto, se pone de manifiesto la dificultad de desarrollar aplicaciones que cumplan con los criterios seguridad y los principios básicos de la seguridad de la información como la integridad, la confidencialidad y la disponibilidad. Incluso aplicando técnicas de inteligencia artificial para asegurar la información en las empresas como hacen Rehman y Saba [10].

Dando un paso hacia la aplicación de modelos emergentes que generen software que cumpla su funcionalidad y, además, sea seguro, se propone afrontar la seguridad en todas las fases del ciclo de desarrollo. Así, en este artículo se presenta la aplicación experimental hecha en un entorno industrial de un nuevo modelo emergente de Desarrollo de Software Seguro.

Se detallan los aspectos que caracterizan el modelo y los puntos fuertes que lo diferencian y mejoran de otros. Entre ellas, la particularización de una herramienta para la formación en desarrollo seguro, el proceso de trazabilidad de la seguridad y seguimiento de los riesgos y requisitos, o las buenas prácticas de codificación segura que se establecen.

El objetivo de este trabajo se centra en comparar el nuevo modelo emergente y preventivo con respecto a un modelo clásico y reactivo.

II. *BACKGROUND* Y TRABAJOS RELACIONADOS

La mayoría de estudios enfocados en el desarrollo seguro van encaminados a introducir comprobaciones y contramedidas en el ciclo de vida de desarrollo de software (SDLC). Jones [11] incide en que la seguridad debe incorporarse al proceso general del ciclo de vida del desarrollo de sistemas. Karim y otros [12] diseñan una extensión de seguridad al modelo del SDLC.

Las metodologías ágiles también irrumpen en estudios recientes que buscan hacer seguro el proceso ágil. De esta forma Kaur y otros [13] proponen un modelo de espiral con la seguridad aplicada; Othmane y otros [14] integran actividades de seguridad en el proceso ágil de desarrollo de software.

Hay pocos estudios que analicen en profundidad metodologías de desarrollo seguro por defecto. Por este motivo, en paralelo a este trabajo se han estudiado diversos marcos de trabajo que integran la seguridad por defecto, todos ellos propietarios y utilizados en la actualidad por grandes empresas tecnológicas que se dedican a la construcción del software. Estos modelos son: Microsoft Security Development Lifecycle (Microsoft SDL) [15] y Agile Security Development Lifecycle (Microsoft ASDL) [16] en su versión ágil, Oracle Software Security Assurance (OSSA) [17], Comprehensive Lightweight Application Security Process (CLASP) de Open Web Application Security Project (OWASP) [18], Team Software Process Secure (TSP-Secure) [19], Software Assurance Maturity Model (OpenSAMM) [20] de OWASP y Building Security In Maturity Model Framework (BSIMM) [21]. Estas metodologías tienen una serie de actividades de seguridad que cubren todo el proceso de construcción del software.

En este sentido, Grégoire, B. De Win y otros [22] comparan teóricamente y en función de la fase las características comunes de los modelos CLASP y Microsoft SDL. Posteriormente, amplían su trabajo en [23] para incluir en la comparativa el modelo Touchpoints [24], agrupando las similitudes entre los procesos según la fase del ciclo de vida tradicional en la cual se realizan.

Sin embargo, algunos marcos de trabajo como Microsoft ASDL, TSP-Secure, OpenSAMM y BSIMM, pasan desapercibidos al no encontrarse estudios que los analicen, siendo modelos de desarrollo seguro muy conocidos.

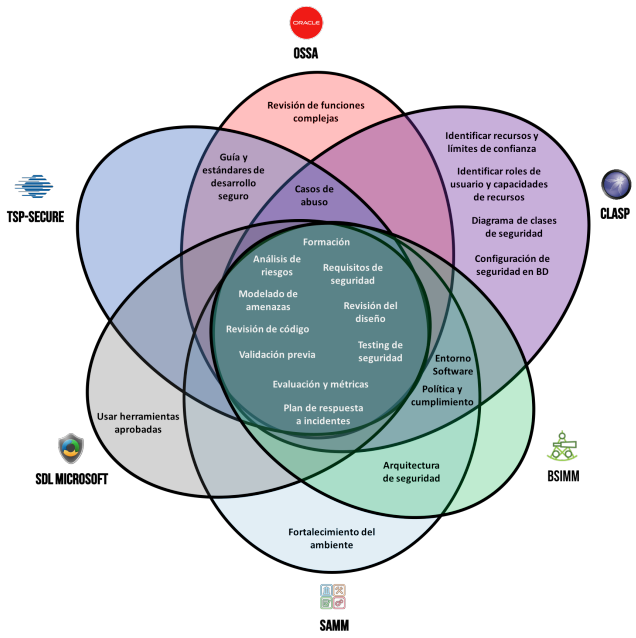


Fig. 2. Diagrama de Venn Euler que compara los modelos estudiados.

Tabla I
ACTIVIDADES DE SEGURIDAD POR MODELO DE DESARROLLO SEGURO EN FUNCIÓN DE FASE DE EJECUCIÓN

Fase/Modelo	Microsoft SDL/ASDL	OSSA	CLASP	TSP-Secure	SAMM	BSIMM
Políticas			• Identificar la política de seguridad global • Identificar recursos y límites de confianza • Identificar roles de usuario y capacidades de recursos • Especificar el entorno operativo		• Política y cumplimiento	• Política y cumplimiento
Formación	• Formación	• Formación	• Programa de sensibilización sobre seguridad institucional	• Formación	• Educación y orientación	• Formación
Análisis	• Análisis de riesgos de seguridad y privacidad • Requisitos de seguridad	• Definición de riesgos • Requisitos de seguridad	• Análisis de superficie de ataques y de riesgos • Requisitos de seguridad	• Análisis de riesgos • Requisitos de seguridad	• Evaluación de amenazas • Requisitos de seguridad • Arquitectura de seguridad	• Análisis de riesgos • Requisitos de seguridad • Análisis de arquitectura
Diseño	• Modelos de riesgos • Analizar superficie de ataques • Requisitos de diseño	• Modelado de amenazas • Revisión del diseño • Caso de abuso	• Modelado de amenazas • Guía y principios del diseño seguro • Diagrama de clases de seguridad • Casos de abuso	• Modelado de amenazas • Diseño seguro • Casos de abuso	• Modelado de amenazas • Revisión del diseño	• Modelos de amenazas • Características y diseño de seguridad
Implementación	• Usar herramientas aprobadas • Prohibir funciones no seguras	• Reglas y principios de codificación segura • Revisión de funciones complejas • Revisión de código	• Configuración de seguridad en BD • Análisis estático	• Guía y estándares de desarrollo seguro • Análisis estático	• Revisión de código	• Revisión de código
Pruebas	• Análisis dinámico • Fuzz testing	• Testing de seguridad	• Testing de seguridad	• Testing de seguridad • Fuzz testing	• Testing de seguridad	• Testing de seguridad
Pre-Release	• Revisión de seguridad final • Lanzamiento o archivado	• Validación del cumplimiento	• Validación de seguridad			
Post-Release	• Plan de respuesta a incidentes	• Corrección de vulnerabilidades	• Dirección de problemas de seguridad reportados	• Lista de vulnerabilidades con sus posibles mitigaciones	• Gestión de vulnerabilidades • Securización del entorno • Habilitación operativa	• Gestión de configuración y de vulnerabilidades
Métricas	• Umbrales de calidad y límites de errores	• Criterios de seguridad que deberá cumplir el producto	• Monitorizar métricas de seguridad	• Medir métricas de seguridad	• Estrategia y métricas	• Estrategia y Métricas
Uso industrial	X	X	X	X	X	X

Tampoco se encuentran estudios que propongan la creación de nuevos modelos adaptados a las necesidades actuales o que añadan nuevas actividades de seguridad que suplan posibles carencias de los anteriores.

Los estudios encontrados se basan únicamente en clasificar y/o mostrar las similitudes extraídas de sus estudios. Por ello, en trabajos anteriores a esta investigación [25] se analizan los principales modelos de desarrollo de software seguro empresariales [15]–[21]. Estos incluyen tanto metodologías ágiles como tradicionales. Como punto de partida, se hace una comparativa de todos los conjuntos citados, basada en las versiones más actualizadas de cada marco de trabajo. En la *Tabla I*, se muestran las actividades de seguridad de los modelos nombrados en función de la fase del ciclo de vida donde se ejecutan.

En *Fig. 2*, se ilustran las similitudes y diferencias de los modelos estudiados en la *Tabla I*, mediante un diagrama de Venn Euler que permite percibir gráficamente la comparativa de los modelos analizados.

La comparativa denota que hay modelos como TSP-Secure, OSSA y CLASP, que comparten actividades de seguridad entre sí, como son los casos de abuso. Del mismo modo, los marcos de trabajo CLASP, SAMM y BSIMM convergen en las actividades relativas a las políticas y la securización del entorno de desarrollo.

III. Paradigmas Emergentes

Ante el panorama actual, resulta necesario cambiar sustancialmente el proceso de desarrollo de software, proponiendo nuevos paradigmas en los que el producto software final sea el resultado de un proceso de desarrollo que combine funcionalidad con seguridad, ofreciendo respuestas preventivas que se anticipen a cualquier vulnerabilidad. En este sentido, Hamid y Weber proponen en [26] un enfoque metodológico de ingeniería dirigida por modelos (MDE) con un enfoque basado en patrones para apoyar el desarrollo de sistemas de software seguros.

Este trabajo presenta un modelo emergente fundamentado en una perspectiva de seguridad preventiva que garantiza un software igual de funcional que el obtenido con los modelos clásicos y reactivos, pero más seguro y con mayor eficacia para la productividad de una empresa desarrolladora.

En *Fig. 3*. se observa el modelo emergente propuesto, organizado en cuatro áreas de desarrollo y compuesto por catorce actividades de seguridad. Once actividades son tomadas de los puntos comunes e identificadas en la parte central de *Fig. 2*. Además, se proponen tres nuevas actividades.

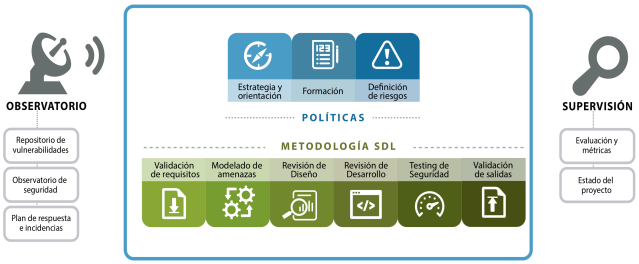


Fig. 3. Modelo de software emergente propuesto, compuesto por catorce actividades de seguridad.

Se considera necesario organizar de forma sistemática y planificada las actividades de seguridad que construyen el software seguro, como se indica en otros trabajos [27].

El modelo propuesto clasifica las actividades comunes y las propuestas en áreas de desarrollo, similar a las funciones de negocio, tal y como se organizan los modelos SAMM y BSIMM. De otra forma distinta, Microsoft SDL y su versión ágil ASDL, ordenan los procedimientos seguros según su ejecución en las fases del ciclo de vida del software tradicional.

Las áreas de desarrollo son cuatro: Políticas, Metodología de Desarrollo Seguro, Supervisión y Observatorio. A continuación, se describen brevemente las competencias que desempeña cada área.

La finalidad del área *políticas* es crear y unificar la estrategia de seguridad que se va a llevar a cabo para conseguir que el software construido sea seguro. Se centra en definir las directrices globales y objetivos de seguridad a nivel normativo que debe cumplir un proyecto software de un sector concreto. Es preciso aclarar que las actividades englobadas en este área van a requerir la participación activa de todos los grupos implicados en el proceso de construcción del software.

El área *metodología SDL* se orienta específicamente en el proceso de construcción para que el software construido sea seguro por defecto.

El área *supervisión* se encarga de controlar los indicadores de seguridad del software y de realizar una evaluación final de la seguridad del software entregado.

El área *observatorio de seguridad* se enfoca en realizar una continua vigilancia con la finalidad, de detectar la aparición de vulnerabilidades desconocidas hasta el momento y abrir nuevas líneas de investigación, desarrollo e innovación (I+D+i) que estudien novedosas e innovadoras técnicas de ciberataques desconocidos.

IV. Modelo Emergente Preventivo

El modelo emergente presentado incluye tres novedosas actividades. Pese a que los modelos estudiados tienen la suficiente validez para tomarlos de referencia, se han detectado algunas carencias en ellos ocasionadas por el paso del tiempo, el acelerado avance de la tecnología en este ámbito y el enfoque reactivo que tienen, anteponiendo la finalidad de solventar fallos de seguridad existentes sobre la prevención de los mismos. De esta forma, se detecta la necesidad de incluir actividades encargadas de investigar la aparición de nuevas vulnerabilidades, de retroalimentar empíricamente el modelo haciendo uso de los fallos cometidos y de actividades que manifiesten y controlen el estado instantáneo sobre la seguridad del producto software que se está construyendo. Seguidamente, se detallan las nuevas actividades propuestas por los autores y su objetivo.

A. *Observatorio de Seguridad*

El objetivo principal de la actividad es evitar que el software sea inseguro el menor tiempo posible. Esto minimiza el tiempo de exposición y los factores de riesgo. Conocer pronto una vulnerabilidad proporciona tiempo en la búsqueda de soluciones o parches de seguridad. Se sabe que el software que se desarrolla con seguridad en la actualidad no tiene por qué ser seguro en el futuro. Por esto, es imprescindible investigar posibles nuevas vulnerabilidades desconocidas que emergen cada día. La información de fuentes de prestigio del campo de la seguridad informática que publican vulnerabilidades y técnicas de ataque recientes es de gran utilidad. Esto permite automatizar las labores de generación del software seguro y su validación. Así mismo, esto afecta positivamente a la reputación de los equipos de desarrollo y la confianza de los clientes.

B. *Repositorio de Vulnerabilidades*

Pese a que algunos modelos TPS-Secure, SAMM y BSIMM listan y gestionan vulnerabilidades, podrían mejorar desde su propia experiencia. Se propone aprender de los fallos de seguridad previamente cometidos. Siendo la finalidad de esta actividad transformar las medidas reactivas en preventivas, la clave fundamental del modelo emergente propuesto. Dichas mejoras deben adoptarse en las fases iniciales del proceso de desarrollo del software. De esta forma, se evita cometer ciertos fallos de seguridad de nuevo en fases avanzadas.

Estos errores van a construir una base de conocimiento que forme al equipo de desarrollo. Se hace uso de la información empírica para mitigar futuros errores. Esta actividad debe ser concebida como una práctica dinámica dentro del ciclo de vida del software seguro. Además, se demuestra que permite una eficiente resolución de vulnerabilidades.

C. *Estado del Proyecto*

La gestión de la seguridad de varios productos software resulta compleja, debido al avance en paralelo de los distintos proyectos. Se propone esta nueva actividad para no perder la perspectiva preventiva de seguridad actual de cada software. El objetivo principal es comprobar el cumplimiento de las directrices de seguridad y normativas marcadas al inicio de su construcción. Esto permitirá adaptar los recursos destinados a solventar situaciones de seguridad críticas y facilitar la resolución de incidencias y cumplimiento que implican algunos estándares de calidad del software, como por ejemplo, el Modelo de Madurez y Capacidad Integrado (CMMI).

V. Puntos Fuertes del Modelo Propuesto

Dos aspectos son considerados diferenciadores en el modelo propuesto. El primero de ellos se encuentra relacionado con la formación de los desarrolladores y auditores del software en materia de seguridad. El segundo es la metodología de trazabilidad para identificar riesgos de seguridad, obtener requisitos o historias de seguridad y definir buenas prácticas de codificación segura. Ambas cuestiones presentadas de manera preventiva.

A. *Formación en Seguridad*

Se ha construido un entorno de entrenamiento en materia de seguridad, particularizado e innovador, que sirve de apoyo y ejemplo para que los desarrolladores aprendan a construir software seguro. Así, se suplen posibles carencias generalizadas que la enseñanza no especializada en seguridad deja al anteponer conocimientos relativos a funcionalidad, usabilidad y escalabilidad del software frente a conocimientos relativos a la seguridad del mismo.

Existen diversas herramientas creadas de forma vulnerable [28], [29] y [30] cuyo objetivo es mejorar las habilidades de los desarrolladores y concienciarlos ante la necesidad de poner especial atención de las cuestiones de seguridad. Sin embargo, el modelo propone una nueva herramienta que, a diferencia de las anteriores, ofrece la posibilidad de visualizar simultáneamente dos escenarios diferentes: uno vulnerable y otro seguro. Como se observa en *Fig. 4*.

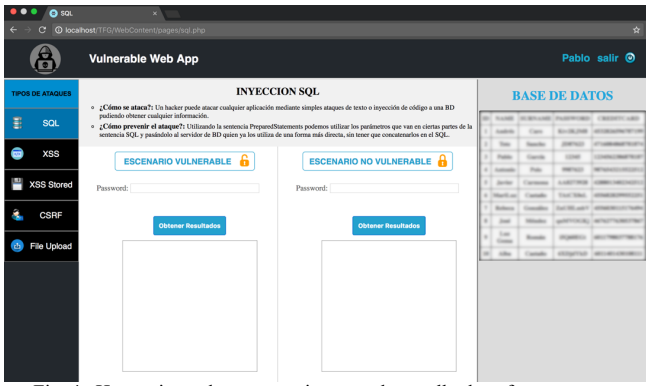


Fig. 4. Herramienta de entrenamiento en desarrollo de software seguro.

El entorno particularizado replica las vulnerabilidades más explotadas en la actualidad [31] y aporta a los roles de auditor-desarrollador una doble visión educadora que les capacita para construir software sin vulnerabilidades.

B. *Trazabilidad en la Metodologías SDL*

El enfoque preventivo y emergente del modelo está basado en la incorporación sistemática de prácticas de seguridad adecuadamente planificadas en el ciclo de vida del software, como se observa en *Fig. 5*. Dicho proceso necesita un seguimiento en su trazabilidad. Así, el modelo emergente evita que el diseño e implementación de software se oriente únicamente a la funcionalidad.

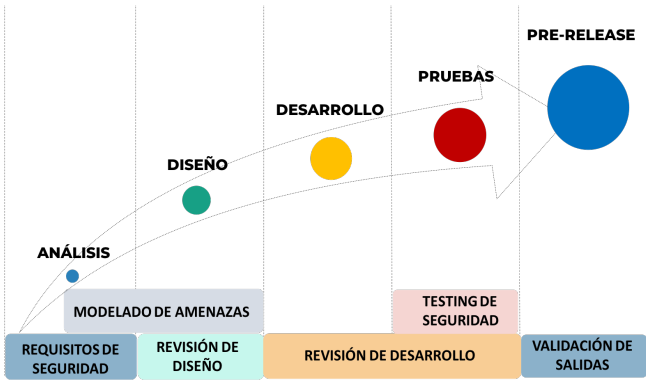


Fig. 5. Actividades de seguridad en función de la fase de ciclo de vida.

Mediante la resolución de un cuestionario automatizado [32], se extraen de los requisitos funcionales aquellos aspectos intrínsecos de seguridad. Similar a lo que proponen Pietikäineny otros [33], para obtener historias de usuarios de seguridad de forma genérica. Así, se identifican los posibles riesgos de seguridad de la aplicación que, posteriormente, se transforman en requisitos o historias de seguridad y se modelan las amenazas relativas a los requisitos de seguridad obtenidos. Esto permite que, finalmente, se defina un libro blanco de buenas prácticas que consiga un software seguro. Como se puede percibir en el modelo priman las actividades de carácter preventivo que pretenden anticiparse al fallo del software. Como método de comprobación en las fases finales se hace uso de herramientas automatizadas que miden la seguridad y calidad del código.

VI. Caso de Estudio: Experimento Comercial

Como caso de estudio se presenta un experimento comercial aplicado a dos escenarios distintos. Un primer escenario basado en modelos tradicionales (enfoques reactivos) y otro escenario fundamentado en modelos emergentes (basados en la prevención y anticipación de vulnerabilidades). Seguidamente se detalla el contexto de aplicación, las características del software y del equipo de desarrollo, así como la evaluación aplicada junto a los indicadores de seguridad y productividad utilizados para medir la eficacia del modelo emergente.

A. *Empresa de Software*

El experimento industrial que se presenta a continuación se lleva a cabo en la empresa Viewnext, del grupo IBM España. Fruto de Cátedra de Patrocinio sobre Seguridad y Auditoría de Sistemas Software, firmada en el año 2015 entre la Universidad de Extremadura (UEx) y la empresa Viewnext, una de las múltiples actividades que la UEx hace en este ámbito [34]. Esta empresa se encuentra formada por un equipo de más de 4.500 profesionales especializados en el desarrollo de software. Distribuida de manera descentralizada en varias oficinas y centros de innovación tecnológica ubicados en España y Portugal. La empresa se segmenta en prácticas, prestando servicios de desarrollo y mantenimiento en remoto desde cualquiera de los centros de innovación. Destacar que en el experimento realizado para construir software seguro, la práctica ADM Desktop/Web se encarga de ejecutar todas las actividades del área de Metodología SDL, salvo la de testing de seguridad. Esta actividad es ejecutada por la práctica de Calidad y Pruebas.

B. *Características del software y del equipo de desarrollo*

El software sometido a la metodología planteada pertenece al sector de la industria eléctrica, un sector que destaca por su alta criticidad, ya que los riesgos y consecuencias de un ataque cibernético a una compañía eléctrica pueden llegar a ser devastadores para la sociedad.

El equipo de desarrollo se encuentra formado por quince personas gestionados a través de una metodología *agile* cuya frecuencia de entregas al cliente es mensual. La metodología *agile* y la frecuencia de entregas al cliente condicionan la planificación de las actividades de seguridad dentro del ciclo de vida.

Se hace uso de herramientas que facilitan la gestión de la demanda y de la imputación del tiempo dedicado a la construcción del software.

C. Evaluación

La evaluación pretende exponer las diferencias existentes a nivel de seguridad y productividad en el software construido en los dos escenarios considerados. En *Fig. 6*. se observan las fases realizadas en ambos escenarios.

El escenario tradicional contempla la construcción del software sin tener en cuenta la seguridad hasta la fase de pruebas. Se presenta con un enfoque reactivo al identificar y corregir vulnerabilidades en sus fases finales. Tras ella, reactivamente se corrigen las vulnerabilidades identificadas.

Antes de aplicar el modelo emergente al siguiente escenario, se capacita al equipo de desarrollo en materia de desarrollo seguro durante 30 horas y se construye un ecosistema seguro mediante la integración de herramientas de seguridad y chequeo continuo.

El escenario preventivo anticipa la detección de vulnerabilidades del software durante su construcción, sin esperar a las fases finales. Este escenario soporta la implantación del modelo emergente preventivo propuesto, siguiendo el procedimiento presentado en [35]. Seguidamente, y al igual que el escenario anterior se realiza una evaluación de seguridad mediante una auditoría y, por último, se corrigen las vulnerabilidades detectadas.

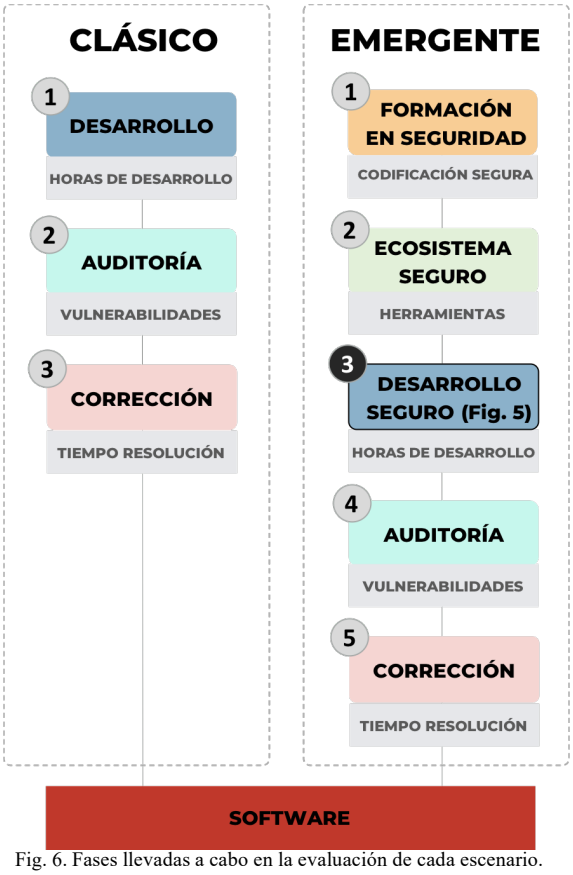


Fig. 6. Fases llevadas a cabo en la evaluación de cada escenario.

Ambos escenarios se consideran equivalentes al encontrarse desarrollados por el mismo equipo de personas, mismo framework arquitectónico y tener la misma complejidad funcional. Únicamente, cambia la metodología de desarrollo del software como se especifica en la imagen anterior (*Fig. 6*). Por este motivo, es posible considerar válidos a efectos de extrapolar y comparar futuros resultados.

D. Indicadores de seguridad y productividad

Para medir la eficacia del modelo emergente propuesto se utilizan diversos indicadores relativos a la seguridad del software y del rendimiento en la productividad de su desarrollo.

Para evaluar el nivel de seguridad se valora el número de vulnerabilidades detectadas, el tipo y su criticidad. El tipo de las vulnerabilidades se relaciona con cuestiones tales como la arquitectura de la aplicación (*web server, application server, database, frameworks, custom code*, etc.), o aspectos más puros del desarrollo de software (*injections, XSS, broken authentication, sensitive data exposure*, etc.). Para medir la criticidad se utiliza el estándar *Common Vulnerability Score System* [36]. Este estándar clasifica las vulnerabilidades en cinco categorías: crítico, alto, medio, bajo e informativo.

Por otro lado, para comparar la productividad o eficacia de cada escenario, se mide el tiempo destinado a su desarrollo. Relativizar el tiempo de desarrollo en coste nos permite conocer si es más rentable seguir utilizando un modelo clásico o avanzar hacia el modelo emergente y preventivo que se presenta.

VII. Resultados escenario Clásico vs. Emergente

Tras aplicar la metodología de evaluación propuesta se presentan en este apartado los resultados obtenidos para ambos escenarios.

En cuanto a los indicadores encargados de medir la seguridad, en el escenario clásico se detectaron 19 vulnerabilidades: 12 relacionadas con cuestiones de arquitectura de la aplicación y 7 con aspectos del desarrollo. Por su parte, en el escenario emergente se identificaron 6 vulnerabilidades, siendo 5 de arquitectura y 1 de desarrollo.

En la *Tabla II* se detallan las vulnerabilidades encontradas en cada escenario, clasificadas según su tipo (de arquitectura o de desarrollo) y su criticidad (según el estándar Common Vulnerability Scoring System [37]).

Es preciso aclarar que, en ningún caso, el plan de pruebas de seguridad ejecutado y que ha identificado las vulnerabilidades presentadas haya influido en el desarrollo del escenario preventivo. No existiendo cruce de información entre el desarrollo de ambos escenarios, pese a ser realizados por el mismo equipo de desarrolladores.

Las vulnerabilidades han sido resueltas debido al alto riesgo del sector de este experimento comercial.

En *Fig. 7*, se observa una reducción considerable del número de vulnerabilidades (19 en el escenario clásico frente a 6 en el emergente), que implica una reducción de, aproximadamente, un 66% en cuanto al número de vulnerabilidades por las que se ve afectado el software desarrollado.

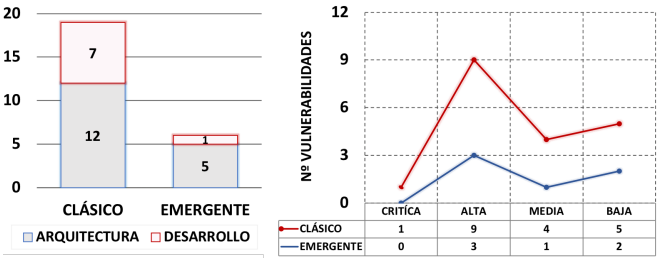


Fig. 7. Comparativa del número, tipo y criticidad, en función del estándar CVSS versión 3 de las vulnerabilidades en función del escenario auditado.

Tabla II
VULNERABILDADES IDENTIFICADAS EN FUNCIÓN DE CADA ESCENARIO. CRITICIDAD CLASIFICADA EN FUNCIÓN DE CVSS VERSIÓN 3

Categoría	Vulnerabilidad	Criticidad	Tipo	Clásico	Emergente
Validación de entradas	Stored Cross Site Scripting	Crítica [9-10]	Desarrollo	X	
	Reflected Cross Site Scripting	Alta [7-8.9]	Desarrollo	X	
	Base de datos accesible	Baja [0.1-3.9]	Desarrollo	X	X
	Atributo autocomplete no inhabilitado	Baja [0.1-3.9]	Desarrollo	X	
	Cambio de peticiones POST por GET	Alta [7-8.9]	Arquitectura	X	
	Directiva POST con parámetros no validados	Alta [7-8.9]	Desarrollo	X	
Autorización	Escalada de privilegios funcional	Alta [7-8.9]	Desarrollo	X	
	Escalada de privilegios por URL	Alta [7-8.9]	Desarrollo	X	
Gestión de sesiones	Identificador de sesión desprotegido	Alta [7-8.9]	Arquitectura	X	
	Cierre de sesión no implementado correctamente	Media [4.0-6.9]	Arquitectura	X	
Manipulación de errores y logs	Información sensible del aplicativo y uso de componentes vulnerables	Media [4.0-6.9]	Arquitectura	X	
	Información sensible en los metadatos	Baja [0.1-3.9]	Arquitectura	X	
	Información sensible en el código fuente	Media [4.0-6.9]	Arquitectura	X	
Gestión de la configuración	Página del servidor por defecto	Baja [0.1-3.9]	Arquitectura	X	
	Aplicativo en HTTP en lugar de HTTPS	Alta [7-8.9]	Arquitectura	X	
	Phising a través marcos	Alta [7-8.9]	Arquitectura	X	
	Inyección de enlaces	Alta [7-8.9]	Arquitectura	X	
	Certificados SSL débiles	Alta [7-8.9]	Arquitectura		X
	Servicios y puertos habilitados indebidamente	Alta [7-8.9]	Arquitectura		X
	Respuesta de TCP timestamp	Baja [0.1-3.9]	Arquitectura	X	X
	Conexión Concurrente desde distintas IPs	Media [4.0-6.9]	Arquitectura	X	X
	Firma SMB (Server Message Block) No requerida	Alta [7-8.9]	Arquitectura		X

No menos importante resulta el estudio sobre el impacto de tales vulnerabilidades. En el escenario clásico, el 73.68% de las vulnerabilidades detectadas se clasifican en las categorías de media/alta/crítica, siendo un porcentaje realmente preocupante si se extrapolan estas cifras al panorama de desarrollo de software habitual en nuestros días.

Considerando que el modelo propuesto reduce considerablemente el número de vulnerabilidades del software, así como el impacto de las mismas, es evidente que la propuesta presentada construye un software más seguro por defecto que los modelos clásicos.

Debido a que los escenarios clásico y emergente son distintos con respecto al tiempo de desarrollo global, se extrapola al valor porcentual para cada fase definida en el apartado anterior.

La *Fig. 8*, muestra que en el modelo emergente dedica porcentualmente más tiempo en las fases tempranas del desarrollo de software, dado que se consideran cuestiones de seguridad para proporcionar software no solo funcionalmente correcto, sino también seguro.

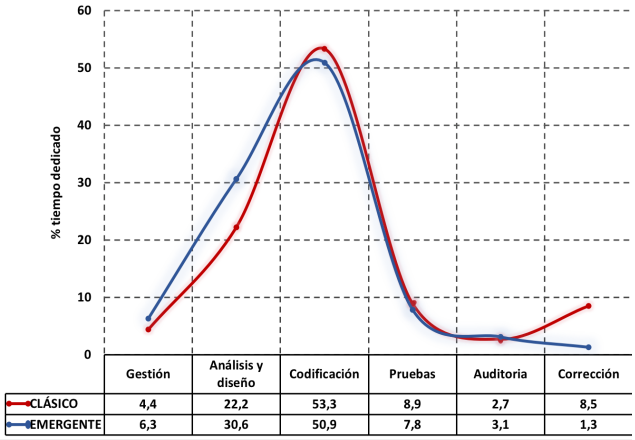


Fig. 8. Comparativa del tiempo dedicado, en porcentaje, para cada fase del proceso de desarrollo de los escenarios sometido a pruebas.

Estos tiempos no solo se recuperan en las fases finales del proceso de desarrollo, sino que, además, como ha quedado demostrado, el producto final incluye menos vulnerabilidades y de menor impacto.

Se puede apreciar fácilmente que existe una gran diferencia entre ambos escenarios con respecto a la fase relativa a la corrección de vulnerabilidades. El escenario clásico sufre un gran impacto temporal en esta fase, debido al enfoque reactivo que le caracteriza. Sin embargo, el escenario emergente al ser preventivo distribuye el coste relativo a las actividades de seguridad a lo largo de todas las fases del desarrollo de software. De esta forma, la fase de corrección de vulnerabilidades tiene un coste mínimo.

Un dato cualitativo detectado por el Scrum Manager del proyecto es la diferencia temporal entre las fases de análisis y diseño con respecto a la de codificación. El escenario emergente evidencia que una mayor dedicación a las fases iniciales de análisis y diseño reduce el tiempo dedicado a la codificación.

Se destaca la novedad de esta propuesta en el ámbito de los modelos de software emergentes al ser la primera investigación que presenta una comparativa de los resultados de aplicar un modelo clásico y reactivo frente a un modelo emergente y preventivo. Hasta el momento no se han encontrado estudios con los que poder contrastar los resultados expuestos. Esto es debido a la gran dificultad práctica que entraña evaluar proyectos reales de software, lo que magnifica la importancia de los resultados obtenidos y las conclusiones expuestas y que se comparte en este trabajo.

Asimismo, es importante destacar la transferencia de conocimiento directa que se produce entre la Universidad de Extremadura y la empresa Viewnext.

VIII. Conclusiones

Se presenta la comparativa de un experimento comercial para dos escenarios distintos, uno clásico que aplica la seguridad de manera reactiva, y otro, emergente que de manera preventiva tiene en cuenta la seguridad durante todo el ciclo de vida del software. Ambos escenarios son desarrollados por el mismo equipo de personas, no existiendo cruce de información del escenario clásico al escenario emergente. Por este motivo, se afirma que los resultados obtenidos son derivados únicamente de las diferencias metodológicas aplicadas a cada escenario.

El uso de metodologías reactivas – modelo clásico – genera un alto impacto temporal en la resolución de vulnerabilidades, debido al retraso en la detección. El resultado de esta contribución indica que no contemplar la seguridad en el desarrollo del software hasta el final del proceso conlleva excesiva dedicación para arreglar los fallos de seguridad.

La implantación de un nuevo modelo preventivo – escenario emergente – reduce el número de vulnerabilidades un 66%, reduce la criticidad de las mismas y mejora la productividad de las empresas desarrolladoras, al minimizar el tiempo de corrección de vulnerabilidades. El modelo propuesto demuestra que es posible producir software funcionalmente correcto y capaz de detectar vulnerabilidades de forma preventiva, generando, de este modo, software más seguro y de mayor calidad.

Agradecimientos

Los autores agradecen la financiación recibida por parte de la Junta de Extremadura (Fondo Europeo de Desarrollo Regional), Consejería de Economía e Infraestructuras (Proyecto GR18138) y la cesión de los datos a ViewNext, empresa de Servicios de Tecnologías del grupo IBM España.

Referencias

[1] Response National Cryptologic Center Computer Emergency Team (CCN-CERT), "Cyberthreats and tendencies. Executive Summary 2018."

[2] H. Tran, E. Campos-Nanez, P. Fomin, and J. Wasek, "Cyber resilience recovery model to combat zero-day malware attacks," *Computers and Security*, vol. 61, pp. 19–31, 2016.

[3] S. S. Murtaza, W. Khreich, A. Hamou-Lhadj, and A. B. Bener, "Mining trends and patterns of software vulnerabilities," *Journal of Systems and Software*, vol. 117, pp. 218–228, 2016.

[4] A. Apvrille and M. Pourzandi, "Secure software development by example," *IEEE Security and Privacy*, vol. 3, no. 4, pp. 10–17, 2005.

[5] D. Mellado, E. Fernandez-Medina, and M. Piattini, "A Security Requirements Engineering Process in Practice," *Latin America Transactions, IEEE (Revista IEEE America Latina)*, 2007.

[6] National Institute of Standards and Technology, "The Economic Impacts of Inadequate Infrastructure for Software Testing," *Quality*, 2002. [Online]. Available: https://www.nist.gov/sites/default/files/documents/director/planning/report02-3.pdf. [Accessed: 18-Jun-2017].

[7] M. Solinas, L. Antonelli, and E. Fernandez, "Software secure building aspects in computer engineering," *IEEE Latin America Transactions*, vol. 11, no. 1, pp. 353–358, 2013.

[8] Y. Yang, J. Du, and Q. Wang, "Shaping the effort of developing secure software," in *Procedia Computer Science*, 2015, vol. 44, no. C, pp. 609–618.

[9] A. S. S. Sodiya, S. A. A. Onashoga, and O. B. B. Ajayi, "Towards building secure software systems," *Information Universe: Journal of Issues in Informing Science & Information Technology*, vol. 3, pp. 635–646, 2006.

[10] A. Rehman and T. Saba, "Evaluation of artificial intelligent techniques to secure information in enterprises," *Artificial Intelligence Review*, vol. 42, no. 4, pp. 1029–1044, 2012.

[11] R. Jones, "Secure Coding: Building Security into the Software Development Life Cycle," *Information Systems Security*, 2004.

[12] N. S. A. Karim, A. Albuolayan, T. Saba, and A. Rehman, "The practice of secure software development in SDLC: an investigation through existing model and a case study," *Security and Communication Networks*, 2016.

[13] H. Kaur, Daljit Kaur, Parminder Singh, "Secure Spiral: A Secure Software Development Model," *Journal of Software Engineering*, 2012.

[14] L. Ben Othmane, P. Angin, H. Weffers, and B. Bhargava, "Extending the agile development process to develop acceptably secure software," *IEEE Transactions on Dependable and Secure Computing*, 2014.

[15] S. Lipner and M. Howard, "The Trustworthy Computing Security Development Lifecycle," *Annual Computer Security Applications Conference, copatrocinada por IEEE*, 2004. .

[16] Microsoft Corporation, "Agile Development Using Microsoft Security Development Lifecycle," 2010. [Online]. Available: https://www.microsoft.com/en-us/SDL/Discover/sdlagile.aspx. [Accessed: 19-Jun-2017].

[17] C. O. C. Redwood Shores, "Oracle Software Security Assurance," 2011. [Online]. Available: https://www.oracle.com/support/assurance/index.html. [Accessed: 20-Jul-2017].

[18] OWASP Project, "Comprehensive, Lightweight Application Security Process." [Online]. Available: https://www.owasp.org/index.php/CLASP_Concepts. [Accessed: 01-Jul-2017].

[19] N. Davis, P. L. Miller, W. R. Nichols, and R. C. Seacord, "TSP-Secure," 2009.

[20] OWASP Project, "Software Assurance Maturity Model," 2009. [Online]. Available: http://www.opensamm.org/downloads/SAMM-1.0.pdf. [Accessed: 20-Jul-2017].

[21] G. McGraw, B. Chess, and S. Miques, "Building security In maturity model," *2012 Faulkner Information Services*, no. May. pp. 1–61, 2011.

[22] J. Grégoire, K. Buyens, B. De Win, R. Scandariato, and W. Joosen, "On the secure software development process: CLASP and SDL compared," in *Proceedings - ICSE 2007 Workshops: Third International Workshop on Software Engineering for Secure Systems, SESS'07*, 2007.

[23] B. De Win, R. Scandariato, K. Buyens, J. Grégoire, and W. Joosen, "On the secure software development process: CLASP, SDL and Touchpoints compared," *Information and Software Technology*, vol. 51, no. 7, pp. 1152–1171, 2009.

[24] G. McGraw, "Software Security: Building Security in," in *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, 2006.

[25] J. C. Sancho Núñez, A. Caro Lindo, and P. García Rodríguez, "Análisis de metodologías de Desarrollo de Software Seguro," in *Jornadas Nacionales de Investigación en Ciberseguridad(JNIC)*, 2016, pp. 42–47.

[26] B. Hamid and D. Weber, "Engineering secure systems: Models, patterns and empirical validation," *Computers and Security*, 2018.

[27] J. C. Sancho Núñez, A. Caro Lindo, P. García Rodríguez, and Á. Quesada, "Categorización de Actividades de Seguridad en el Desarrollo de Software," in *Jornadas de Ingeniería del Software y Bases de Datos*, 2016, pp. 565–568.

[28] OWASP, "OWASP WebGoat Project." [Online]. Available: https://www.owasp.org/index.php/Proyecto_WebGoat_OWASP.

[29] J. Druin and C. Walker, "Introduction to the OWASP Mutillidae II Web Training Environment," 2013.

[30] D. Vulnerable and W. Application, "Damn Vulnerable Web Application (DVWA) Official Documentation," *ReVision*, 2010.

[31] OWASP, "OWASP Top 10 - The Ten Most Critical Web Application Security Risks," 2017.

[32] J. C. Sancho Núñez, A. Caro Lindo, L. Fondón, and J. A. Félix de Sande, "Herramienta para la identificación de requisitos de seguridad en un Modelo de Desarrollo Seguro," in *Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, 2018, pp. 92–95.

[33] P. Pietikäinen, J. Röning, T. Siiskonen, and V. Ylimannela, *Handbook of The Secure Agile Software Development Life Cycle*. 2014.

[34] A. Caro Lindo, "Una apuesta por la educación en ciberseguridad desde el ámbito universitario," in *Jornadas Nacionales de investigación en Ciberseguridad (JNIC)*, 2015, pp. 189–202.

[35] J. C. Sancho Núñez, A. Caro Lindo, P. García Rodríguez, and J. A. Félix de Sande, "Metodología de Implantación Empresarial de un Modelo de Desarrollo de Software Seguro," in *Jornadas Nacionales de investigación en Ciberseguridad (JNIC)*, 2017, pp. 128–133.

[36] P. Mell, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System," *IEEE Security and Privacy Magazine*, 2006.

[37] FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," *Forum of Incident Response and Security Teams (FIRST)*, pp. 1–21, 2015.

Mejora de la seguridad de esquemas de gestión de identidades federados mediante técnicas de User Behaviour Analytics

Alejandro G. Martín y Marta Beltrán
ETSII, Universidad Rey Juan Carlos
28933 Móstoles, Madrid
alejandro.garciam@urjc.es y marta.beltran@urjc.es

***Resumen*—Los esquemas federados para la gestión de identidades y accesos se han extendido espectacularmente en los últimos años en entornos web, cloud y móviles. Este tipo de especificación permite que un recurso, servicio o aplicación delegue en un proveedor de identidades externo los procesos de identificación, autenticación, autorización y auditoría (IAAA). A pesar de la madurez que están adquiriendo estas especificaciones todavía suponen amenazas para la seguridad de las infraestructuras en las que se incorporan. Este trabajo propone cinco estrategias diferentes para incorporar técnicas de User Behaviour Analytics (UBA) a los flujos IAAA federados, de manera que se puedan emplear tanto en prevención como en detección de incidentes de seguridad. Además, estas cinco estrategias se validan y evalúan en un caso de uso real.**

***Index Terms*—Esquemas federados, Gestión de identidades y accesos, Mobile Connect, OpenID Connect, User Behaviour Analytics**

Tipo de contribución: *Investigación original*

I. Introducción

En la actualidad, casi todos los usuarios de Internet trabajan en un momento u otro con un proveedor de identidades. Esto les permite gestionar todos sus accesos a diferentes recursos, servicios y aplicaciones con una o dos cuentas abiertas en estos proveedores en lugar de tener que gestionar una cuenta local en cada uno de estos recursos, servicios y aplicaciones y de tener que recordar su contraseña en cada uno de ellos.

Para los proveedores de recursos, servicios y aplicaciones, esta solución, que implica confiar en un proveedor externo, también suele ser ventajosa, ya que les garantiza escalabilidad, les permite ahorrar costes en la solución de la identificación, autenticación, autorización y auditoría (IAAA) de sus usuarios e incluso les permite transferir parte de los riesgos que estos procesos suponen hoy en día.

Por este motivo grandes proveedores tecnológicos como Facebook, Google o Twitter, las operadoras de telecomunicaciones o los bancos han apoyado el desarrollo de especificaciones para la gestión de identidades federadas como SAML, OpenID, OAuth, OpenID Connect o Mobile Connect, convirtiéndose ellos mismos en proveedores de identidades en estos esquemas.

Las diferentes especificaciones que han surgido desde el año 2002 hasta el momento han demostrado su utilidad en multitud de contextos, pero diferentes trabajos de investigación e incidentes de seguridad en entornos de producción han demostrado también que es necesario mejorar los niveles de seguridad que ofrecen en la actualidad.

En este trabajo se propone la utilización de técnicas de User Behaviour Analitycs (UBA) para conseguir este tipo de mejora. En realidad, no se trata más que de aprovechar la última A, la de auditoría, para analizar el comportamiento que los usuarios tienen cuando usan su identidad a partir de toda la información que los diferentes agentes que construyen las federaciones suelen almacenar. Esta información permite comprender lo que los usuarios hicieron en el pasado, modelar lo que se considera que es normal para ellos en el presente (y por lo tanto, detectar anomalías), predecir lo que harán en el futuro, agruparles con otros usuarios que tienen comportamientos similares, etc. Todas estas capacidades pueden emplearse para mejorar los actuales flujos IAAA, tanto en aspectos de prevención (evitando así posibles suplantaciones) como de detección (levantando alertas cuando estas suplantaciones ya se han producido).

Las principales contribuciones de este artículo son: 1) Analizar el estado del arte en esquemas federados de gestión de identidades y en técnicas de User Behaviour Analytics para encontrar las vulnerabilidades de seguridad de los primeros que pueden evitarse o mitigarse integrando de alguna forma las segundas. 2) Proponer cinco estrategias de integración concretas, discutiendo en cada caso la forma de incorporación a los esquemas federados, las funcionalidades que permiten incorporar y las limitaciones encontradas. 3) Analizar la viabilidad y evaluar estas propuestas en un caso de uso real.

El resto de este artículo se estructura de la siguiente manera. La Sección 2 analiza el estado del arte en esquemas federados de gestión de identidades y en técnicas de User Behaviour Analytics, partiendo de este análisis para motivar la investigación realizada. La Sección 3 presenta las cinco estrategias propuestas para mejorar la seguridad de esquemas IAAA federados mediante la integración con técnicas de UBA. La Sección 4 analiza, evalúa y discute la utilidad, rendimiento y seguridad de estas propuestas en un caso de uso real. Y finalmente la Sección 5 presenta las principales conclusiones obtenidas y sugiere algunas líneas interesantes de trabajo futuro.

II. Estado del arte y motivación

II-A. Esquemas federados de gestión de identidades

SAML y OpenID fueron los dos primeros estándares abiertos que se extendieron en entornos web para resolver la autenticación de forma federada. SAML es un producto del comité OASIS, se lanzó en el año 2003 y actualmente se

encuentra en la versión 2.0. OpenID surgió unos años después, en el 2006, impulsado por la OpenID Foundation y la última versión, la 2.0, se considera obsoleta en la actualidad.

La versión que se recomienda utilizar es OpenID Connect, que no es más que la combinación de OpenID con OAuth. OAuth es una especificación centrada en la resolución de la autorización. También se trata de un estándar abierto cuya primera versión estuvo disponible en el año 2007, actualmente se utiliza la versión 2.0.

Por lo tanto, OpenID Connect es capaz de resolver autenticación y autorización en un único flujo IAAA (ya que combina OpenID y OAuth). La especificación Mobile Connect se basa en ella y es muy similar, pero está propuesta por la GSMA, la asociación que agrupa a las operadoras de telefonía, por lo que se centra en adaptarla para poder utilizar el número de teléfono como identidad y las diferentes posibilidades que ofrece un dispositivo actual como alternativas o complemento a la contraseña como autenticadores (algo que se tiene, algo que se sabe, algo que se es, algo que se hace).

Multitud de trabajos de investigación han analizado hasta el momento la seguridad de todas estas especificaciones. En un primer grupo de trabajos se encuentran las investigaciones que analizan implementaciones concretas de estas especificaciones y/o que proponen patrones de ataque específicos que permiten materializar distintas amenazas en escenarios en los que estas implementaciones se emplean. Buenos ejemplos de estos trabajos son [1] para OpenID; [2] para OAuth, [3] ó [4] para OpenID Connect. En los trabajos más recientes se proponen herramientas que permiten automatizar pruebas de seguridad para estos escenarios. En [5] hay una comparativa muy completa de herramientas para OAuth.

En un segundo grupo de este tipo de trabajos, se incluyen investigaciones que proponen análisis formales de las propias especificaciones buscando mejorar la seguridad de sus sucesivas versiones. Este tipo de análisis formales suelen llevar a la propuesta de contramedidas o mitigaciones que incrementen los niveles de seguridad ofrecidos, en algunos casos se han modificado las especificaciones directamente, en otros casos estas propuestas se han traducido en recomendaciones o mejores prácticas para las fases de implementación y despliegue. En esta línea resultan muy interesantes los análisis propuestos en [6] para OpenID , [7] para OAuth o en [8] para OpenID Connect. Además, el IETF publicó en el año 2013 su RFC 6819 con una serie de recomendaciones muy exhaustivas para mejorar la seguridad de entornos OAuth [9].

II-B. User Behaviour Analytics (UBA)

User Behavior Analytics es una disciplina científica basada en modelar los comportamientos de los usuarios de un sistema, servicio o aplicación para conseguir objetivos de negocio. La finalidad principal de UBA es comprender, analizar y predecir los comportamientos pasados, presentes y futuros utilizando técnicas de Machine Learning (ML) [10]. En la mayoría de investigaciones realizadas hasta el momento, los comportamientos se modelan como sucesiones de interacciones del usuario con el sistema. Por ejemplo, sucesiones de páginas web visitadas por el usuario a lo largo del tiempo o sucesiones de pulsaciones en la pantalla táctil de un dispositivo móvil.

Las soluciones basadas en UBA se encuentran en un momento de gran expansión en el área de la ciberseguridad, con casos de uso en la detección de intrusiones, ataques, suplantaciones de identidad o fraude (sirviéndose para ello de diferentes técnicas de ML). Es por este motivo que existen bastantes ejemplos que incorporan este tipo de técnicas para mejorar sistemas de control de accesos. Por ejemplo, en [11] confía directamente en el uso de técnicas de UBA para resolver el problema de la autenticación de usuarios. En [12] se avanza un poco más, convirtiendo esta autenticación en continua para dispositivos móviles (utilizando como fuente de datos las pulsaciones en la pantalla que realiza el usuario al interactuar con el dispositivo). Otro claro ejemplo de incorporación de UBA al control de accesos es [13], en este caso se utiliza para resolver la autorización. En esta investigación los autores definen una métrica de riesgo de acceso a un documento protegido en base al comportamiento previo del usuario que solicita acceso y del contenido del propio documento.

II-C. Motivación

Hasta donde nosotros sabemos, ningún trabajo previo aborda el problema de integrar técnicas de UBA en esquemas federados para la gestión de identidades. Aunque ya se han modelado, descrito y analizado todas las amenazas que estos esquemas pueden suponer para la seguridad de una infraestructura, las soluciones y mitigaciones propuestas hasta el momento no se plantean el uso de estas técnicas sino que suelen ir orientadas al enriquecimiento de las peticiones y de los tokens, al fortalecimiento de la gestión de las sesiones de usuario, a la mejora de las SDKs y APIs ofrecidas a los desarrolladores en las RPs o al uso de criptografía a diferentes niveles.

Sin embargo, como se ha explicado en la sección anterior, estas técnicas de UBA están demostrando ser herramientas muy potentes para realizar control de accesos en entornos complejos (multi-dispositivo, multi-plataforma, con usuarios muy heterogéneos, etc.). Las siguientes secciones de este artículo tienen como objetivo proponer diferentes estrategias para realizar esta integración dependiendo de los objetivos planteados y de la posibilidad/imposibilidad de modificar los flujos IAAA en diferentes pasos. Cabe destacar que las estrategias propuestas podrían incluirse en futuras especificaciones de los esquemas federados o simplemente, proporcionarse como extensiones que mejoran sus capacidades cuando así se considere necesario (extensiones añadidas en las fases de implementación y despliegue con funcionalidades adicionales a las descritas en las especificaciones estándar).

III. Incorporación de técnicas de User Behaviour Analytics a esquemas federados de gestión de identidades

III-A. Arquitectura y flujo genéricos para un esquema federado de gestión de identidades

En esta sección se proponen cinco estrategias para incorporar análisis de comportamiento de usuarios a las especificaciones ya mencionadas de manera que se incrementen los niveles de seguridad ofrecidos por ellas. Estas propuestas pretenden ser genéricas, es decir, no estar ligadas a una especificación o

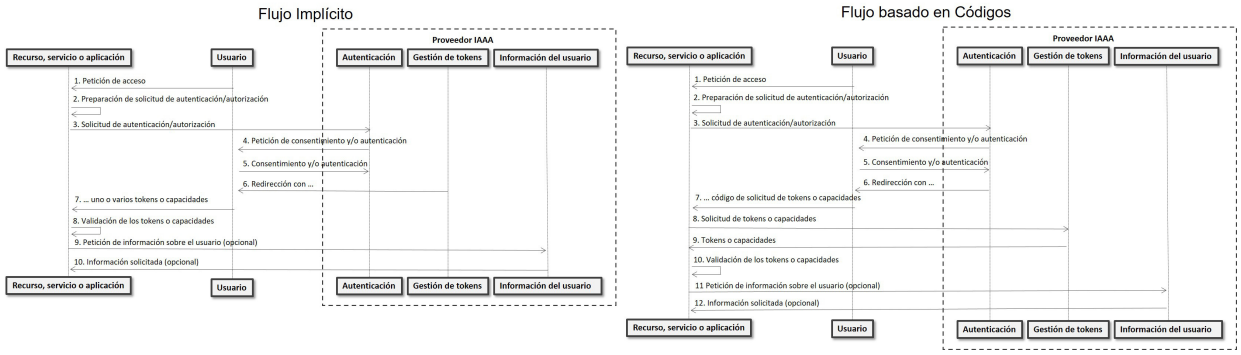


Figura 1. Flujos IAAA con un esquema federado: flujo implícito y flujo basado en códigos.

versión concreta. Por ello es necesario comenzar, en esta sección, por identificar una arquitectura y un flujo genérico para todas las especificaciones generadas que permitan trabajar con lo que todas ellas tienen en común a la hora de proponer las estrategias de integración con técnicas UBA.

En lo que se refiere a la arquitectura, en todas las especificaciones federadas se distinguen los siguientes elementos o agentes:

- El cliente, *principal* o Relying Party (RP): Es cualquier recurso, aplicación o servicio que confía en la especificación y en un proveedor externo para decidir si un usuario final tiene acceso o no a lo que solicita. Como ya se ha comentado, algunas especificaciones resuelven la autenticación, otras la autorización y otras, ambos aspectos.
- El proveedor de identidades o Identity Provider (IdP): Se trata del proveedor en el que recae la responsabilidad de resolver la función IAAA y en el que los clientes y usuarios finales confían dentro del esquema federado.
- Usuario final o End-User (EU): Es el usuario que inicia el flujo IAAA al intentar acceder a un recurso, aplicación o servicio alojado en el cliente.

Desde el punto de vista del flujo IAAA, la Figura 1 resume los pasos o etapas que se suelen seguir en todas las especificaciones en los dos enfoques habituales, el de flujo implícito y el basado en códigos.

Como se puede observar, en el paso 1 el usuario final solicita acceder a un recurso, servicio o aplicación. El servidor que lo aloja prepara una solicitud de servicio para el proveedor de identidades (pidiendo que se resuelva la autenticación del usuario, la autorización o ambas en el paso 2). A continuación, en el paso 3, esta solicitud se envía al proveedor de identidades adecuado, o bien porque se ha preguntado por él al usuario final cuando este ha solicitado el acceso en el paso 1 o bien porque se utiliza alguno de los servicios de descubrimiento dinámico especificados para este esquema en concreto.

Cuando el IdP recibe esta solicitud, pueden pasar dos cosas: o bien el usuario ya tiene iniciada sesión en él y basta con pedirle su consentimiento para realizar este flujo IAAA, o bien no la tiene y es necesario pedirle autenticación explícita además de este consentimiento (paso 4). El usuario responde en el paso 5, normalmente en el primer caso se trabaja con *cookies* de sesión mientras que en el segundo se suele realizar la autenticación mediante una contraseña (aunque algunas especificaciones como Mobile Connect, por ejemplo, permiten autenticación multi-factor con autenticadores algo más sofisticados).

Si se trata de un flujo implícito (Figura 1), en el paso 6 el proveedor de identidades redirige al usuario con uno o varios *tokens* o capacidades hacia el cliente (paso 7). El cliente valida lo que recibe (en el paso 8 es necesario comprobar campos como el emisor, la audiencia, el alcance, los tiempos de validez, etc.) y si todo es correcto, el acceso se podría realizar con garantías. Se suele usar un *token* o capacidad cuando el flujo resuelve únicamente autenticación o autorización (SAML, OpenID, OAuth), suelen ser dos cuando resuelve los dos aspectos (OpenID Connect o Mobile Connect).

Si se trata de un flujo basado en códigos (Figura 1), en el paso 6 se redirige al usuario con un código hacia el cliente (paso 7). Este código permite al cliente recuperar en el proveedor de identidades el o los tokens o capacidades (pasos 8 y 9). Este tipo de flujos evitan que estas piezas de información, críticas para la seguridad de todo el esquema, pasen por el dispositivo del usuario final, que suele ser vulnerable a malware, exploits maliciosos, ataques de cross-site scripting, etc. O que incluso pueden perderse o ser robados.

Por lo demás, este segundo tipo de flujo es igual que el primero. Cabe destacar que todas las especificaciones se proponen para escenarios en los que las comunicaciones se realizan con HTTP (y por lo tanto se protegen criptográficamente mediante el uso de TLS cuando es necesario) y en los que el agente involucrado en el dispositivo del EU es el navegador web.

Por último es importante conocer el concepto de Level of Assurance o LoA tal y como lo propone el estándar ISO/IEC 29115. Según este estándar, una autenticación errónea no siempre implica el mismo riesgo, por lo que diferentes peticiones de autenticación pueden requerir diferentes niveles de seguridad en esta autenticación. De momento sólo Mobile Connect incorpora la posibilidad de realizar peticiones de autenticación con diferentes niveles de LoA, pero poco a poco el resto de especificaciones federadas están contemplando, de una forma u otra, este concepto. Un nivel 1 de LoA implica muy poca confianza en la autenticación realizada, puede conseguirse con un botón de OK en la pantalla de un móvil o con una *cookie* con tiempo de vida largo en un navegador web. El nivel 2 se basa en algo que sólo el usuario conoce, que es típicamente una contraseña (suele ser el LoA por defecto cuando no se contempla que se puedan solicitar

diferentes niveles. El nivel 3 implica autenticación multifactor, por lo que añade nuevos autenticadores que pueden ser de nuevo algo que sólo el usuario conoce (una contraseña de un sólo uso, un PIN, etc.), algo que sólo el usuario posee (un token hardware, una *smart card)*) o algo que sólo el usuario es o hace (una fotografía de la cara, una huella dactilar). El nivel 4 de LoA implica autenticación criptográfica y requiere del uso de una Public Key Infrastructure o PKI.

III-B. Estrategia 1: Establecimiento de LoA para la Authentication Request

Como ya se ha explicado, algunas especificaciones federadas permiten que la RP determine el grado de confianza que el IdP debe tener en la autenticación que realiza el usuario. Esta decisión se deja completamente en manos de cada RP, que suele optar por requerir siempre el mismo nivel de LoA para todos los usuarios o por clasificar las interacciones por niveles de seguridad y pedir siempre el mismo LoA para todas las del mismo grupo (por ejemplo, el más bajo para lecturas, el siguiente nivel para escrituras y el siguiente para ejecución). En este trabajo se propone que el LoA se calcule basándose en técnicas UBA (entre los pasos 1 y 2 de los flujos representados en la Figura 1). La RP posee información acerca del comportamiento de sus usuarios, por lo menos cuando ha habido interacciones previas a una solicitud de acceso específica, y estos datos son los que permiten la utilización de este tipo de técnicas.

Supóngase el caso en que una aplicación ofrece dos formas de acceder a un recurso protegido por dos rutas diferentes: la ruta a y la ruta b. Después de analizar la información histórica de los usuarios, se detecta que los usuarios que utilizan el sistema desde hace más tiempo utilizan mayoritariamente la ruta a. Por otro lado, los usuarios nuevos utilizan indistintamente ambas rutas. Se puede concluir entonces, que existen dos tipos de comportamientos distintos dependiendo de la antigüedad del usuario, es decir, de cuánto tiempo lleven utilizando la aplicación. Si un usuario con antigüedad que accede normalmente por la ruta a, intenta acceder al recurso por la ruta b, su comportamiento se puede considerar anómalo. Esto puede llevar a la RP a solicitar un LoA superior al habitual.

Este ejemplo tan sencillo sirve para ilustrar cómo se podría emplear esta estrategia de integración de UBA en esquemas federados mediante el uso, en la RP, de modelos de comportamiento automatizados basados en algoritmos de ML. Algunos ejemplos de cómo modelar comportamientos de usuario con ML que podrían servir para definir el LoA de una petición son [14], [15]. El resultado de estos algoritmos de ML es un valor numérico que representa la confianza que se tiene en el comportamiento del usuario, normalmente un porcentaje de seguridad sobre una secuencia de acciones. Discretizando esta métrica de confianza se podrían obtener directamente los distintos niveles de LoA que se deberían pedir en la solicitud de autenticación/autorización (paso 3 de la Figura 1).

La principal ventaja de esta estrategia es que la RP es el agente del flujo de IAAA que posee información más detallada, actualizada y precisa sobre el comportamiento de sus propios usuarios cuando acceden a los recursos que ofrece. Esta información permite construir modelos fiables y robustos, que además podrían complementarse con otras estrategias (explicadas a continuación) ofrecidas desde el IdP. El principal inconveniente es el coste que implican para la RP, que no siempre dispone del personal y/o de la infraestructura para implementarlas con éxito.

III-C. Estrategia 2: Risk-based authentication

Los IdP almacenan toda la información relacionada con el histórico de interacciones de cada una de las identidades que gestionan para cada uno de los recursos, servicios y aplicaciones a los que se ha accedido (de ahí la última A del acrónimo IAAA, la Auditoría). Esto permite entrenar modelos de ML que cuantifiquen cómo de normal es una solicitud de autenticación/autorización recibida en un instante concreto. Estas métricas se pueden extraer, no sólo con información del histórico de una identidad, sino también con información del comportamiento de identidades similares a la evaluada con el recurso, servicio o aplicación solicitado. Además, también se puede evaluar el comportamiento de todas las identidades para recursos, servicios o aplicaciones similares al solicitado, consiguiendo así modelos más precisos y potentes.

En este trabajo se propone que el IdP, una vez recibida la solicitud de autenticación/autorización en el paso 3 de la Figura 1, calcule cómo de normal o anormal es esta petición para realizar la autenticación del usuario que se produce a continuación y que además cuantifique el riesgo que se corre al atenderla. Si la petición que llega de la RP va sin LoA, es el propio IdP el que asume la responsabilidad de decidir hasta qué punto debe exigirle al usuario durante esta autenticación que demuestre que es quien dice ser. Si la petición lleva un LoA, el IdP puede decidir, dependiendo de cómo se implemente esta estrategia, modificar lo que la RP ha solicitado (relajándolo o endureciéndolo) teniendo en cuenta que posee más información que este agente del flujo (de otras identidades similares, de otras RPs similares).

Supóngase una aplicación corporativa con dos tipos de usuarios, privilegiados y estándar, y dos tipos de datos etiquetados, confidenciales y públicos. Los accesos más frecuentes son los de usuarios privilegiados a datos confidenciales y los de usuarios estándar a datos públicos, pero esto no significa que no se puedan dar otras combinaciones. El comportamiento de todos los usuarios en esta aplicación puede modelarse en el IdP. De esta forma, si un usuario privilegiado intenta acceder a un dato confidencial, se considera que el riesgo es medio y se le pide un factor de autenticación concreto. Si intenta acceder a un dato público, es una anomalía, pero dada la naturaleza del recurso, el riesgo que se corre es bajo, por lo que basta con que el usuario tuviera iniciada sesión en el IdP. Si un usuario estándar accede a un dato público, ocurre exactamente lo mismo. Y si un usuario estándar solicita acceso a un dato confidencial, se trata de una petición anómala y además de alto riesgo, por lo que el IdP decide solicitar durante la autenticación posterior dos factores de autenticación, uno de ellos biométrico.

Esta estrategia propone que el IdP utilice técnicas de UBA basadas en los datos almacenados en su infraestructura: secuencias de solicitudes de acceso por parte de los usuarios cuyas identidades se gestionan. Existen trabajos que muestran como modelar dichas secuencias para extraer patrones de comportamiento y establecer métricas de riesgo, de manera efectiva y con una precisión alta [16], [13]. Además, estos

modelos se pueden combinar con una clasificación de los recursos, servicios o aplicaciones para determinar el riesgo que supone una suplantación de identidad en cada uno de ellos. Es decir, cuantificar los posibles daños que supondría aceptar una solicitud maliciosa dependiendo del recurso, servicio o aplicación al que se esta intentando acceder.

La principal ventaja de esta estrategia es que el IdP, utilizando técnicas sencillas de UBA, puede generar modelos de riesgo muy fiables, proporcionando así seguridad adaptativa que se ajuste al riesgo que se corre en cada momento. El principal inconveniente vuelven a ser los costes asociados a la utilización de estas técnicas, aunque es menos crítico que en el caso de la estrategia anterior dada la naturaleza de los proveedores de identidades actuales (grandes empresas tecnológicas como Facebook o Google, operadoras de telecomunicaciones, bancos).

III-D. Estrategia 3: Autenticación continua

La autenticación continua, al contrario que la tradicional, define la autenticación como un proceso que se alarga en el tiempo, que comienza cuando el usuario inicia sesión y no concluye hasta que el usuario la finaliza. Este proceso consiste en evaluar los comportamientos del usuario una vez que ya ha establecido la sesión con el fin de detectar anomalías que indiquen que el usuario, aún habiendo proporcionado correctamente sus credenciales, no es quien dice ser (porque ha sufrido un robo de credenciales, porque se ha secuestrado la sesión una vez iniciada, porque se ha perdido su dispositivo, etc.). Esta forma de autenticación ha demostrado ser efectiva tanto en ordenadores [17] como en dispositivos móviles [12].

En este trabajo se proponen dos formas de incorporar autenticación continua basada en UBA en el flujo de IAAA de esquemas federados, ya que la autenticación continua puede dar buenos resultados en dos momentos diferentes de estos flujos. En primer lugar, a partir del paso 4, petición de consentimiento y/o autenticación, por parte del IdP. En segundo lugar, a partir del paso de validación de los tokens o capacidades (paso 8 o paso 10 dependiendo del tipo de flujo). En este segundo caso, es la RP es la encargada de realizar la autenticación continua.

En el primer caso (paso 4), se parte de la misma premisa y se utilizan los mismos datos y modelos que en la estrategia 2. La diferencia reside en que para realizar autenticación continua el usuario ya ha sido previamente autenticado y que además, no se realiza un análisis puntual de una petición concreta sino que se analiza de manera constante el comportamiento del usuario mientras tiene iniciada sesión en el IdP. Supóngase que un usuario se autentica para acceder a un servicio con un IdP específico. Si durante la sesión el IdP detecta comportamientos extraños en ese usuario que le hacen sospechar que ha habido una brecha de seguridad, puede bloquear su *cookie* de sesión (que suele ser el método escogido por todos los IdPs para evitar que los usuarios tengan que introducir constantemente sus contraseñas y demás autenticadores) y obligarle a iniciar de nuevo la sesión proporcionando autenticadores adicionales, por ejemplo.

En el segundo caso (paso 8 o paso 10), la propia RP puede evaluar las interacciones del usuario con el recurso, servicio o aplicación una vez validados sus *tokens* o capacidades. El principio es el mismo que en el caso anterior pero empleando los datos y modelos que están a disposición de la RP, es decir, los mencionados en la estrategia 1. Mientras que el IdP analiza el comportamiento de un usuario evaluando su histórico de solicitudes de autenticación/ autorización y atributos estáticos y dinámicos de las mismas (direcciones IP, horarios, dispositivos utilizados, etc.), la RP evalúa la sucesión de acciones que realiza un usuario sobre su recurso, servicio o aplicación. Por ejemplo, en el caso de una aplicación móvil se podrían evaluar las dinámicas de pulsaciones en la pantalla. En caso de que las métricas de normalidad en un instante determinado obtuvieran valores bajos y la RP sospechara que ha ocurrido una brecha de seguridad, la RP debería invalidar los *tokens o capacidades* recibidos y comenzar un nuevo flujo de IAAA (a ser posible, con una LoA superior, por ejemplo).

Las ventajas de incorporar autenticación continua a los flujos de IAAA son obvias, ya que al alargar el proceso de autenticación en el tiempo se pueden prevenir y detectar muchas de las amenazas de suplantación que se sufren en la actualidad. Y además sin que esto repercuta apenas en la calidad de experiencia de los usuarios, para los que estos procesos son prácticamente transparente. Sin embargo, en esta estrategia el aumento de los costes para el IdP o la RP sí puede considerarse un grave inconveniente. El motivo se debe a que se ha de monitorizar constantemente el comportamiento de todos y cada uno de los usuarios para poder aplicar estas estrategias en tiempo real.

Una posible solución para reducir el coste computacional que esta estrategia implica para el IdP o la RP es delegar la computación necesaria para realizar autenticación continua, o al menos parte de ella, al propio dispositivo del usuario (se podría ofrecer como una extensión o mejora de seguridad). Sin embargo, esta solución no siempre es posible, ya que, en primer lugar si el usuario utiliza un dispositivo móvil puede ver mermado su rendimiento. En segundo lugar, perder el control sobre el cómputo que permite implementar autenticación continua y realizarlo en un dispositivo tan vulnerable como el del usuario final añadiría nueva superficie de exposición a los esquemas IAAA.

Por otro lado, hay que mencionar que una incorrecta implementación de estos métodos puede resultar en modelos de ML con un *ratio* de falsos positivos alto. Esto se traduce en la invalidación de sesiones activas de usuarios legítimos, empeorando su calidad de experiencia y consumiendo recursos innecesariamente. Este suceso ocurre sobre todo cuando el IdP o la RP no tienen apenas información acerca de los usuarios, es decir, con usuarios nuevos; o cuando los usuarios cambian de comportamiento en un corto espacio de tiempo. Ya se han propuesto soluciones muy interesantes para solventar estos problemas, por ejemplo [18].

III-E. Estrategia 4: Detección de suplantación y/o fraude

Las estrategias propuestas hasta este momento se incorporan dentro del flujo de IAAA para prevenir suplantaciones en tiempo real (o casi real), es decir, para evitar el impacto que tendría que un adversario se hiciera pasar por su víctima al interactuar con el IdP y/o con la RP. Sin embargo, es posible que en ciertos casos no se pueda optar por estas estrategias, por ejemplo, por falta de personal especializado, por el coste que suponen los recursos de cómputo necesarios para conseguir esta ejecución en tiempo real o por la falta

de los datos necesarios para construir los modelos. Esto no significa que no se pueda aprovechar el potencial de las técnicas UBA.

En este trabajo se propone la utilización, a posteriori (off-line), de las mismas técnicas empleadas para implementar las estrategias anteriores. Es decir, con esta estrategia las técnicas de UBA permiten sacar el máximo partido posible a la última A del acrónimo IAAA (Accountability o Auditoría), que en muchos casos se olvida, desaprovechando toda la información almacenada. Con esta estrategia los modelos se ejecutan en *batches* por parte de la RP o del IdP, de manera periódica o cuando se produzca un evento determinado, y con el objetivo de analizar todo el histórico de interacciones o un cierto periodo de tiempo concreto. Esto permite detectar situaciones anómalas, suplantaciones y fraudes, pero no predecirlos o evitarlos. El objetivo sería alertar a las víctimas y adoptar las contramedidas necesarias para evitar que se repitan este tipo de incidentes en el futuro.

La principal ventaja de esta estrategia es que es más sencilla (no exige el tratamiento de nuevas solicitudes en streaming o la monitorización continua del comportamiento de los usuarios) y que, por lo tanto, se reducen mucho sus costes en comparación con las tres anteriores. Incluso permite, por ejemplo, programar las ejecuciones en periodos de baja carga para aprovechar recursos que de otra forma estarían desaprovechados en la RP o el IdP. Por otro lado, el principal inconveniente es que no permiten realizar prevención, sólo detección, y los efectos de una suplantación y/o fraude pueden ser irreversibles.

III-F. Estrategia 5: Registro dinámico de usuarios/dispositivos

Como ya se ha explicado anteriormente, casi todas las especificaciones federadas estandarizan la manera en la que un usuario puede registrarse en un IdP. Cuando se trata de un IdP como Facebook, Google o Twitter, este registro se hace de manera remota, a través de un formulario, ya que que estos proveedores no están obligados a conocer la identidad real de una persona que se asocia a la identidad digital registrada (aunque lo intentan ya que es beneficioso para su modelo de negocio). Si el IdP es una operadora de telecomunicaciones o un banco, normalmente pueden realizar este registro en remoto porque en algún momento, físicamente, el usuario habrá probado su identidad mediante su DNI o documento identificativo equivalente. Lo mismo ocurre con agencias públicas o gubernamentales que necesitan mapear identidades digitales a físicas.

Como también se ha explicado ya, las especificaciones federadas han evolucionado en los últimos años proponiendo perfiles y versiones específicas que permiten identificar dispositivos y objetos en escenarios IoT en lugar de usuarios. En este caso, el proceso de registro manual para cada dispositivo se convierte en una tarea tediosa y laboriosa que incluso puede llegar a ser inviable (un proyecto IoT puede incorporar millones de sensores y actuadores). Es el típico escenario en el que se recurre al registro dinámico para ahorrar tiempo y costes [19]

En este trabajo se propone incorporar a estos procesos de registro dinámico de dispositivos técnicas de UBA, más específicamente, técnicas de SBA (por Sensor Behaviour Analytics). El objetivo es realizar el enrolment de un dispositivo IoT en su correspondiente IdP comprobando para ellos que es quien dice ser mediante el análisis de su comportamiento y/o contexto ambiental como dispositivo aislado e independiente [20], o incluso analizando también el comportamiento de dispositivos vecinos [21]. En este caso, el encargado de realizar el modelado es de nuevo el IdP, en una fase que es previa a la realización de cualquier flujo de IAAA, ya que se trata de la fase de Registro. El IdP trabajará con unas métricas de confianza extraídas de los modelos SBA construidos, de manera que, por ejemplo, en una fase inicial no se permita realizar flujos de IAAA con el dispositivo, después cuando se haya comprobado su identidad hasta un determinado grado ya se pueda comenzar a trabajar con él y sólo se deje que realice tareas sensibles cuando se haya comprobado su identidad con un grado de certeza suficiente (es decir, es habitual combina esta estrategia de registro dinámico con una de tipo *risk-based*).

La principal ventaja de esta estrategia es que permite validar la identidad de un dispositivo que nunca a interactuado con una RP específica o con el propio IdP de manera dinámica y remota, teniendo en cuenta su comportamiento y contexto. Esto implica un ahorro de costes importante en proyectos de IoT, Smart Cities, etc. donde el número de dispositivos es muy elevado. Las técnicas UBA permiten validar de forma automática la identidad de la mayoría de usuarios y dispositivos, pero si se quiere conseguir certeza plena siempre es necesario el registro manual, es muy importante tener esto presente. Además, en este trabajo no se propone el uso de esta estrategia para el registro dinámico de usuarios, porque los datos en los que se basarían los modelos necesarios para realizarlo son demasiado sensibles y las técnicas para recogerlos (normalmente a través de dispositivos móviles y *wearables*) muy invasivas.

IV. Análisis y discusión

IV-A. Caso de uso

Las ciudades inteligentes permiten a los ayuntamientos y administraciones locales mejorar la calidad de las personas que viven en ellas mediante diferentes tipos de servicios y aplicaciones. Este trabajo se centra, como caso de uso, en una aplicación de aparcamiento inteligente, que permite monitorizar las plazas de aparcamiento que están libres en una zona de la ciudad (tanto en superficie como en aparcamientos municipales), estimar el tráfico que hay hasta llegar a ellas, aconsejar acerca de las mejores rutas (teniendo en cuenta diferentes criterios) y reservar y pagar el aparcamiento. El proyecto en el que se ha diseñado y desplegado esta aplicación integra los datos que provienen de diferentes tipos de sensores y cámaras en la ciudad (para localizar las plazas de aparcamiento libres, para medir la densidad del tráfico, para estimar la contaminación, para medir el tiempo que una plaza está ocupada, etc.). Toda esta información se sube al centro de control de la ciudad inteligente, donde se procesa, analiza y almacena. Además, se completa con información que proviene de teléfonos inteligentes o tabletas y que proporcionan los propios ciudadanos, agentes de la policía o de movilidad y vigilantes de aparcamientos, por ejemplo, para alertar sobre cualquier incidente o imprevisto.

Por lo tanto en la arquitectura de este caso de uso, el ciudadano que utiliza la aplicación es el EU (se permite acceder a través de navegador y también se proporciona una app para el móvil), el servidor de la ciudad desde el que se ofrece es la RP (la aplicación está programada casi por completo en Java) y el IdP está controlado por el propio ayuntamiento, empleando OpenID Connect/Mobile Conect para gestionar las identidades y accesos. En concreto, se ha utilizado el software OpenAM (Open Access Manager) 13.5 de ForgeRock para desplegar este proveedor. Se trata de una solución certificada por la OpenID Foundation, muy extendida, de código abierto y fácilmente extensible por la cantidad de APIs y SPIs que expone así como los plugins, scripts y módulos que se pueden utilizar para incorporar nuevas funcionalidades.

Los usuarios de la aplicación se deben registrar en el IdP por medio de un formulario de forma manual. Para dar de alta todos los sensores y cámaras que utiliza la aplicación es necesario que un administrador realice también de manera manual este procedimiento de registro.

IV-B. Análisis de viabilidad

En este trabajo el caso de uso nos sirve para validar si es posible integrar las cinco estrategias propuestas en los flujos IAAA federados tal y como están especificados en la actualidad, o si sería necesario introducir modificaciones en las especificaciones.

- Estrategia 1: Para integrar esta estrategia en el caso de uso ha sido necesario modificar la RP para tomar decisiones acerca del LoA requerido para cada petición. En este primer prototipo, se ha desarrollado una funcionalidad muy sencilla en relación con el pago del aparcamiento a través del teléfono móvil y de Mobile Connect: si un usuario solicita realizar un pago que no encaja son su comportamiento habitual de movilidad (por la zona de aparcamiento, por la duración del estacionamiento, etc.) se sube el nivel de LoA requerido de 2 a 3 (es decir, se solicita un segundo factor de autenticación). La integración de esta estrategia en el flujo IAAA sólo afecta al desarrollo de la RP (que pasar de tomar decisiones acerca del LoA solicitado de manera estática a hacerlo de manera dinámica y que necesita para ello un repositorio eficiente que almacene el comportamiento pasado de sus usuarios), ya que el IdP no necesita ninguna modificación y no es necesario tampoco modificar la especificación de OpenID Connect/Mobile Connect.
- Estrategia 2: La integración de esta estrategia en el caso de uso se ha realizado con OpenID Connect, es decir, sin que las peticiones que vienen de la RP indiquen ningún tipo de LoA. En este caso se deja al IdP la responsabilidad de autenticar con un nivel de seguridad suficiente, bajo su criterio, a los usuarios finales. Nuestro primer prototipo evalúa un conjunto de atributos estáticos de la petición que llega (principalmente usuario, servicio solicitado, hora, geo-localización, dispositivo empleado y dirección IP) y una serie de atributos dinámicos (extraídos de un modelo UBA sencillo que sólo tiene en cuenta el comportamiento anterior de usuario individuales) para decidir si deja que el usuario se dé por autenticado si ya tenía sesión iniciada en el IdP, si se le exige que se re-autentique con una contraseña o si se exige que se re-autentique con dos factores de autenticación. Todo esto sin ninguna intervención ni conocimiento por parte de la RP. Aunque hemos conseguido implementar esta funcionalidad sobre un producto que implementa la versión actual de la especificación de OpenID Connect (mediante un *proxy* que añade las técnicas UBA en el Authorization Server de OpenAM), lo ideal sería que esta estrategia se incorporara a futuras especificaciones, ya que modifica ligeramente los flujos IAAA (añadiendo un paso en el IdP entre el 3 y el 4 de la Figura 1 y modificando la estructura del ID token, por ejemplo).
- Estrategia 3: Esta estrategia también se ha podido integrar en el caso de uso escogido para validar nuestras propuestas, los mismos modelos que se han utilizado en las estrategias 1 y 2 nos han permitido realizar autenticación continua (de momento, basándonos en técnicas muy sencillas) de los ciudadanos que utilizan la aplicación, tanto en la RP como en el IdP. En este segundo caso, como ocurre con la estrategia 2, los cambios más importantes se han introducido en el Authorization Server del IdP y lo ideal sería que vinieran dados desde la propia especificación de OpenID Connect, no como una funcionalidad adicional añadida durante la implementación.
- Estrategia 4: Al tratarse de una estrategia off-line, no modifica en absoluto los flujos IAAA. Nos ha bastado añadir un nuevo end-point en el IdP y modificar ligeramente la estructura del OpenAM data store (que no es más que un Java Identity Repository) para hacer más eficientes las tareas de auditoría. En nuestro primer prototipo, al final de cada día se aplican modelos muy similares a los empleados en la estrategia 2 pero más orientados a la detección de anomalías. Si se sospecha de un uso fraudulento de la aplicación, para una reserva o pago de aparcamiento o para notificar un incidente, se avisa al ciudadano implicado por si su cuenta/dispositivo han sido utilizados por terceros sin su conocimiento.
- Estrategia 5: De nuevo esta estrategia no obliga a modificar los flujos IAAA, se ha podido integrar en el caso de uso enriqueciendo la funcionalidad del módulo de Discovery. En este primer prototipo se ha realizado un pre-registro de sensores y cámaras basado en técnicas UBA, de manera que tras este procedimiento, sus datos todavía no se tienen en cuenta para la toma de decisiones pero se van almacenando. En el momento en el las condiciones ambientales de estos sensores y cámaras permiten saber con un cierto grado de certeza que son los legítimos, se realiza el registro definitivo y se empiezan a tener en cuenta los datos que recogen. Si pasado un tiempo no se pasa a este registro definitivo, es necesario que un administrador se desplace en persona para realizar el registro manual.

IV-C. Discusión

La Tabla I permite comparar todas las estrategias propuestas en el presente trabajo tras su validación. El *Paso* es la etapa en la que se incorpora la estrategia dentro del flujo IAAA (Figura 1). El *Agente* es el encargado de ejecutar las técnicas UBA. El *Tipo* define la función de los modelos UBA: prevención o

Tabla I
COMPARATIVA DE LAS ESTRATEGIAS PROPUESTAS.

	Paso	Agente	Tipo	Ejecución	¿Modifica el flujo?	Datos	Dificultad	Puntos débiles
Estrategia 1	1,2	RP	Prevención	Online	No	Aplicativos	2	Latencia
Estrategia 2	3	IdP	Prevención	Online	No	Histórico solicitudes	2	Latencia
Estrategia 3	4 y 8,10	RP/IdP	Prevención	Online	Si	Ambos	3	Experiencia de usuario
Estrategia 4	8,10	RP/IdP	Detección	Offline	No	Ambos	1	Ninguna
Estrategia 5	1	IdP	Prevención	Online	No	Aplicativos	2	Latencia

detección. La *Ejecución* puede tomar dos valores de nuevo, *online* y *offline* dependiendo de si las predicciones de los modelos son en tiempo real o no. La característica *¿Modifica el flujo?* permite saber si la integración de la estrategia implica una modificación de los pasos del flujo IAAA y por lo tanto sería necesario actualizar las especificaciones federadas para no depender de implementaciones concretas. *Datos* se refiere al tipo de datos que necesita el *Agente* para poder implementar las técnicas UBA. La *Dificultad* esta categorizada en 3 niveles, siendo 1 el nivel mínimo de dificultad para integrar la estrategia y 3 el nivel de máximo. Un valor de 3 no implica que la estrategia no sea viable (en la sección anterior se ha visto que todas lo son) simplemente se han considerado los costes asociados (complejidad de la integración en el flujo IAAA y consumo de recursos de cómputo teniendo en cuenta nuestra validación con un caso de uso real). Por último, *Puntos débiles* se refiere a las desventajas que supondría implantar dicha estrategia.

V. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se han propuesto cinco estrategias diferentes para integrar técnicas de User Behaviour Analytics en esquemas de gestión de identidades y accesos federados tan extendidos en la actualidad como SAML, OAuth, OpenID Connect y Mobile Connect. Un caso de uso real ha permitido demostrar que estas formas de integración son viables y adecuadas para los escenarios típicos de utilización, además de comparar sus características fundamentales. Los proveedores de identidades y de recursos que incorporen estas estrategias deberán informar a sus usuarios (consentimientos, condiciones de uso, etc.) de que observarán niveles de seguridad superiores a los tradicionales pero sacrificando, en parte, su privacidad (en mayor o menor medida dependiendo de la estrategia concreta). En la actualidad estamos trabajando en el desarrollo de modelos de Machine Learning adecuados para cada una de las estrategias propuestas y de técnicas asociadas (por ejemplo, para reducir las tasas de falsos positivos o negativos en algunos casos).

AGRADECIMIENTOS

Esta investigación ha sido financiada en parte por el Fondo Social Europeo a través del Programa Operativo de Empleo Juvenil y la Iniciativa de Empleo Juvenil (PEJ-2017-AI/TIC-6403).

REFERENCIAS

[1] B. van Delft and M. Oostdijk, "A security analysis of OpenID," in *Proceedings of the Second IFIP WG 11.6 Working Conference*, 2010, pp. 73–84.

[2] W. Li and C. J. Mitchell, "Security issues in OAuth 2.0 SSO implementations," in *Proceedings of the 17th International Conference on Information Security*, 2014, pp. 529–541.

[3] J. Krautwald, "Security analysis of the OpenIDConnect standard and its real-life implementations," Master's thesis, Ruhr Universität, 2014.

[4] W. Li and C. J. Mitchell, "Analysing the security of Google's implementation of OpenID connect," in *Proceedings of the 13th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2016, pp. 357–376.

[5] R. Yang, G. Li, W. C. Lau, K. Zhang, and P. Hu, "Model-based security testing: An empirical study on OAuth 2.0 implementations," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, 2016, pp. 651–662.

[6] S.-T. Sun, K. Hawkey, and K. Beznosov, "Systematically breaking and fixing OpenID security: Formal analysis, semi-automated empirical evaluation, and practical countermeasures," *Computers and Security*, vol. 31, no. 4, pp. 465–483, 2012.

[7] D. Fett, R. Küsters, and G. Schmitz, "A comprehensive formal security analysis of OAuth 2.0," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1204–1215.

[8] J. Navas and M. Beltrán, "Understanding and mitigating OpenID Connect threats," *Computers and Security*, vol. 84, pp. 1–16, 2019.

[9] "IETF RFC 6819 "OAuth 2.0 threat model and security considerations"," https://tools.ietf.org/html/rfc6819.

[10] L. Cao, S. Y. Philip, and V. Kumar, "Nonoccurring behavior analytics: A new area," *IEEE Intelligent Systems*, vol. 30, no. 6, pp. 4–11, 2015.

[11] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE transactions on information forensics and security*, vol. 8, no. 1, pp. 136–148, 2013.

[12] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, 2016.

[13] Z. Lu and Y. Sagduyu, "Risk assessment based access control with text and behavior analysis for document management," in *Military Communications Conference, MILCOM 2016-2016 IEEE*. IEEE, 2016, pp. 37–42.

[14] K. A. A. Bakar and G. R. Haron, "Adaptive authentication based on analysis of user behavior," in *2014 Science and Information Conference*. IEEE, 2014, pp. 601–606.

[15] I. Traore, I. Woungang, M. S. Obaidat, Y. Nakkabi, and I. Lai, "Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments," in *2012 Fourth International Conference on Digital Home*. IEEE, 2012, pp. 138–145.

[16] M. Misbahuddin, B. Bindhumadhava, and B. Dheeptha, "Design of a risk based authentication system using machine learning techniques," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*. IEEE, 2017, pp. 1–6.

[17] O. Aljohani, N. Aljohani, P. Bours, and F. Alsolami, "Continuous Authentication on PCs using Artificial Immune System," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2018, pp. 1–6.

[18] B. Tang, Q. Hu, and D. Lin, "Reducing False Positives of User-to-Entity First-Access Alerts for User Behavior Analytics," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 804–811.

[19] P. Hirmer, M. Wieland, U. Breitenbücher, and B. Mitschang, "Automated sensor registration, binding and sensor data provisioning." in *CAiSE Forum*, 2016, pp. 81–88.

[20] M. Bohge and W. Trappe, "An authentication framework for hierarchical ad hoc sensor networks," in *Proceedings of the 2nd ACM workshop on Wireless security*. ACM, 2003, pp. 79–87.

[21] T. Zahariadis, H. C. Leligou, P. Trakadas, and S. Voliotis, "Trust management in wireless sensor networks," *European Transactions on Telecommunications*, vol. 21, no. 4, pp. 386–395, 2010.

Seguridad de redes y sistemas de información: de la Directiva 2016/1148 al Real Decreto-Ley 12/2018

M. Robles Carrillo
Universidad de Granada- Network Engineering and Security Group (NESG)
mrobles@ugr.es

Abstract – **La seguridad de redes y sistemas de información ha sido objeto de regulación en la Directiva (UE) 2016/1148. Esta norma establece una serie de obligaciones que, en el caso de España, se han traducido en la adopción del Real Decreto-Ley 12/2018, de seguridad de redes y sistemas de información, y de la Estrategia Nacional de Ciberseguridad de 2019. El análisis de estas medidas y su comparación con los preceptos de la Directiva no arroja un balance completamente positivo. El incumplimiento de la misma tiene consecuencias mayores y más graves de lo que, generalmente, implica la falta de respeto de las normas internas o de otro tipo de normas internacionales.**

***Index Terms*- Directiva NIS, Real Decreto-Ley 12/2018, Estrategia Nacional de Ciberseguridad, seguridad de redes y sistemas**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

En 2016 se adopta la Directiva (UE) 2016/1148 relativa a las medidas destinadas a garantizar un elevado nivel común de seguridad de las redes y de los sistemas de información en la Unión Europea (en adelante, Directiva NIS) [1]. Las directivas son una modalidad normativa original de la UE porque establecen un objetivo y un plazo para alcanzarlo y porque solo tienen como destinatarios a los Estados que, a su vez, asumen la obligación de adoptar las normas necesarias en su derecho interno para hacer efectivo ese objetivo dentro del plazo. Esta obligación no se considera cumplida por la mera adopción de una norma interna que formalmente responda o pretenda dar respuesta a dicho objetivo. Cada Estado debe adoptar las medidas materiales, sustantivas y formales, requeridas para dar efectividad al conjunto de lo dispuesto en la directiva. Por ello, el incumplimiento de esa "obligación de transposición" puede producirse por un doble motivo: por la no adopción de la norma interna dentro del plazo previsto y/o por la transposición incorrecta de sus disposiciones. Las consecuencias jurídicas son igualmente graves en ambos casos.

En España, la Directiva NIS -que había de estar transpuesta el 9 de mayo de 2018- ha sido objeto de dos medidas principales. Por una parte, el 7 de septiembre, se aprueba el Real Decreto-Ley 12/2018, de seguridad de redes y sistemas de información (en adelante, RDL) [2] Por otra parte, el 30 de abril de 2019 se publica en el BOE la Estrategia Nacional de Ciberseguridad de 2019 (ENC) [3].

El objetivo de esta investigación es determinar si, además de no haber respetado el plazo indicado, la transposición de la Directiva NIS en derecho español ha sido o no correcta. No solo es un tema con importantes implicaciones jurídicas sino, sobre todo, un serio y gravísimo problema para la seguridad de las redes y sistemas de información. El desajuste normativo puede implicar que quienes actúen en la creencia de estar cumpliendo los requisitos de seguridad y notificación, en realidad, no lo estén haciendo o que quienes hayan invertido tiempo, esfuerzo y dinero para adaptarse a la normativa española puedan haber perdido lo último y lo primero y malgastado todo ello si esa normativa no es absolutamente conforme con la europea.

II. SEGURIDAD DE REDES Y SISTEMAS DE INFORMACIÓN

El análisis de las medidas adoptadas en España en aplicación de la Directiva NIS pone de manifiesto que la normativa interna merece en determinados aspectos una valoración positiva, mientras que en otros es cuestionable o negativa [4]. El RDL tiene una estructura y una organización clara, comprensible y más transparente que la directiva. No es un aspecto solo formal porque ello contribuye a una mayor transparencia y seguridad jurídicas por la mejor comprensión de su marco jurídico. Hay, sin embargo, aspectos en sus disposiciones que son controvertidos (A y B). Por su parte, la ENC de 2019 constituye un avance significativo, pero aún insuficiente en el ámbito de la seguridad de redes y sistemas de información (C).

A) La identificación de los OSE

El art. 4 de la Directiva NIS define un OSE como la entidad pública o privada dentro de los tipos recogidos en el anexo II que cumple los criterios establecidos en el art. 5.2. El art. 6.1 del RDL resulta innecesariamente más complejo. Por una parte, remite al procedimiento previsto en la Ley 8/2011 relativa a la protección de infraestructuras críticas para la identificación del OSE (Ley PIC). Por otra, reformula la definición de OSE de la Directiva NIS mezclando los criterios del art. 5.2 de la misma con los factores determinantes de la existencia de un "efecto perturbador significativo" recogidos en su art. 6 y modificando la clasificación de la norma europea.

Además, la norma española extiende el número de sujetos potencialmente considerados OSE por dos motivos: a) incluye los recogidos en el anexo de la Ley PIC que cubre un mayor número de ámbitos que el anexo II de la Directiva; y b) puede aplicarse a los proveedores de servicios de comunicaciones electrónicas y servicios de confianza, si se consideran operadores críticos de conformidad con la normativa española, a pesar de estar excluidos expresamente de este régimen según el art. 1.3 de la directiva. Más allá de las consecuencias jurídicas sobre un posible incumplimiento en la transposición, hay que reconocer los perjuicios que pueden derivarse para esos proveedores por el hecho de que haberse debido adaptar a una normativa, el RDL, que posiblemente no debería aplicárseles conforme a lo dispuesto en la Directiva.

B) La existencia de un "efecto perturbador significativo".

El art. 5.2 de la Directiva NIS establece el "efecto perturbador significativo" de un incidente para la prestación de

un servicio como uno de los tres criterios para la identificación de los OSE. En su art. 6 enumera los factores que determinarán la producción de dicho efecto clasificándolos en dos tipos: intersectoriales y específicos por sector.

El art. 6 del RDL se aparta innecesariamente de ese precepto por dos motivos: porque no mantiene esa distinción y porque solo enuncia los factores intersectoriales separándolos, a su vez, en dos categorías que atienden a la importancia del servicio y a la relación con los clientes de la entidad evaluada. Pero es que, además, esta norma tampoco refleja el criterio seguido por el art. 3 del Reglamento de Ejecución (UE) 2018/151 de la Comisión por el que se establecen normas de aplicación de la Directiva NIS específicamente para PSD [5].

C) La Estrategia Nacional de Seguridad de 2019

El art. 7 de la Directiva NIS establece la obligación de adoptar una estrategia nacional de seguridad de redes y sistemas de información, incluyendo una lista extensa y pormenorizada de sus contenidos. El art. 8 del RDL hace referencia a la anterior ENC como respuesta a esa obligación. Pero, finalmente, se confirma la necesidad de proceder a su revisión en la Orden PCI/870/2018 [6]. El proceso concluye con la publicación de una nueva ENC en abril de 2019.

En una primera lectura, el análisis de los contenidos de esta ENC, atendiendo a los componentes incluidos en el art. 7 de la directiva, permite avanzar algunas conclusiones previas. En primer lugar, se ha optado por una estrategia general y no específica que, posiblemente, habría facilitado la comprensión del modelo y de las obligaciones de sus componentes. En segundo lugar, hay que destacar que el Objetivo I se dedica a la seguridad y resiliencia de las redes y los sistemas de información y comunicaciones del sector público y de los servicios esenciales, pero con una innecesaria limitación de los sujetos implicados y, entre ellos, los PSD, a pesar del tratamiento conjunto que dispensa el RDL a OSE y PSD. En tercer lugar, las Líneas de acción 1 y 2, que responden a dicho objetivo, constituyen una relación de medidas lo suficientemente amplia para interpretar que cumple lo dispuesto en la directiva, pero tan ambiciosa como genérica e imprecisa. Por otra parte, la inclusión de un apartado sobre "Infraestructura digital" como segundo punto del Capítulo 1, la definición de la resiliencia como principio rector, el objetivo de potenciar la industria española de ciberseguridad, incluido el apoyo a la I+D+i en seguridad digital, las actividades de normalización y la capacitación de profesionales, así como el modelo de gobernanza merecen una valoración positiva. Es necesario, en cualquier caso, continuar el análisis de la ECN y sus desarrollos.

III. Conclusiones

Los tres temas considerados en el apartado anterior ponen de manifiesto que la normativa española de transposición de la Directiva NIS no está exenta de problemas. La investigación en desarrollo sobre los mismos ha de centrarse en la argumentación sobre su conformidad o no con el derecho europeo y sobre las consecuencias de una eventual incompatibilidad, así como sobre los contenidos de la ENC 2019.

El conjunto de problemas que plantea la transposición en España de la Directiva NIS requiere mantener una línea de investigación abierta para controlar su desarrollo, gestión y eventuales soluciones jurídicas y técnicas. La adopción de la Guía de Seguridad de las TIC sobre la gestión de incidentes en el marco del Esquema Nacional de Seguridad en junio de 2018 [7] y de la Guía Nacional de Notificación y Gestión de Incidentes por parte del Consejo Nacional de Ciberseguridad, en enero de 2019 [8], contribuye a una mejor comprensión del modelo, pero no resuelve la problemática que plantea una eventual falta de conformidad de la normativa española de desarrollo con la normativa europea.

Siguiendo la jurisprudencia del Tribunal de Justicia de la Unión Europea, las consecuencias de la transposición incorrecta de una directiva pueden ser dos: por una parte, el planteamiento de un recurso por incumplimiento contra España ante ese Tribunal, que puede desembocar, en caso de sentencia condenatoria, en una multa a tanto alzado o una multa coercitiva; y, por otra parte, la interposición de recursos ante los órganos jurisdiccionales españoles invocando directamente el incumplimiento de las disposiciones de la directiva por parte de los sujetos de derecho interno perjudicados por esa transposición incorrecta. En este caso, hay dos situaciones posibles. Si el recurso se plantea frente al Estado, podría invocarse la denominada eficacia directa vertical de las directivas en caso de no transposición o transposición incorrecta. En cambio, si el problema se plantea en el marco de relaciones horizontales, no cabría esa opción porque las directivas no tienen reconocida eficacia directa horizontal. En este supuesto, la única vía de actuación para el perjudicado sería exigir responsabilidad al Estado por los daños que le he ha causado el incumplimiento del derecho comunitario al haber transpuesto incorrectamente la directiva.

En suma, las consecuencias jurídicas en caso de transposición incorrecta de una directiva son mayores de lo que cabría imaginar. No sería un problema si la transposición de la directiva hubiese sido correcta y modélica en su alcance y contenidos garantizando la seguridad jurídica de sus destinatarios.

Agradecimientos

Este trabajo ha sido financiado parcialmente por el Gobierno de España, con fondos FEDER, a través del proyecto TIN2017-83494-R.

Referencias

[1] *DOUE*, L 194, de 19 de Julio de 2016, p. 1. M. Robles Carrillo, "Seguridad de redes y sistemas de información en la Unión Europea: ¿Un enfoque integral?", *Revista de Derecho Comunitario Europeo*, vol. 60, 2018, pp. 563-600.

[2] Real Decreto-Ley 12/2018, de 7 de septiembre, de seguridad de redes y sistemas de información, *BOE* nº 218, de 8 de septiembre de 2018, p. 87675*BOE* nº 193, de 10 de agosto de 2018, p. 80896.

[3] *BOE* nº 103, de 30 de abril de 2019, p. 43437.

[4] M. Robles Carrillo, "El proceso de transposición de la Directiva sobre seguridad de redes y sistemas de información en el derecho español", *Instituto Español de Estudios Estratégicos*, nº 78/2018, 2018, pp. 1-22.

[5] *DOUE*, L 26, de 31 de enero de 2018, p. 48. M. Robles Carrillo y P. García Teodoro, "Medidas de Aplicación de la Directiva NIS a Proveedores de Servicios Digitales:

Alcance y Limitaciones", *Actas de las IV Jornadas Nacionales de Investigación en Ciberseguridad*, Donostia-San Sebastián, 2018, pp. 151-158

[6] *BOE* nº 193, de 10 de agosto de 2018, p. 80896.

[7] CCN, *Guía de Seguridad de las TIC CCN-STIC 817, Esquema Nacional de Seguridad. Gestión de Ciberincidentes,* Junio 2018.

[8] Consejo Nacional de Ciberseguridad, *Guía Nacional de Notificación y Gestión de Incidentes*, 9 de enero de 2019.

Intelligence-Led Cyber Attack Taxonomy (C@T)

Francisco Luis de Andrés Pérez
S21Sec
España
fdeandres@s21sec.com

Mildrey Carbonell Castro
S21Sec
España
mcarbonell@s21sec.com

Resumen- **Se presenta CAT (Cyber Attack Taxonomy) como un nuevo modelo para analizar y representar ciberataques en su fase estratégica de alto nivel que permitirá organizar las tácticas y técnicas utilizadas por los atacantes de modo estructurado. Permite la representación de ataques generados por cualquier tipo de actor o escenario, incluyendo, por ejemplo, los ataques internos originados por Insiders (persona que materializa la ciber amenaza desde el interior de las infraestructuras del objetivo) no contemplados en modelos anteriores. Asimismo, CAT podrá ser utilizado para el modelado de ejercicios de ataque; desarrollo de frameworks específicos para sectores de especial riesgo como las infraestructuras críticas con impacto sobre la población o los repositorios de datos personales entre otros. Esta taxonomía podrá ser utilizada para la representación de cualquier tipo de ciberataque, tanto los presentes como los futuros, permitiendo el modelado desde los más simples a los ataques dirigidos, pasando incluso por los desarrollados para comprometer entornos industriales o internet de las cosas (IoT). Del mismo modo, podrá ser utilizado para definir las infraestructuras de defensa ante ciberataques mediante el análisis de contramedidas organizadas para cada fase de la estrategia CAT, incluso contra tácticas o técnicas concretas de cara a facilitar la detección, engaño o destrucción del ataque entre otras medidas.**

Index Terms*- Taxonomía de ataque, CAT, Tácticas, Técnicas y procedimientos, TTP's, *Cyber Kill Chain, DML, Intelligence-led cyberattack Taxonomy, Cyber Attack Matrix.

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La búsqueda de un lenguaje común que permita analizar y modelar todo tipo de ciberataques, desde los más sencillos hasta los organizados por estados rivales, es una línea de investigación muy activa en los últimos años y potenciada por la globalización de ciberataques como los ocurridos en el año 2017 por los conocidos *Wannacry* [2] o *Petya* [3].

Son numerosos los modelos generados para la representación y análisis de ciberataques, unos centrados en la estrategia, y otros en las tácticas, técnicas y procedimientos, pero hasta ahora no se ha definido un modelo unificado sólido, sino referencias a unos u otros de forma arbitraria y en ocasiones sin una argumentación técnica fundamentada.

La taxonomía de ataque CAT que se presenta en este documento, forma parte de una metodología completa de ataque, análoga a la definida por capas mediante el modelo DML (Detection Maturity Levels) orientada a la defensa, muy compacta y desarrollado por el autor Ryan Stillions en 2014 [1].

DML Define una estructura por niveles basada en conceptos militares tradicionales, tales como la estrategia, la tácticas, técnicas o los procedimientos, siempre por debajo del objetivo definido. Sobre este modelo se refleja la metodología de ataque CAT, que, del mismo modo desarrollará tácticas, técnicas y procedimientos en cada una de sus fases, desarrolladas en paralelo, introduciendo un nuevo concepto de matriz de ataque en lugar de los modelos tradicionales, lineales y centrados en el concepto de cadena de ataque (*Chain*), donde erróneamente se creía que interferir en uno de sus eslabones (fases) podría parar el ataque, algo que se ha demostrado ineficaz a día de hoy donde se ha generalizado las operaciones en equipo, ejecutadas por roles y con múltiples grupos coordinados de forma simultánea. (Fig. 1)

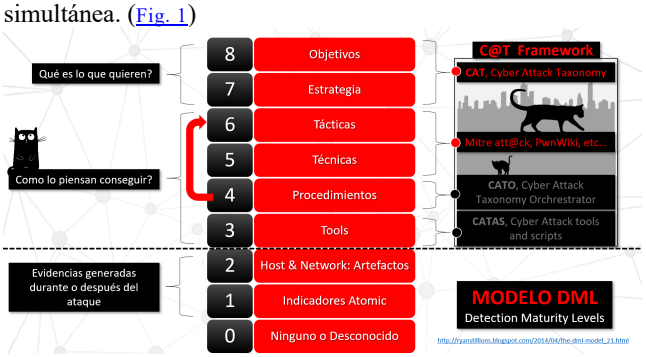


Fig. 1. Metodología CAT sobre el modelo DML

CAT es, por lo tanto, una taxonomía de ciberataque que representa la capa siete del modelo DML, únicamente para su nivel estratégico. La metodología CAT abarca desde el nivel tres DML hasta el nivel siete. En este documento solo será abordado el desarrollo de la taxonomía estratégica capa siete DML.

El modelo estratégico CAT permitirá la representación de ciberataques, simplificará la comprensión y diseño de los mecanismos de protección, estandarizará la interpretación del "*modus openrandi*" de los actores implicados en estos ataques o permitirá reproducir fielmente las acciones de los atacantes mediante tácticas y técnicas de simulación de adversarios, así como la propuesta de contramedidas mucho más eficaces, con el objetivo de verificar los sistemas de seguridad, permitiendo dar soporte en el análisis efectivo de proyectos que incorporen la seguridad por diseño (*Security by design*) o la privacidad por diseño (*Privacy by design*), en base a las recomendaciones de la Comisión Europea y fundamentada en sus propios estudios [4].

En lugar de rivalizar o competir entre las distintas partes interesadas que conforman el ecosistema de seguridad sobre la población, es necesario desarrollar métodos de análisis más eficaces y trabajar en la unificación de criterios para la comunicación de incidentes de seguridad, así como establecer las contramedidas más efectivas contra unos ciberataques que producen cada vez mayor impacto en la sociedad.

Por otro lado, el riesgo está creciendo exponencialmente, entre otros motivos, por la exposición de millones de dispositivos o la democratización del ciber armamento a través de internet, la

Deep web [5] o *Dark web* [6].
Ante un panorama tan complejo es necesario dar un paso más en la evolución para el análisis y representación de los ciberataques, unificando los criterios de todas las partes implicadas: cuerpos y fuerzas de seguridad, los afectados, las empresas de ciberseguridad y los organismos públicos.
Entre los modelos de representación estratégica de ciberataques más utilizados a nivel mundial está el conocido como *Cyber Kill Chain* del fabricante de armas americano Lockheed Martin [7]. Este modelo es muy cuestionado y posee multitud de modelos alternativos. Como referencia puede analizarse de forma clara en el documento *Unified Cyber Kill Chain,* de Paul Pols [8] que genera un mar de dudas sobre la validez con respecto al modelo inicial y donde su mayor fuerza para mantenerse vivo en la actualidad es la de haber sido pionero, pero ese argumento no tiene base científica.
Entre los modelos alternativos que cuestionan la eficacia del modelo de Lockheed Martin, encontramos varios ejemplos como el modelo de Marc Laliberte [9], el propuesto como *kill chain* 3.0 y definido por Corey Nachreiner [10], el modelo del profesor de la Universidad de Kansas, Blake D. Bryant [11] o el presentado en la Black Hat de 2016 por el autor Seant T Malone [12].
Las causas que justifican la presentación de estos nuevos modelos basados en el modelo original, son, por ejemplo, el cuestionamiento y necesidad de la controvertida fase dos o de *Weaponization*, considerada superflua, ya que no puede ser utilizada para plantear medidas defensivas al desarrollarse plenamente en el lado del atacante; también se justifica la incorporación de los movimientos laterales como fase necesaria, por su generalización, para representar los ataques actuales o la dificultad para representar los ataques internos.
Algunos de los estudios de estos autores proceden además del análisis de expertos, como Patrick Reidy [13] que, en el año 2013, ya planteaba las carencias del modelo de Lockheed Martin en los ataques iniciados desde el interior de la organización por los conocidos como Insiders en su ponencia "Combating the Insider Threat at the FBI" en la conferencia Black Hat USA. Otro experto, Giora Engel [14], en su artículo "Deconstructing the *Cyber Kill Chain*", publicado en Dark Reading 2014, ya presentaba que el modelo conocido como *kill chain* estaba únicamente orientado a la prevención de ataques de malware y pensando solo en el perímetro. Otra referencia importante es el experto Matt Devost [15], con su artículo "In today's environment, every cyber attacker is a potential insider", OODA Loop 2015.
Hay otras referencias mucho más recientes, pero las mencionadas en este artículo son muy significativas puesto que el modelo sigue vigente con críticas muy bien argumentadas por expertos desde hace años.
Por otro lado, el profundo análisis de las tácticas utilizadas por los atacantes, así como las técnicas y procedimientos, resumido con el acrónimo *TTP´s*, es base de estudio por parte de organizaciones y empresas. En este momento, el testigo lo ha tomado Mitre, y mediante su modelo *Mitre Att@ck* [16], define un compendio de tácticas y técnicas de un modo muy sólido.
Sin embargo, en el análisis de *Mitre att@ck* se detectan algunas carencias en la categorización estratégica por fases que permita representar gráficamente y de forma estructurada la organización de un ciberataque, por ese motivo, en muchos análisis existentes se distribuyen las tácticas y técnicas de Mitre en el modelo estratégico conocido como *Cyber kill Chain* sin tener en cuenta las carencias de este último.
Existen también otros modelos para representar las acciones del atacante, como el modelo de diamante [17] o las representaciones con redes de Petri [18], sin embargo, estos modelos también son referenciados continuamente al modelo *Cyber kill Chain* [19] o incluso planteados como rivales [20] por falta de definición de capas estratégicas en sus estudios, obteniendo resultados menos precisos de lo que supondría mediante su aplicación al modelo CAT.
Motivado por todo este estudio se presenta la creación de un nuevo modelo estratégico de alto nivel, que permitirá organizar de forma inclusiva las tácticas y técnicas procedentes de cualquier otro modelo conocido. Permitiendo analizar, representar y compartir así cualquier tipo de ciberataque, desde los más simples, pasando por los ataques dirigidos e incluso los desarrollados en entornos industriales. De esta manera, se presenta una potente herramienta para el desarrollo preciso, razonado y unificado de todo tipo de ciberataques.

II. ¿Por qué CAT?

Cuando se plantea la necesidad de representar un ciberataque, ya sea en entornos simulados tales como servicios de Red Team o bien para ataques reales actuales y con el objetivo de establecer modelos de defensa más efectivos, es necesario abordar las tácticas, técnicas, procedimientos y herramientas que se utilizan, pero sin olvidar también la representación del ataque de inicio a fin, detallando cada uno de los pasos y acciones que se ejecutan, así como el orden de las mismas, para lograr los objetivos finales del ataque de modo que facilite su comprensión. Es decir, es necesario disponer de una estructura organizativa basada en estrategias de ataque similares a los conceptos militares, y que permita su fácil comprensión por todas las partes involucradas.
En este sentido, el modelo más utilizado en la actualidad es el conocido como Mitre Att@ck, el cual aporta un conjunto de tácticas y técnicas usadas por atacantes, pero no plantea su organización en capas superiores de base estratégica. Por otro lado, existe otro modelo ampliamente utilizado desde hace muchos años, mencionado anteriormente y denominado Cyber *Kill Chain*, el cual si constituye un modelo de estrategia de ataque que representa la consecución de los pasos de inicio a fin de un ataque, pero carece de integración con otros modelos tácticos. De hecho, la mayoría de los estudios y modelos defensivos que desean estructurar la estrategia de ataque utilizan *Kill Chain* y, en muchos casos, organizados con las tácticas de Mitre.
Independientemente de los múltiples estudios de deficiencias antes mencionados en este trabajo relativos a *Kill Chain*, nos encontramos con más inconvenientes que nos llevan a descartarlo como taxonomía de ataque estratégica válida, y proponer una nueva taxonomía que represente la forma en que se desarrollan los ataques actuales.
Kill Chain como estrategia, parte del concepto en el cual un ataque se ejecuta de forma lineal y en cadena, y por lo tanto, según el propio concepto de *chain*, aplicando medidas de protección en cada fase (eslabón) de la cadena se puede cortar el ataque con contramedidas por nivel. La representación de pasos es lineal desde el inicio del ataque hasta el final o hasta alcanzar alguna fase intermedia, sin embargo, no permite retornos, bucles o saltos entre las fases. La realidad de los ataques a día de hoy es bastante más compleja, muchos ataques se ejecutan con fases en paralelo, coordinados con varios equipos especializados y con varias vías de entrada para llegar

al objetivo.
Un ataque puede comenzar con un infiltrado interno (*insider*) que compromete alguno de los sistemas atacando desde el interior de las infraestructuras de la víctima, o bien, dejando una puerta trasera C&C en la escena del ataque y dar paso posteriormente a un ataque desde el exterior. Este modelo de entrada con *insiders*, ni siquiera fue considerado cuando se creó el modelo estratégico *Kill Chain*.
Otro punto de entrada bastante común en la actualidad son las redes de corto alcance. En este caso, un atacante que se encuentra en las proximidades de la víctima, mediante creación entre otros de puntos de accesos falsos (Rogue AP), ataques de vulnerabilidades en sistema bluetooth, dispositivos de radio frecuencia o vulnerabilidades en los entornos IoT utilizados; ejecuta su objetivo final o establece un sistema de C&C y sale del campo de acción para dar paso a la ejecución remota del ataque. Volviendo a la analogía entre CAT y *Kill Chain*, en esta ocasión, el punto de entrada del ataque no sería correcto hablar de la fase tres de "*Delivery"* sino de un compromiso abordado desde un punto de entrada de proximidad.
Todo este análisis nos lleva a la aparición de una fase Compromiso (fase 2 CAT) que se aborda con diferentes vectores de entrada.
Estos vectores de compromiso precisan de un pormenorizado análisis inicial denominado "Perfil del objetivo" (fase 1 CAT) con tres vertientes fundamentales a estudiar, las personas para hacer una correcta selección del método de ingeniería social , las tecnologías para decidir las posibles vías de entrada por ejemplo remoto o próximo y por último, los procesos que permitirá seleccionar opciones más interesantes de entrada basándose en la actividad o el impacto en la operativa del negocio del objetivo (por ejemplo: ataque a través de proveedores que impactan sobre la cadena de suministro).
Aunque el perfil del objetivo representado en CAT puede asemejarse a la fase uno denominada "*Reconnaissance*" del modelo *Kill Chain*, se presenta de forma generalista y sin especificar cuáles son sus líneas de análisis. CAT presenta un modelo organizado en tres bloques: personas, procesos, y tecnologías, como base de fundamento necesaria para identificar la exposición del objetivo y provocar el mayor impacto posible.
Por otra parte, nos encontramos también con la necesidad de modelar ataques de descredito que generalmente son desarrollados por grupos sin ánimo de lucro, cuyo objetivo es hacer públicas las vulnerabilidades que las organizaciones mantienen expuestas en internet y en otras ocasiones, ataques DoS provocados por grupos *hacktivistas,* para descrédito de las organizaciones, impactando sobre su imagen o reputación del negocio. Estos modelos de ataque dificultan su representación en taxonomías basadas en *chain*, ya que se pasa desde el proceso de selección y reconocimiento del objetivo a la ejecución del ataque, sin necesidad de fases como "*Weponization*", "*Delivery*", y mucho menos "*Installation*", esto corrobora que los ataques no funcionan en cadena y que las contramedidas de niveles intermedios no impiden alcanzar las fases finales del objetivo de ataque.
La necesidad de representar los modelos anteriormente expuestos queda claramente reflejada con la incorporación reciente de su nueva táctica denominada "Impact" en el modelo conocido como Mitre Att@ck (abril 2019), donde se representan un listado de técnicas sin criterio organizativo claro, que van desde los métodos de ejecución de ataque hasta sus resultados finales, utilizando además un nombre ambiguo "impact" donde presumiblemente se clasifiquen todo tipo de técnicas debido a un modelo estratégico inexistente. Cabe destacar que CAT aborda estas necesidades y muchas otras, en un tiempo anterior a la publicación de Mitre.
CAT además de representar estas fases, en su modelo en bucle, con ejecución de objetivo colindante al resto de fases, permite pasar de una fase a otra sin necesidad de ejecutar todas en orden y pudiendo volver hacia alguna de ellas en cualquier momento del ataque.

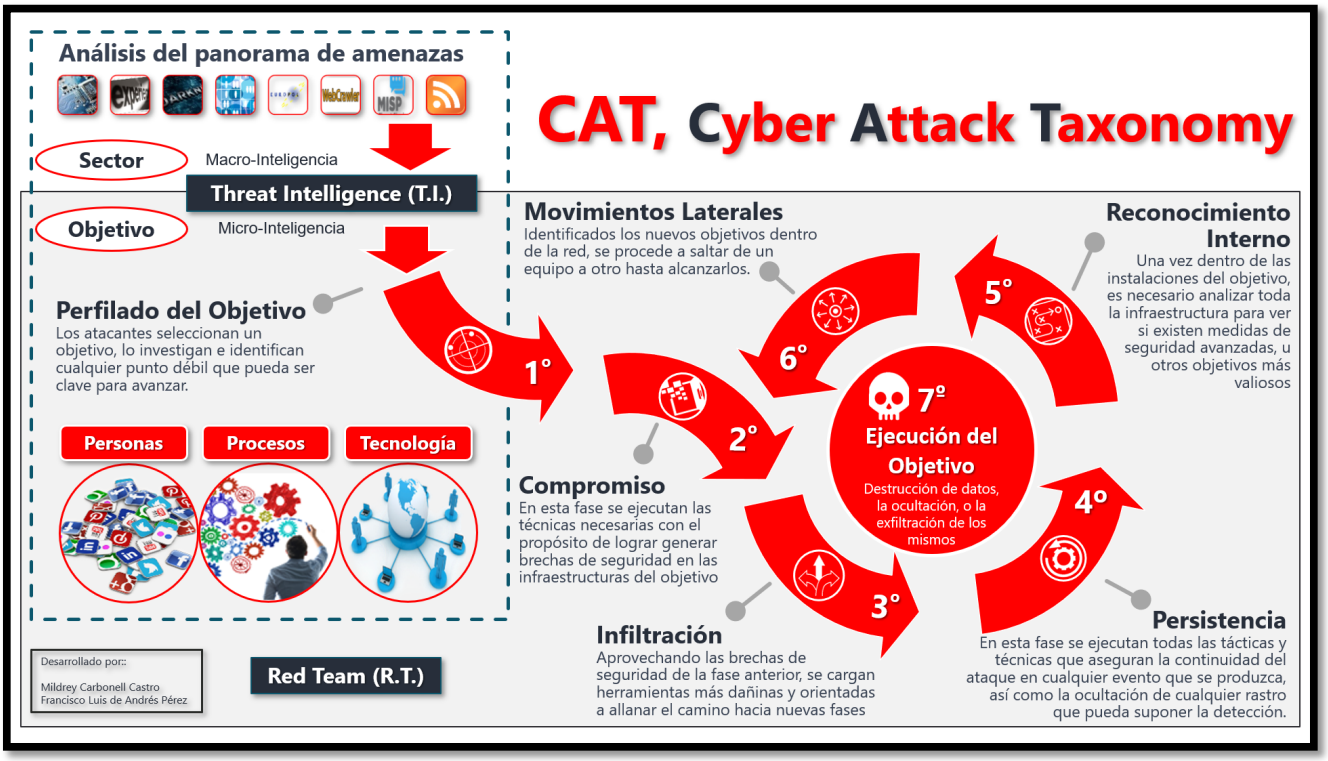


Fig. 2. Taxonomía de ataque CAT

La única fase que siempre será necesario ejecutar es compromiso, a partir de ese momento el ataque puede culminar (ir a fase 7. Ejecución de objetivos) o desarrollar el resto de las fases o alguna de ellas de forma indiscriminada.
Este nuevo modelo que se ha definido como CAT- Cyber Attack Taxonomy- será descrito en la siguiente sección.

III. Modelo CAT – Estrategia de ciberataque

La taxonomía CAT modela el ciclo de un ciberataque de principio a fin. Su representación se fundamenta en el nivel estratégico, que se sitúa en las capas más altas de abstracción.
El modelo CAT se representa en siete fases (Fig. 2) y contempla la ejecución parcial o completa de cada una de ellas, así como los ciclos recurrentes para impacto sobre múltiples sistemas mediante movimientos laterales:

1. Perfilado del Objetivo
2. Compromiso del Objetivo
3. Infiltración
4. Persistencia
5. Reconocimiento Interno
6. Movimientos Laterales
7. Ejecución de objetivos

IV. ¿Qué representa cada una de las fases CAT?

CAT pretende modelar la forma en que se organiza y piensa un atacante. Por este motivo, las fases siguen un orden lógico. Según el tipo de ataque se pueden excluir o repetir fases, pasar desde cualquiera de ellas hacia la fase final de ejecución de objetivos o realizar un número indeterminado de ciclos combinando fases hasta llegar a la fase final. Este argumento lo diferencia de forma clara de cualquier otro modelo basado en *Kill Chain*, ya que estos proponen como fundamento que el bloque sobre la fase en ejecución neutraliza al atacante e impide que se pueda seguir ejecutando en fases superiores.
Sin embargo, el modelo CAT sugiere que estos ataques se producen de un modo más sofisticado, en ocasiones con fases en paralelo sobre múltiples sistemas y de forma simultánea.
Así podemos decir de forma clara que CAT no es un modelo más basado en *Kill Chain*, sino que revoluciona cualquier modelo previo basado en esta idea. Con el objetivo de comprender el modelo CAT en detalle, a continuación, se describen cada una de sus fases.

Fase 1: Perfilado del Objetivo

Representa la fase inicial del análisis y recogida de información del objetivo a atacar. Pretende hacer un mapa de exposición del objetivo a atacar.
Tras culminar esta fase, el atacante conocerá con precisión el sujeto atacar, las formas de aproximación y los caminos alternativos para lograr su objetivo final

Esta fase se fundamenta en dos tipos de análisis:

- ***Análisis Macro o PESTLE*** [21]***:*** consiste en el estudio político, económico, social, tecnológico, legal y de entorno (*PESTLE*) asociado al objetivo. Enumera las amenazas actuales a las que está sometido, el sector a atacar o las localizaciones geográficas donde se encuentra ubicada entre otros.
  Este estudio identificará aquellos ataques más probables a los que está sometido, aquellos que puedan tener mayor impacto y trazará un mapa de posibles ataques futuros con probabilidad de éxito.
- ***Análisis Micro:*** centrado exclusivamente en el análisis de exposición del objetivo. Se analizan los tres factores fundamentales de exposición: las personas, los procesos y las tecnologías.
  - *Personas:* análisis de toda la información relativa a las personas relacionadas directamente o indirectamente con la empresa. Datos confidenciales o privados que puedan estar expuestos.
    - Información de los empleados que pueda ser utilizada.
    - Empleados críticos para la empresa.
    - Datos de proveedores relacionados que puedan aportar una vía de entrada.
    - Análisis de posibles servicios a usuarios vulnerables, tales como:
      - Centros de atención al usuario
      - Recursos Humanos
      - Marketing

El análisis de estos datos y entornos facilitan, en muchas ocasiones, ataques de phishing o ingeniería social entre otros.

  - *Procesos:* análisis de todos los procesos centrales del negocio, su cadena logística para distribución, mecanismos y procedimientos de funcionamiento, los proveedores de materias primas o servicios esenciales para el negocio entre muchos otros. Este estudio, basado en fuentes abiertas, puede orientar el ataque hacia objetivos secundarios que provoquen mayor impacto o más vulnerables que el objetivo principal. (por ejemplo, técnicas de watering hole, secuestros de servidores DNS, etc.)
  - *Tecnologías:* análisis de los distintos perímetros de riesgos tecnológicos que posee el objetivo, tales como:
    - Servicios expuestos en internet y bajo qué modelo:
    - Alojamiento interno
    - Cloud público, privado
    - Modelos compartidos, etc.
    - Análisis general de la infraestructura: ya sea por pruebas más o menos intrusivas como el escaneo de vulnerabilidades o por búsqueda pasiva de datos expuestos en la red.
    - Información pública disponible:
      - Tecnologías implementadas
      - Registros de dominios
      - Presencia de información técnica en medios sociales o foros entre otros.
    - Localización geográfica y tecnología desplegada en la misma: red cableada, wifi, IoT, entornos industriales, redes de corto alcance, telefonía, etc.
      El resultado permitirá modelar escenarios de ataque, no solo por exposición de perímetros en internet sino por proximidad a sus instalaciones, o si se debe recurrir a modelos más agresivos dentro de sus instalaciones. Permitirá también modelar las siguientes fases de su estrategia con mayor probabilidad de éxito.

Fase 2: Compromiso

Constituye el primer movimiento o ataque necesario para abrir una vía de acceso en el objetivo final. Es el momento en que el atacante ejecuta una acción directa contra su objetivo para vulnerar el sistema o activo seleccionado, como el lanzamiento de un correo *phishing* o la ejecución de código para explotar una vulnerabilidad entre otros.

Esta fase puede constituir también el inicio y el fin de un ataque, pasando directamente desde la fase de compromiso a la fase de ejecución del objetivo, como es el caso de los ataques *DoS,* donde la disponibilidad del sistema es comprometida (objetivo final) pero el atacante no tiene ningún interés en acceder a los sistemas de su víctima (no se infiltra).

También puede representar ataques de inyección de código, por ejemplo, inyección de *SQL* donde obtenemos información confidencial o privada de sistemas internos, tales como las bases de datos de una web, con el objetivo final de hacer públicos sus datos y desacreditar así a la organización.

Durante los años 2017 y 2018 estos tipos de ataques han aumentado de forma exponencial generalmente motivados por temas políticos o para el descrédito en la imagen de sus víctimas y pérdida de confianza de los afectados por la fuga de su información personal.

Los grupos cibercriminales han definido nuevos modelos de negocio, facilitando mediante modalidad de alquiler por uso a través de internet, el acceso a sistemas de ataque, conformados por miles de equipos infectados previamente con software malicioso y obedecen nuestras ordenes de forma unificada, son las conocidas como redes de *zombies o bots*.

Esta fase se puede ejecutar o representar mediante varios **vectores de aproximación,** basados fundamentalmente en el estudio de la fase previa CAT de perfilado del objetivo: Ataques de proximidad, Ataques de red y Ataques sociales.

- Vector 1, ataques por proximidad o internos: son todos aquellos ataques en los que está involucrada la tecnología o infraestructura del cliente y que solo se puede acceder a ella estando ubicado muy cerca o dentro de sus instalaciones.

Cuando hablamos de proximidad podemos representar ataques donde el atacante puede asumir un cierto riesgo, tales como ataques a la infraestructura wifi, internet de las cosas, bluetooth, radio frecuencia, *NFC*, por citar algunos ejemplos. El ataque a este tipo de redes se puede realizar en muchas ocasiones desde las inmediaciones del entorno, incluso se podrían abordar opciones de acercamientos con nuevas tecnologías combinando dispositivos físicos, como drones o robots, con otros sistemas ciber, generando así ataques ciber-físicos muy sofisticados.

En estos ataques el atacante deberá estar dentro de sus instalaciones para dañar la infraestructura cableada, u otros entornos, asumiendo en este caso un riesgo mayor.

- Vector 2, ataques de red: son todos aquellos ataques perimetrales realizados sobre la infraestructura del cliente de forma remota, impactando sobre los servicios públicos de la red, tanto en las zonas expuestas, la *DMZ*'s (*Demilitarized Zone*), como otros servicios ubicados en la nube. Dentro de este modelo también están incluidos los ataques sobre aplicaciones móviles.
- Vector 3, ataque social (a las personas): son todos aquellos ataques dirigidos a las personas como por ejemplo los ataques de spear phishing, la ingeniería social por red, la ingeniería social por cercanía (*USB* maliciosos, códigos *QR*) o los ataques sociales basados en la empatía, la persuasión o la confianza entre otros.

Un ataque puede hacer uso de uno o varios vectores de aproximación, así como utilizar los resultados de uno para poder ejecutar u avanzar por otros. Por ejemplo, un ataque de suplantación wifi donde se puedan obtener credenciales internas válidas y que luego permitan acceder de forma remota a servicios expuestos a internet para comenzar un ataque mayor (con menor exposición del atacante) hacia el interior de la red, saltando así a otras fases.

Fase 3: Infiltración

Esta fase depende del éxito de la anterior, por lo tanto, una vez conseguido el compromiso inicial, el atacante se hace con el control del activo y se establece en el mismo para comenzar a atacar hacia el interior. Se asocia a el lanzamiento de *payloads*, instalación de troyanos, apertura de *backdoors*, etc.

De igual forma que en las anteriores, esta fase puede ser el final de un ataque, dañando la confidencialidad, integridad o disponibilidad del activo, como ejemplos, la modificación de una web, la modificación de datos de una BBDD, el reinicio de los sistemas, los cambios de configuraciones que generan un mal funcionamiento, etc.

No obstante, lo más interesante de esta fase es el uso de esta infiltración para continuar hacia las siguientes fases de un ataque, posibilitando al atacante el movimiento por la zona donde se encuentra el activo comprometido e incluso el salto, aplicando o no, técnicas de persistencia.

Fase 4: Persistencia

Una vez comprometido uno o varios activos, el atacante ejecuta técnicas para la ocultación de su rastro ante los sistemas de protección, así como para mantener el compromiso de los sistemas ante cualquier cambio de estado tales como el reinicio o actualizaciones entre otros, desarrollando así un ataque lo más resiliente posible.

Realiza entre otros, cambios de configuraciones, borrado de todas las pruebas posibles que generen alerta o faciliten la investigación, como logs, etc., aplica técnicas de esteganografía para la ocultación de información, despliega las herramientas de *C&C* (Sistema utilizado por grupos de cibercriminales para mantener un acceso y control remoto sobre el objetivo comprometido a través de internet), crea usuarios privilegiados o se enmascara dentro del sistema entre muchas otras acciones.

Fase 5: Reconocimiento interno

En esta fase el atacante analiza con el máximo detalle posible el mayor número de sistemas de la red interna, tales como dispositivos de protección, detección etc. Se ejecutan técnicas para el descubrimiento de dispositivos y su ubicación en un mapa de red, se análiza y captura de tráfico de red para detección de protocolos, datos, o imágenes entre otros, y todo esto de cara a identificar nuevos objetivos de interés dentro de la misma infraestructura.

Puede ser también la fase final en el caso de modelos de ataque, cuyo objetivo sea obtener información y publicarla, venderla en el mercado negro o extorsión, aunque su utilidad para continuar el ataque hacia fase de movimientos laterales es mucho más interesante.

Fase 6: Movimientos laterales

En base a los resultados obtenidos en la fase cinco o de reconocimiento interno, el atacante, planifica nuevos objetivos detectados y comienza nuevos ciclos de ataque, partiendo desde fases previas, hasta lograr el máximo impacto en cada uno de los sistemas, así pues, el atacante va pasando de uno a otro hasta conseguir el objetivo planificado.

Fase 7: Ejecución del objetivo

La finalidad de todo ataque es provocar el mayor impacto en la victima para beneficio propio o de terceros. En esta fase se ejecuta el objetivo, que varía dependiendo de muchos factores: Impacto en la población, Impacto reputacional, Fuga o extorsión de la información, Robo y otros tipos de impactos económicos, Ciberterrorismo y Espionaje industrial.

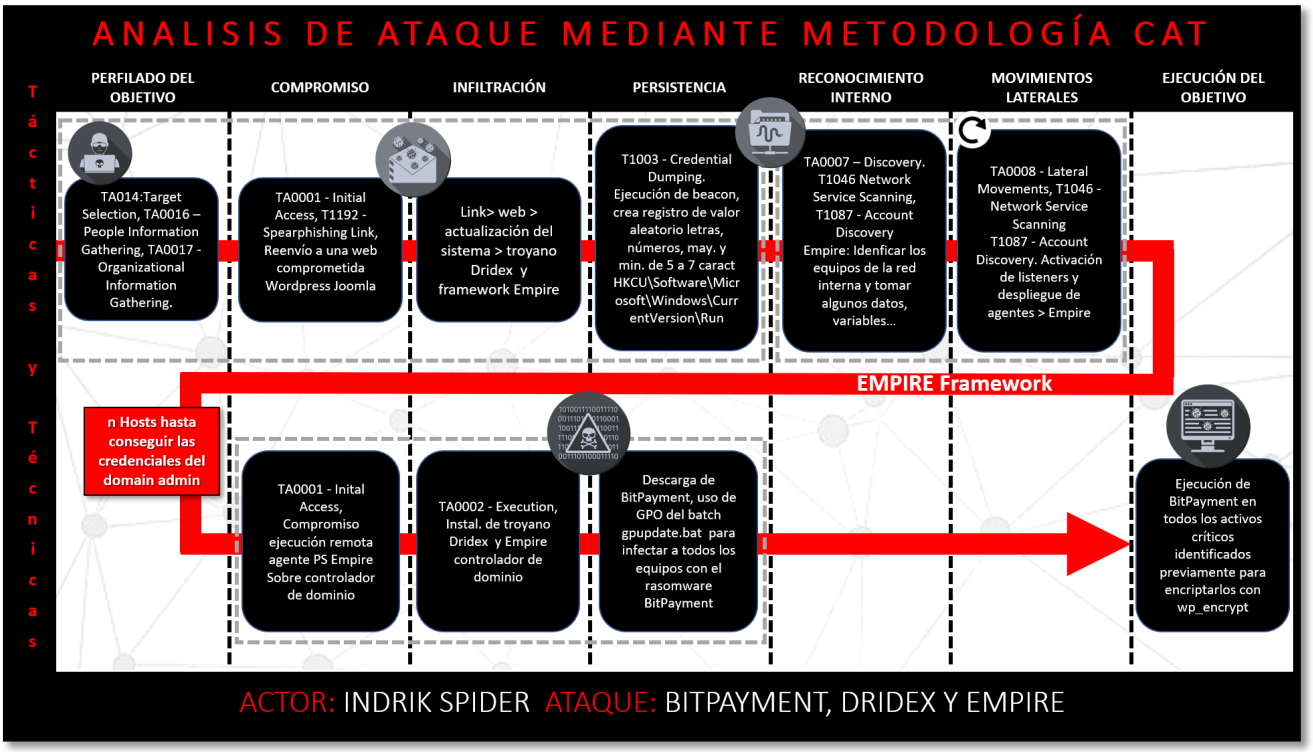


Fig 3. Análisis CAT de ataque Indrick Spider

V. Ventajas del modelo

La taxonomía de ataque CAT, permite representar ataques que quedan fuera del alcance de los modelos más utilizados hoy en día como el de los *insiders*, también elimina el falso argumento acerca de la cadena de ataque o *Kill Chain* introduciendo un modelo cíclico y basado en el análisis de inteligencia e incorpora la clasificación de inteligencia macro y micro completamente alineados con las especificaciones de marcos internacionales, como el descrito en el *Framework Tiber*-EU para el sector bancario europeo [22] .

Define una taxonomía nueva de siete fases al igual que muchas de sus predecesoras, que, por su sencillez, supuso uno de los motivos para el sostenimiento en el tiempo de estos modelos mermados e introduciendo también conceptos novedosos como el de micro inteligencia y macro inteligencia o la clasificación de tipología de ataque como por ejemplo los ataques de proximidad.

CAT empasta con otros modelos como el de *Mitre Att@ck* o el de Pwnwiki [23] para definir tácticas, técnicas, procedimientos y herramientas que le confieren una capacidad mayor de implementación práctica.

Además de dar soporte al análisis de los diferentes ataques a los que podemos estar sometidos, se convierte en una estrategia para estructurar modelos de servicios Red Team y facilita una forma de organizar las herramientas que incorporen la inteligencia artificial para posibles desarrollos automatizados de ataques, facilitando la toma de decisiones y la representación gráfica del ataque.

A continuación, se representa la simulación mediante el modelo CAT de un ataque wifi con el objetivo de identificar usuarios y comprometer la red interna (Fig. 3): Ataque Rouge AP y MITM para compromiso interno.

VI. Conclusiones

Son muchos los ataques que a día de hoy se realizan de forma exitosa y son muchos los grupos organizados que trabajan de forma coordinada modelando bien sus estrategias para llegar a los objetivos finales con el menor coste posible y el menor riesgo de ser descubierto. En ese sentido, es importante crear un lenguaje común para modelar, compartir información sobre ataques y proponer medidas defensivas adecuadas.

La taxonomía de ataque CAT presenta una capa de estrategia de alto nivel de siete fases según se organiza y ejecuta un ciberataque, tomando en cuenta la paralelización de las iniciativas de los atacantes, así como la posibilidad de despliegue de múltiples tácticas por cada nivel estratégico. También pretende demostrar las capacidades para la representación de ataques no reproducibles bajo las taxonomías previas como son los ataques de compromiso a la disponibilidad, los ataques *DoS* y *DDoS*, los ataques internos o los ataques sobre entornos industriales.

Por último, pero no menos importante, destaca también su utilidad en el modelado de ejercicios de Red Team, así como la representación de arquitecturas de defensa, dejando además abiertas varias líneas de investigación para el futuro como el desarrollo de sistemas de modelado y orquestación de ataques, distribuciones software de ataque estructuradas según la metodología CAT, o adaptaciones para entornos específicos como el industrial o internet de las cosas.

Referencias

[1] Ryan Stillions: "DML Model", posted in https://ryanstillions.blogspot.com/, 22 april, 2014.

[2] Wikipedia: "WannaCry Malware", https://es.wikipedia.org/wiki/WannaCry

[3] Wikipedia: "Petya Malware", https://es.wikipedia.org/wiki/Petya(malware)

[4] Alex Rayon: "Privacy y Security by design: ¿qué son y por qué son relevantes?", Blog Universidad de Deusto, Artículo sobre la seguridad y privacidad por diseño en el marco de las recomendaciones de la Comisión Europea https://blogs.deusto.es/bigdata/privacy-y-security-by-design-que-son-y-por-que-son-relevantes/ , 22 enero, 2016.

[5] Wikipedia: "DeepWeb Wikipedia", en https://es.wikipedia.org/wiki/Internet_profunda

[6] Wikipedia: "DarkWeb Wikipedia". https://es.wikipedia.org/wiki/Dark_web

[7] Lockheed Martin: "Taxonomía de ataque desarrollada por Lockheed" Martin, https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, 2015

[8] Paul Pols: "Unified Cyber Kill Chain", public in Cyber Security Academic https://www.csacademy.nl/images/scripties/2018/Paul-Pols---The-Unified-Kill-Chain.pdf, 7 december, 2017

[9] Marc Laliberte: "A Twist On The Cyber Kill Chain: Defending Against A JavaScript Malware Attack", online darkreading https://www.darkreading.com/attacks-breaches/a-twist-on-the-cyber-kill-chain-defending-against-a-javascript-malware-attack/a/d-id/1326952 , 21 sept, 2016.

[10] Corey Nachreiner: "Kill Chain 3.0: actualización de la cadena de "Cyber Kill" para una mejor defensa", online cilicon https://www.silicon.es/experto-opinion/kill-chain-3-0-actualizacion-de-la-cadena-de-cyber-kill-para-una-mejor-defensa, 17 feb, 2015

[11] Bryant, Blake, Saiedian, Hossein: "A novel kill-chain framework for remote security log analysis with SIEM software" https://www.researchgate.net/publication/314782193_A_novel_kill-chain_framework_for_remote_security_log_analysis_with_SIEM_software , Computers & Security. 67. 10.1016/j.cose.2017.03.003, 2017.

[12] Sean T Malone: "Using an expanded cyber kill chain model to increase attack resiliency" presentado en la Black Hat 2016 https://www.blackhat.com/docs/us-16/materials/us-16-Malone-Using-An-Expanded-Cyber-Kill-Chain-Model-To-Increase-Attack-Resiliency.pdf, 2016.

[13] Patrick Reidy: "Combating the Insider Threat at the FBI", Black Hat. https://media.blackhat.com/us-13/US-13-Reidy-Combating-the-Insider-Threat-At-The-FBI-Slides.pdf , USA 2013

[14] Giora Engel: "Deconstructing The *Cyber Kill Chain"*, en la web Dark Readinghttps://www.darkreading.com/attacks-breaches/deconstructing-the-cyber-kill-chain/a/d-id/1317542, 18 November, 2014.

[15] Matt Devost: "Every Cyber Attacker is an Insider", en la web OODA Loop https://www.oodaloop.com/cyber/2015/02/19/every-cyber-attacker-insider/, 19 feb, 2015

[16] Mitre Corporation "Mitre Att@ck". https://attack.mitre.org/ , 2018.

[17] Pendergast, A., Caltagirone,S: "The Diamond Model of Intrusion Analysis" public on http://www.activeresponse.org/wp-content/uploads/2013/07/diamond.pdf, 2013.

[18] Jasiul, Bartosz & Szpyrka, Marcin & Śliwa, Joanna: "Detection and Modeling of Cyber Attacks with Petri Nets. Entropy", online https://www.researchgate.net/publication/269872882_Detection_and_Modeling_of_Cyber_Attacks_with_Petri_Nets DOI: 10.3390/e16126602, december 2014.

[19] Levent Ertaul, Mina Mousa: "Applying the Kill Chain and Diamond Models to Microsoft Advanced Threat Analytics", Int'l Conf. Security and Management of California State University East Bay https://csce.ucmss.com/cr/books/2018/LFS/CSREA2018/SAM9723.pdf , Hayward, CA, USA, SAM 2018.

[20] PWC: "Diamonds or chains", desarrollado por la empresa Pwc, en su sitio web https://pwc.blogs.com/cyber_security_updates/2015/05/diamonds-or-chains.html, 29 may, 2015.

[21] Wikipedia: " Análisis PESTLE o PESTEL", https://es.wikipedia.org/wiki/An%C3%A1lisis_PESTEL

[22] European Central Bank: Tiber-EU framework: "How to implement the European framework for Threat Intelligence-based Ethical Red Teaming". May 2018. https://www.ecb.europa.eu/pub/pdf/other/ecb.tiber_eu_framework.en.pdf

[23 PwnWiki: "Collection of TTP", online copy http://pwnwiki.io/#!index.md,

[24] Francisco Luis de Andres., Mildrey Carbonell: "Estrategia CAT. Desarrollo metodológico", presentación interna s21sec, noviembre 2018.

Sistema de Cálculo de Riesgo Dinámico en Dominios Administrativos Basado en Ontologías

Fernando Monje[1], Cristina Galván[1], Raúl Riesco[2], Víctor A. Villagrá[1]
[1]Universidad Politécnica de Madrid (UPM), DIT. ETSI Telecomunicación. Avda. Complutense 30, 28040, Madrid
[2]Instituto Nacional de Ciberseguridad (INCIBE), Avda. José Aguado, 41, 24005 León
f.monjer@alumnos.upm.es, cristina.galvan.prieto@alumnos.upm.es, raul.riesco@incibe.es, victor.villagra@upm.es

Resumen- **Con la creciente complejidad de las ciberamenazas, se hace necesario disponer de herramientas para conocer el contexto cambiante en tiempo real. En este documento se va a presentar una arquitectura y un prototipo diseñados para modelar el riesgo de dominios administrativos, ejemplarizándolo al caso de un país en tiempo real, concretamente el de España. Para llevar a cabo esta tarea se ha realizado un modelado de los activos y de las amenazas detectadas por diversas fuentes de información. Toda esta información se almacena como conocimiento haciendo uso de ontologías, que permita aplicar motores de razonamiento para así poder inferir nuevo conocimiento utilizable posteriormente en los siguientes razonamientos. Este modelado y razonamiento se ha enriquecido con un sistema dinámico de gestión de confianza de las distintas fuentes de información, y de capacidades de incremento de fiabilidad con la inclusión de información de inteligencia de amenazas adicional.**

***Index Terms*- riesgo dinámico, ontologías, métricas.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

El área de la gestión dinámica de riesgos trata de evolucionar los enfoques clásicos de análisis estáticos de riesgos, basadas en metodologías bien definidas tales como MAGERIT [1], que permiten una caracterización de los riesgos de una organización. Sin embargo, este enfoque es estático, es decir, se hace una instantánea de la situación de la organización en un momento dato, y a partir de ahí se generan las métricas y políticas de seguridad.

Sin embargo, actualmente la mayoría de parámetros que intervienen en un análisis de riesgos son dinámicos, cambian constantemente con el tiempo, por lo que los análisis de riesgos realizados pueden quedar obsoletos muy rápidamente.

La gestión dinámica de riesgos trata de adaptarse a estas circunstancias incluyendo las variaciones temporales de los elementos componentes de estos análisis. Por ello, es necesario caracterizar estos elementos y sus variaciones temporales, y generar las salidas adecuadas, muchas veces conocidas como Sistemas de Conciencia Situacional.

Un ejemplo de aplicación de estos sistemas es la generación de métricas para un cálculo de nivel de riesgo en un entorno global, como un país, una región o un sector, en el que tenemos la dinamicidad de las amenazas, pero los activos no están claramente definidos al tratarse de un dominio administrativo y tener únicamente estimaciones de sus activos.

Por ello, este artículo propone una solución a la problemática del cálculo del nivel de riesgo dinámico en estos entornos, cuando la definición de activos no se puede realizar con gran precisión.

Los principales objetivos que se quieren cumplir en este trabajo son:

- La obtención de un nivel de riesgo para un dominio administrativo (país, región, sector), mediante el modelado con ontologías, tanto de los eventos de amenazas, su enriquecimiento con inteligencia adicional, así como de los activos para posteriormente poder definir métricas expresadas como reglas de un motor de razonamiento.
- Lograr una definición de activos los más específica posible teniendo en cuenta la limitación debido a la definición borrosa del dominio administrativo y por tanto, de la cantidad de activos a analizar.
- Diseñar un sistema capaz de manejar varias fuentes de información mediante un sistema de gestión de confianza que permita realimentar el sistema generando nuevo conocimiento.
- Construir un sistema que modele asimismo la influencia del tiempo transcurrido desde la aparición de cada amenaza y su efecto en el nivel de riesgo, a la vez que sea capaz de representarlo visualmente con adecuados sistemas de visualización de gráficos.
- Por último, y más importante, es que todo ello se consiga realizar en tiempo real.

Por ello, en primer lugar, este artículo analizará el estado del arte en sistemas y aproximaciones similares, para validar su aplicabilidad al problema. Posteriormente, se define la arquitectura propuesta y se detalla el diseño, desarrollo y validación de un prototipo realizado en un entorno real.

II. Estado del Arte

Teniendo en cuenta que los sistemas de información cambian de manera continua, la necesidad de metodologías de análisis de riesgos dinámicas es innegable. El dinamismo de dichas metodologías permite a los sistemas variar el riesgo en tiempo real, lidiando con eventos a medida que se producen y la naturaleza cambiante de las amenazas. Para ello, resulta de gran utilidad el uso de ontologías que proporcionan un amplio vocabulario con el que expresar el conocimiento según distintos niveles de detalle, pudiendo así modelar cualquier dominio necesario. A continuación, se muestran diferentes propuestas en las cuales se aplican ontologías para el modelado de distintos dominios relacionados con nuestra propuesta, así como la generación dinámica de métricas.

Herzog et al. en [2] definen un dominio de seguridad genérico de ontologías especificado en OWL que cubre la mayoría de los aspectos de un dominio de seguridad de información. Proporciona un vocabulario detallado y soporta capacidades de razonamiento. Se proporcionan unas subclases detalladas y relaciones entre ellas.

Fenz, S. en [3] hacen una contribución a las ontologías mediante la definición de Medidas de Seguridad de IT. El autor planea alinearlo con los estándares de ISO 27004 y aplicarlo en escenarios de auditoría del mundo real, así como ir más allá en el grado de automatización.

Obrst, L. et al. en [4] introducen una ontología propuesta para la Ciberseguridad, especialmente como una extensión de MAEC (Caracterización de Atributos y Enumeración del Malware). Los autores usan como referencia el "Diamond Model of Malicious Activity".

Singapogu, S. et al. en [5] describen una ontología propuesta para la realización de la evaluación del riesgo empresarial, apoyando el proceso de análisis de riesgos de la seguridad de IT.

Erbacher, R.F. en [6] desarrolla una ontología centrada en paquetes llamada PACO, lo cual les permitía representar y capturar elementos atómicos de las redes de comunicación, es decir, paquetes y secuencias de paquetes. El modelo propuesto es una base para enfoques más holísticos.

Por otro lado, el área de Inteligencia de Amenazas (*Threat Intelligence*) engloba todo el conocimiento que se posee sobre las posibles amenazas para poder tomar decisiones adecuadas. Cuando la información compartida es técnica, cobran importancia los indicadores de compromiso (IoC). Sin embargo, los esquemas basados en IoC son poco eficientes debido a que se basan en firmas, las cuales son efímeras. Estándares como STIX™, TAXII™, CybOX™ comienzan a ganar fuerza de cara al intercambio de información de amenazas, debido a que proporcionan una estructura para indicadores de ciberseguridad, caracterización de amenazas y diferentes opciones para el intercambio de información. Analizar y compartir la información obtenida mediante Inteligencia de Amenazas de una manera efectiva requiere una representación común, estándares y protocolos para poder compartir, y un conocimiento común de los conceptos y la terminología relevante. Una solución muy adecuada para dicho enfoque es el uso de ontologías, a partir del cual los siguientes autores han realizado diversos estudios y enfoques.

Ekelhart, A. et al. en [7] realiza un aporte a las ontologías mediante la realización de un análisis de riesgo cuantitativo, y visualizan el daño producido por ciertas amenazas, el coste de corte y el tiempo de recuperación. La ejecución de la herramienta con salvaguardas adicionales muestra sus beneficios y ofrece datos objetivos para la toma de decisiones sobre qué salvaguardas implementar, y como evitar la instalación de contramedidas no económicas.

Vergara, JEL. et al. en [8] proponen un modelo basado en ontologías para compartir alertas entre diferentes Sistemas de Gestión de la Seguridad de la Información.

Syed, Z. et al. en [9] trabajaron en la integración entre STIX™ y las ontologías para la Conciencia Situacional, lo cual es un enfoque muy interesante. Los autores demuestran los beneficios para diferentes casos de uso (vulnerabilidades asociadas con lectores de PDFs, sugerencias de software similar, etc.) como una contribución muy interesante, por ejemplo, para la comprobación del impacto del cambio de proveedores.

Riesco R. et al. [10] propone un modelado del área de gestión dinámica de riesgos e inteligencia de amenazas sobre el que se pueden generar reglas de inferencia para su uso en distintas aplicaciones. La contribución realizada en este artículo se basa en esta propuesta, aplicándola al caso de estudio de cálculo de riesgo para dominios administrativos.

Por último, es importante comentar la importancia que tiene la utilización de las métricas en el ámbito de toma de decisiones de Ciberseguridad. El uso de dichas métricas permite saber la efectividad de los controles implementados, conociendo así información sobre los sistemas de información.

Fenz, S. et al. en [11] integra un concepto ontológico de la seguridad de la información en la gestión de procesos de negocio relacionados con el riesgo. La ontología está basada en el NIST y los autores proporcionan subontologías de amenazas, vulnerabilidades y control. La comunicación entre el servicio web de seguridad de ontologías y el motor de simulación dependiente del riesgo está realizada vía XML. Los autores proponer mejorar y extender la clasificación de amenazas para considerar la amenaza a la vida humana la prioridad máxima en caso de riesgo. Adicionalmente, la información complementaria considerada por la ontología puede proporcionar detalles valiosos y esenciales para la toma de decisiones.

Mateos, V. et al. en [12] proponen un Sistema Automático de Respuesta a Intrusiones (AIRS) basado en ontologías. El sistema infiere las respuestas óptimas a nivel de red.

Romero, I. et al. en [13] proponen una arquitectura dinámica que se basa en el uso de inteligencia (implementada a través de ontologías) para relacionar los conceptos involucrados en un análisis de riesgos: amenazas, activos, impacto y probabilidad; calculando esta última a partir de probabilidades condicionadas.

Todos los trabajos presentados hasta el momento tratan de desarrollar diferentes ontologías para representar conocimiento en ellas, o de proponer arquitecturas basadas en las ontologías, sin llegar a tratar muy profundamente el análisis dinámico de riesgos y su aplicación al entorno de dominios administrativos. En este trabajo, no solo se hace uso de las ontologías para representar el conocimiento, sino que mediante dicho conocimiento se realiza una gestión del riesgo a nivel de dominio administrativo, geográfico o sectorial. Además, se tienen en cuenta las amenazas pasadas a la hora del cálculo de riesgo mediante la realización de un histórico

de riesgos. Por último, para poder ofrecer un análisis de riesgo con mayor exactitud se realiza una realimentación continua, a diferencia del resto de trabajos propuestos, mediante el uso de la confianza de las diferentes fuentes.

III. Arquitectura

La arquitectura presentada se basa en el uso de ontologías. Sin embargo, a diferencia de otras propuestas mencionadas en el estado del arte, el sistema pretende lograr una fuerte realimentación teniendo en cuenta el contexto y los eventos pasados, todo ello, manejando un alto flujo de datos en tiempo real. Para crear un sistema de cálculo de riesgos que pueda inferir nueva información, primero es necesario identificar los elementos que intervienen un sistema de gestión dinámica de riesgos:

- **Identificación de activos:** Se identifican los activos relevantes del dominio administrativo.
- **Evaluación de activos:** Se otorga un valor cuantitativo y cualitativo a los activos identificados previamente.
- **Identificación de amenazas:** Se han de identificar las amenazas que pueden afectar a los activos.
- **Evaluación de amenazas:** Se han de parametrizar las amenazas, tanto su impacto sobre los activos como la probabilidad de que se materialicen.
- **Evaluación de riesgos:** Una vez evaluados todos los elementos anteriores se puede calcular un nivel de riesgo.

De esta forma, se podría seguir un análisis genérico de riesgos, como el que se puede observar en Fig. 1.

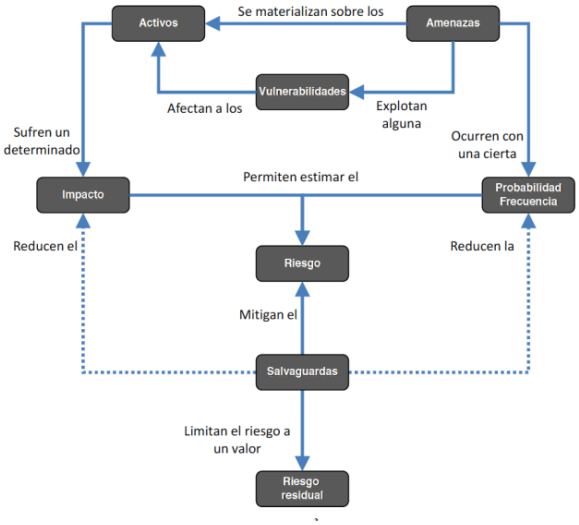


Fig. 1. Diagrama de análisis de riesgos.

Sin embargo, al querer calcular el riesgo de un dominio administrativo, se pone de relieve otra problemática: resulta de gran complejidad identificar y modelar en detalle los activos de un dominio administrativo (como por ejemplo un país, un sector, etc.), tanto por su cantidad como por su diversidad y, en mayor medida, las vulnerabilidades de dichos activos que pueden ser aprovechadas por las amenazas.
Por lo tanto, es necesario emplear otra metodología, siguiendo el esquema de la figura:

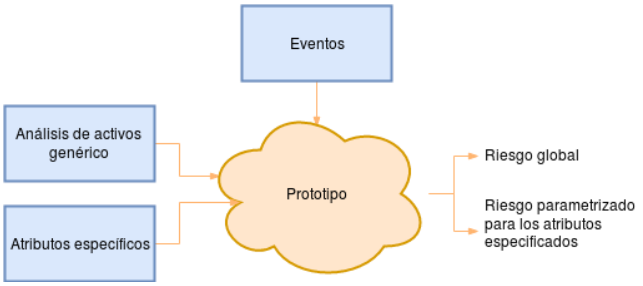


Fig. 2. Metodología seguida en el prototipo.

Como se puede observar, se parte de un análisis genérico de los activos en el dominio administrativo, y se les caracteriza mediante unos atributos específicos. Esta información se modela junto con los distintos eventos que llegan al sistema representando distintas amenazas.
La arquitectura está basada en el uso de un razonador semántico que debe manejar una gran cantidad de información, por lo que es necesario utilizar las técnicas adecuadas para tratarla manteniendo su semántica. Por ello se ha escogido un sistema basado en ontologías, capaces de estructurar dicha información convenientemente y además inferir nuevo conocimiento, en este caso, el nivel de riesgo.
Por lo tanto la primera parte de la arquitectura, consiste en recolectar la información, tanto de activos como de eventos de amenazas y modelarla usando OWL, el lenguaje de ontologías más utilizado actualmente y definido por W3C.
Así pues el primer paso será realizar la identificación y evaluación de activos y su modelado en OWL.
El modelado de activos de un dominio administrativo para la gestión dinámica del riesgo de ciberseguridad se ha estructurado de la siguiente forma. En primer lugar, se han definido los distintos actores intervinientes en el dominio administrativo, con una clasificación de acuerdo a su potencial grado de exposición a riesgos de ciberseguridad.

- Ciudadanos:
  - Básico
  - Medio
  - Avanzado
- Empresa
  - Autónoma:
    - Estándar
    - Tecnológica
  - PYME:
    - Estándar
    - Tecnológica
  - Grande:
    - Estándar
    - Tecnológica
  - Industrial
  - Infraestructura Crítica
- Investigación y Universidad

Una vez identificados los distintos actores de un dominio administrativo, es necesario hacer una evaluación de cada uno. Es decir, dotarlo de atributos específicos. Los atributos escogidos para caracterizarlos han sido los siguientes:

- Conocimiento en ciberseguridad del actor
- Importancia del actor en el marco del dominio administrativo.
- Importancia de los dispositivos para el actor
- Distribución de Sistemas Operativos si procede por cada dispositivo.

Como se puede observar, cada activo posee unos dispositivos, y estos a su vez, si procede, tienen una penetración de Sistemas Operativos por dispositivo para ese activo. Los dispositivos son:

- Móviles
  - iOS
  - Android
- Ordenadores
  - Windows
  - Linux
  - MacOS
- Servidores
  - Windows Server
  - Linux
- Routers
- Switches
- IoT
- Dispositivos de Control Industrial

Por ejemplo un Ciudadano Básico tendrá un bajo conocimiento en ciberseguridad, con una cierta importancia en el marco de un dominio administrativo nacional o regional, y a su vez, con una relevancia al tipo de dispositivo. Por ejemplo, tendrá un 30% de sistemas MacOS, por otro lado tendrá un peso cercano a 0% en servidores. Sin embargo, para una Empresa Grande Técnologíca un servidor tendrá mucha más importancia, y tendrá más porcentaje de ordenadores Linux que un Ciudadano Básico.

Para inferir el cálculo del riesgo considerando los parámetros anteriores, la arquitectura hace uso de SWRL (*Semantic Web Rule Language*), mediante un conjunto de reglas, que consisten en un antecedente y un consecuente, es decir, si se cumple el antecedente, se pasará a cumplir el consecuente. En el caso que nos ocupa, si una amenaza detectada afecta a un tipo de dispositivo (antecedente) se procede a ejecutar el consecuente, es decir, se calculará el impacto sobre los activos que posean dicho dispositivo afectado.

Además el sistema está diseñado para recibir eventos de amenazas de varias fuentes. Así pues, se ha diseñado también un módulo para caracterizar la confianza en las fuentes de información. Dada una fuente de información, sus eventos de amenazas serán considerados con más o menos relevancia en función de la confianza en la misma.

En Fig. 3 se puede observar el sistema descrito anteriormente en su conjunto:

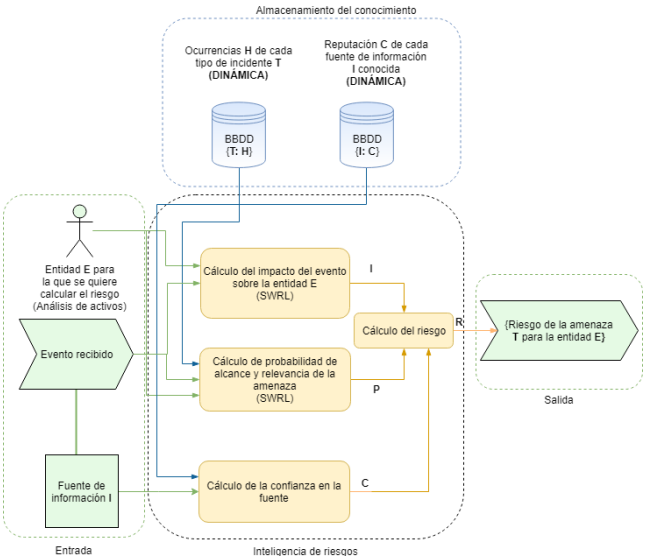


Fig. 3. Arquitectura del prototipo.

Además, si las amenazas están geolocalizadas, el sistema calcula el riesgo en los distintos dominios administrativos geográficos, como Comunidad Autónoma (C. A.), incluyendo a las ciudades autonomas Ceuta y Melilla. Así pues, el sistema es capaz de calcular los siguientes riesgos en distintos dominios adminstrativos, tales como:

- Riesgo global de España.
- Riesgo de un activo específico en España.
- Riesgo global de una Comunidad Autónoma.
- Riesgo de un activo específico en una Comunidad Autónoma

IV. Prototipo

Se ha desarrollado un prototipo basado en la arquitectura propuesta. Este prototipo se ha diseñado para ser capaz de realizar mediciones y cálculos con gran frecuencia, y obtener niveles de riesgo en un escenario cercano a tiempo real.

Ha sido implementado usando el lenguaje de programación Java ya que la principal librería para el manejo de ontologías programáticamente, OWLAPI [14], ha sido desarrollado para este lenguaje.

El primer paso es caracterizar los activos anteriormente definidos. Para ello el prototipo analiza un archivo de configuración que ha de haber sido configurado previamente con los atributos específicos definidos en la sección III del documento. A partir de toda esta información y haciendo uso de OWLAPI se generan ejemplares de activos en la ontología.

Por otra parte, el prototipo desarrollado recibe las amenazas a través de Syslog [15] y éstas a su vez se caracterizan siguiendo el formato CEF. El sistema recolecta la siguiente información de las amenazas: fecha, nombre de la amenaza (ej *Andromeda*), tipo de amenaza, dirección IP de origen, URL de destino, severidad, código de la región y confianza con la que se detecta la amenaza.

El prototipo realiza adicionalmente un enriquecimiento de conocimiento sobre la amenaza utilizando fuentes externas de

inteligencia de amenazas, que nos permita mejorar su correlación con el análisis de activos. Haciendo uso del servicio Antibotnet de INCIBE, ofrecido a través de la web de la Oficina de Seguridad del Internauta (OSI), se recaba información sobre los tipos de dispositivos a los que afecta la amenaza. Esta búsqueda se realiza mediante un *web scraper* buscando palabras claves como "Windows", IoT", "Router", etc. En caso de no encontrarse dicha información se realiza una asignación de dispositivos afectados basada en probabilidades definidas en la configuración del prototipo.

Tanto las amenazas cuya información es encontrada como de las que no, son registradas en bases de datos. Así se evita realizar peticiones web innecesarias. En caso de que no se encuentre información sobre las amenazas, igualmente se registran para recopilar las amenazas sobre las que el servicio de inteligencia de amenazas utilizado no dispone de información. El diagrama de flujo se puede observar en laFig.4

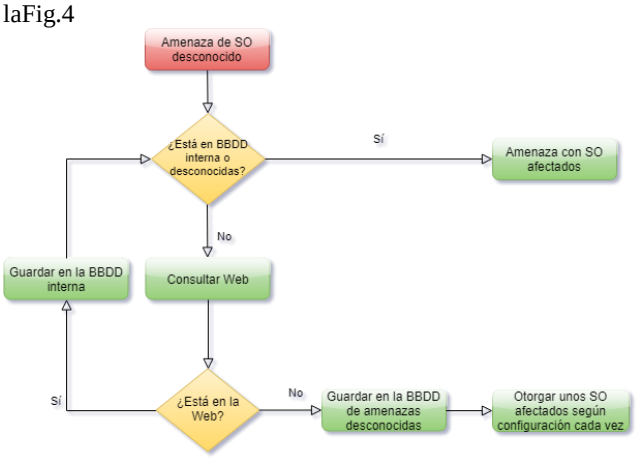


Fig. 4. Diagrama de flujo del cálculo de la confianza.

Con toda esta información se pasa a modelar cada evento de amenaza en formato OWL para que pueda ser utilizada por el razonador. En la Fig. 5 se puede observar mediante la aplicación *Protégé* como quedaría un ejemplar de amenaza:

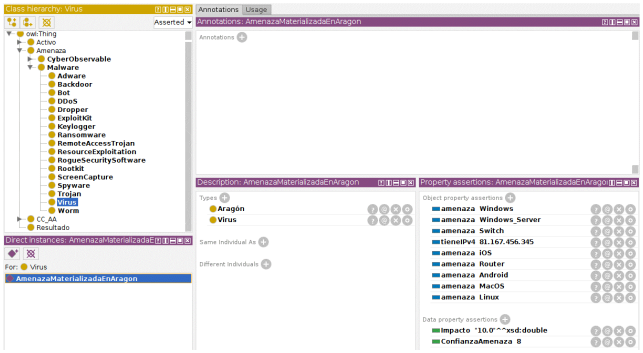


Fig. 5. Información de una amenaza representada en una ontología.

Una vez que tanto los activos como las amenazas están modelados en formato OWL se pasa a ejecutar las reglas SWRL. Estas cubren toda la combinatoria entre los distintos dominios administrativos, tipos de activos, tipos de dispositivos y distintos Sistemas Operativos de los dispositivos, si procede. En total son 2508 reglas. En la Fig. 6 se puede observar un ejemplo de una regla que analiza el riesgo de las amenazas materializadas en el dominio administrativo Comunidad Autónoma de Aragón sobre los servidores de una empresa crítica que usan Windows Server.

```
Modric2:Amenaza(?amenazaAAnalizar) ^ Modric2:Aragón(?amenazaAAnalizar) ^
Modric2:Impacto(?amenazaAAnalizar, ?impact) ^
Modric2:Infraestructura_Crítica(?tipoIndividuo) ^
Modric2:amenaza(?amenazaAAnalizar,Windows_Server) ^
Modric2:ConocimientoCiberseguridad(?tipoIndividuo, ?conocimiento) ^
Modric2:ConfianzaAmenaza(?amenazaAAnalizar,?confianza) ^
Modric2:PesoServidor(?tipoIndividuo, ?pesoDeDisp) ^
Modric2:Servidor_Windows_Server(?tipoIndividuo, ?porcentajeSOdisp) ^
swrlb:multiply(?resultado1, ?impact, ?conocimiento) ^
swrlb:multiply(?resultado2, ?resultado1, ?porcentajeSOdisp) ^
swrlb:multiply(?resultado3, ?resultado2, ?confianza) ^
swrlb:multiply(?riesgofinal, ?resultado3, ?pesoDeDisp) ->
Modric2:Riesgo_Servidor(Resultado_Aragón_Empresa_Infraestructura_Crítica,?
riesgofinal)
```

Fig. 6. Ejemplo de una regla SWRL para el cálculo de riesgo.

Todos los pasos anteriormente descritos quedan reflejados en Fig. 7

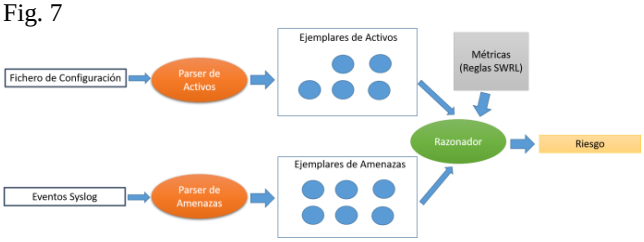


Fig. 7. Esquema a alto nivel de funcionamiento del prototipo.

Sin embargo, cada regla SWRL solo aporta la contribución instantánea al riesgo de un activo, ya que únicamente calcula el riesgo sobre un determinado dispositivo, o sistema operativo de un activo específico en dominio administrativo determinado.

En este prototipo se ha realizado la aplicación de este sistema a los dominios administrativos geográficos de comunidades autónomas (CC. AA.) y país, por lo que es necesario analizar las interrelaciones entre todos estos dominios administrativos.

La suma de los valores arrojados por las reglas que atañen a un activo en una C. A. determinada serán el riesgo del mismo, sin tener en cuenta las amenazas materializadas en el resto de las comunidades.

Por lo tanto, para obtener el riesgo de dicho activo a nivel de país, bastará con aplicar la Ecuación (1), siendo *j* el tipo de activo e *i* la Comunidad Autónoma.

$$Riesgo_j = \frac{\sum_i R_{j,i} * A_i}{\sum_i A_i} \quad (1)$$

En una Comunidad Autónoma, la suma ponderada de los activos por su peso, según se especifique en la configuración inicial, tendrá como resultado el riesgo de la C. A. Así pues, el riesgo en todo el territorio nacional se obtendrá según la Ec. (2).

$$Riesgo\ España = \frac{\sum_i R_i * A_i}{\sum_i A_i} \quad (2)$$

Si bien estos cálculos resultan adecuados para la obtención del riesgo de España, no lo son tanto para el riesgo de una C. A. o de un activo dentro de la misma. Esto es debido a que no se está modelando que las amenazas materializadas en una C. A. determinada pueden con gran facilidad materializarse en otras. Para caracterizar esta transferencia de riesgo se usa el

parámetro de correlación (ρ) entre comunidades, que indica cuánto porcentaje de riesgo del resto de comunidades afecta a la comunidad siendo analizada. Con esta nueva apreciación el riesgo de un activo *j* en una Comunidad Autónoma *k* quedaría como Ec. (3).

$$Riesgo_{j,k} = \frac{R_{j,k} * A_k + \rho * \sum_{i \neq k} R_{j,i} * A_i}{\sum_i A_i} \quad (3)$$

Y a su vez, el riesgo de una C. A. *k* pasaría a considerar el riesgo del resto de Comunidades Autónomas según la Ec. (4).

$$Riesgo_k = \frac{R_k * A_k + \rho * \sum_{i \neq k} R_i * A_i}{\sum_i A_i} \quad (4)$$

Por lo tanto, el prototipo ya es capaz de ofrecer los riesgos con un alto nivel de detalle. Sin embargo, hasta ahora solo se realizaban cálculos de riesgo instantáneos, solamente usando una medición. Para que tenga en cuenta los resultados de mediciones anteriores se ha dotado al sistema de memoria.

Cuando se realiza una medición se calculan los niveles de riesgo siguiendo el proceso descrito anteriormente, pero además se analizan las mediciones realizadas en el pasado, y mediante el uso de una función de tiempo se computa un nuevo nivel de riesgo, que ya no es instantáneo porque también considera los eventos pasados.

Cuanto mayor es el tiempo transcurrido entre la medición actual y una del pasado, menor será la importancia de le última, es decir, la relevancia de las mediciones decrece con el tiempo. Si se tuviese que analizar todas las mediciones del pasado, a medida que creciesen en número ralentizaría el sistema hasta que lo hiciese impracticable. Por ello en la configuración hay que definir el parámetro tiempo de olvido, que representa la cantidad de tiempo a la que hay que remontarse para tener en cuenta las mediciones pasadas, es decir, el tiempo de memoria del sistema.

Para este modelado, se ha escogido una función hiperbólica cuya ecuación es Ec. (5).

$$f(t) = e^{-\frac{4t^3}{(Tiempo\ de\ Olvido)^3}} \quad (5)$$

Con un valor de tiempo de olvido en 60 minutos la gráfica sigue la siguiente forma de la Fig. 8.

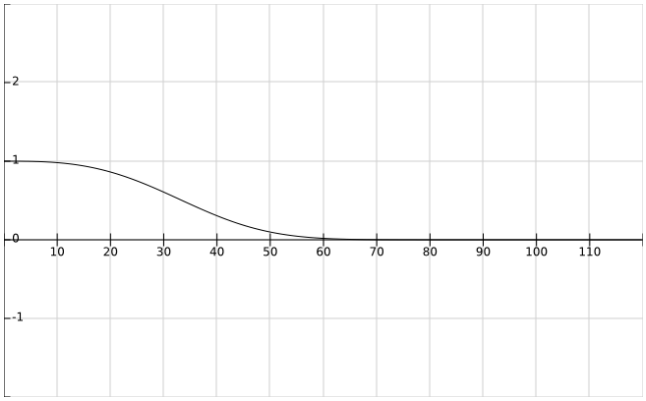


Fig. 8. Representación de la ecuación del tiempo. Ejemplo con un tiempo de olvido seleccionado en 60 minutos.

Como se puede observar, a medida que las medidas analizadas son más antiguas, son penalizadas por dicha función.

También es necesario ponderar por el número de amenazas analizadas en cada medición, ya que, a mayor número de amenazas analizadas, mayor habrá de ser su representatividad en el histograma. El uso de la función del tiempo *f(t)* y del número de amenazas se puede apreciar en Ec. (6).

$$Riesgo\ actual = \frac{\sum_i R(t_i) * N(t_i) * f(t_i)}{\sum_i N(t_i) * f(t_i)} \quad (6)$$

Donde t_0 es el instante actual y por lo tanto $f(t_0) = 1$.

Por último, una vez realizada una medición y todos los cálculos de riesgo realizados es necesario almacenar toda la información recabada. El prototipo almacena la información siguiendo el estándar JSON.

También se ha desarrollado una interfaz gráfica para una mejor visualización de la información que se verá más en detalle en la siguiente sección.

V. Pruebas

Para comprobar las capacidades del sistema se le ha sometido a multitud de pruebas. El sistema se ha probado con una gran cantidad de amenazas entrantes, llegando a ser capaz de analizar cada una en un tiempo medio de 0,01 segundos. También se ha probado con la entrada de amenazas desde más de una fuente de información. En Fig. 9 se puede observar el JSON producido al realizar una medición:

fechaCalculo:2019/03/25 17:02:39
riesgoEspana:3.990046153846283
riesgoEspanaConMemoria:3.296827160476196
numeroAmenazasTotales:126
▾ numeroAmenzasFuentesInformacion{2}
CONFIADA:126
INTERNET:0
▸ individuos{12}
▾ individuosConMemoria{12}
Ciudadano_Medio:4.0020403207625215
Ciudadano_Avanzado:3.66715404929009
Ciudadano_Básico:4.557418226082124
⋮
▾ comunidades [19]
nombreComunidad:Andalucía
riesgoComunidad:2.1675076923077388
riesgoComunidadConMemoria:2.129086239298577
numeroAmenazasComunidad:48
▸ individuosComunidad{12}
▾ individuosComunidadConMemoria{12}
Ciudadano_Medio:2.2994025102746045
Ciudadano_Avanzado:2.094598559546031
Ciudadano_Básico:2.593361070135563
⋮

Fig. 9. Ejemplo del JSON generado en una medición.

Se han analizado un número significativo de amenazas, todas provenientes de una única fuente de información con un nivel de confianza elevado, y, por lo tanto, todas ellas afectadas por la confianza asignada a dicha fuente. Como se puede observar, el campo del riesgo instantáneo de España es considerablemente superior al que tiene con memoria. Por lo tanto, significa que se ha producido un gran incremento instantáneo en el riesgo en esta última medición. Si esta situación se mantuviese en el tiempo, el riesgo con memoria acabaría convergiendo con el instantáneo.

Para facilitar la visualización de los datos arrojados por el sistema, se ha realizado una interfaz gráfica usando la librería *JFreeChart*. En la Fig. 10 se puede observar un histórico del riesgo de España con memoria y el instantáneo.

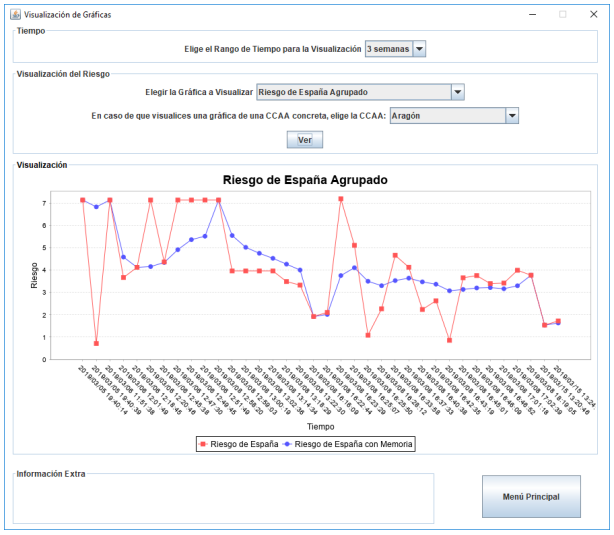


Fig. 10. Visualización del histórico del riesgo de España.

Como se puede observar, el riesgo con memoria tiende a seguir la tendencia del instantáneo, pero si las variaciones son muy rápidas, tiende a representar la media de estas. Los picos más fuertes del riesgo con memoria se deben a que ciertas mediciones se componen de una gran cantidad de amenazas y por lo tanto gran representatividad.

El histórico de la Fig. 10 se compone de 36 mediciones de riesgo, cada una generada con un rango de eventos de amenazas de entre 100 y 500. Cada evento de amenaza se ha compuesto de su localización geográfica, IP, instante temporal, severidad, confianza en el propio evento, nombre de la amenaza detectada, etc. Se han introducido amenazas con posible gran impacto, y gran confianza en las fuentes de información, y también el caso contrario, con bajo impacto, y escasa confianza en las fuentes de información. Es por ello por lo que la tendencia instantánea, representada con la línea roja presenta picos tan pronunciados.

Además, el sistema también nos permite analizar un histograma pormenorizado por activos, e incluso por activos de una C. A. También permite observar el riesgo de las Comunidades Autónomas, como es el caso de la Fig. 11.

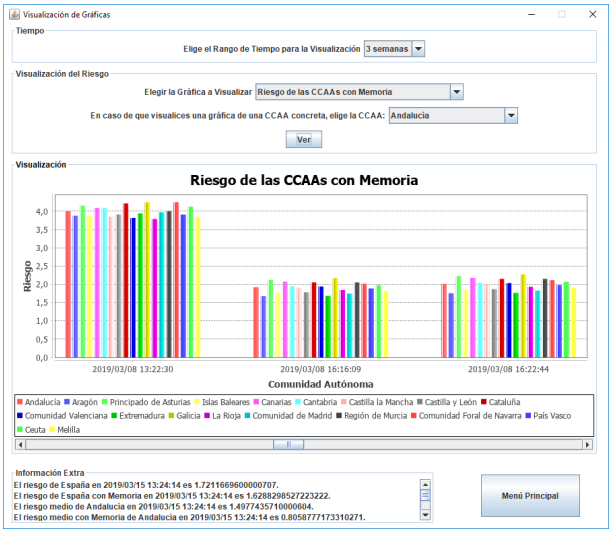


Fig. 11. Visualización del histograma de los riesgos de las distintas Comunidades Autónomas.

Por último, también se puede observar (Fig. 12) el riesgo actualizándose cada 10 segundos, a medida que se producen las mediciones:

Fig. 12. Visualización del riesgo de España en tiempo real.

VI. Conclusiones

En este documento se ha presentado una posible solución a la problemática que presenta obtener un nivel de riesgo en un dominio administrativo en tiempo real. Se ha diseñado una arquitectura capaz de procesar un gran flujo de información y de modelar el riesgo con unos activos de los que es imposible conocer características muy específicas, así como vulnerabilidades concretas de los mismos.

Debido a estas limitaciones, el modelado se ha realizado usando una definición de activos orgánica; dividiendo entre, ciudadanos, empresas y entornos universitarios e investigación. Se han diseñado subgrupos de activos para lograr una caracterización adecuada y se les han dotado de atributos, como: conocimiento de ciberseguridad, importancia dentro del marco del dominio administrativo, importancia de cada tipo de dispositivo, etc. Además, para poder correlar las amenazas detectadas con los activos se han creado grupos de

dispositivos, como móviles, ordenadores personales, dispositivos IoT, etc. Además, se han representado los principales sistemas operativos por tipo de dispositivo en función de la penetración de mercado; por ejemplo, los dos sistemas operativos dominantes para los móviles son Android e iOS. De esta manera, al recibir un evento de amenaza, se recopila inteligencia adicional, y así, por ejemplo, en el caso que una amenaza afecte a Android sabremos que afectará a los activos que posean un móvil con dicho SO.

La arquitectura se ha diseñado para ser capaz de tratar adicionalmente varias fuentes de información de eventos de amenazas, modelándolas mediante un sistema de confianza dinámico. Por último, el sistema se ha ejemplarizado para el caso del dominio administrativo nacional, por lo que la caracterización de los activos y la información proveniente de las amenazas permite no solo calcular el riesgo de España, sino también de cada activo, de cada Comunidad Autónoma y de cada activo por Comunidad Autónoma. Para procesar la gran cantidad de información en tiempo real se ha usado un modelo basado en ontologías, mediante el uso de reglas *SWRL* y del razonador semántico *Pellet*.

Se ha desarrollado un prototipo basado en la arquitectura anteriormente descrita en un entorno real, y así se ha podido probar su validez con pruebas de campo reales. Se ha sometido a gran cantidad de pruebas satisfactoriamente, después de refinar los motores de razonamiento para obtener la velocidad de inferencia necesaria. Ha demostrado ser un acercamiento válido al problema, donde es necesario razonar qué activos son afectados por las distintas amenazas y en qué grado, todo ello con un escaso conocimiento de los activos, y un alto flujo de eventos de amenazas, funcionando en un sistema en tiempo real.

La principal contribución de la propuesta presentada se basa en la propuesta de métrica de inferencia del nivel de riesgo en tiempo real en un entorno de definición difusa de los activos. Estas métricas permiten modelar genéricamente las características del dominio administrativo y se trata de un modelo extrapolable a otros dominios administrativos, con la adecuada adaptación de las características del dominio, y los modelos de activos. Por otra parte, este hecho también puede constituir una limitación en el momento en el que las fuentes utilizadas para el cálculo de las métricas sean diferentes, con otras semánticas, en el que la extrapolación del modelo puede tener una mayor complejidad.

AGRADECIMIENTOS

Los autores agradecen el apoyo prestado por el Instituto Nacional de Ciberseguridad (INCIBE) para la realización de este trabajo.

REFERENCIAS

[1] Ministerio de Administraciones Públicas. "*MAGERIT v.3: Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información*". NIPO: 630-12-171-8. Octubre 2012.

[2] A. Herzog, N. Shahmehri, C. Duma. "*An ontology for information security*", Int. Journal of Information Security and Privacy, 2007.

[3] S. Fenz, "*Ontology-based Generation of IT-Security Metrics*", Proceeding of the 41 Hawaii Int. Conference on Systems Sciences, 2008.

[4] L. Obrst, et al., "*MITRE-Developing an Ontology of the Cyber Security Domain*", MITRE, 2012.

[5] S. Singapogu, et al., "*Security Ontologies for Modeling Enterprise Level Risk Assessment*", 2012 Annual Computer Security Applications Conference, Orlando, 2012.

[6] R.F. Erbacher, "*Ontology-based Adaptative Systems of Cyber Defense*", Semantic Technology for Intelligence, Defense and Security Conference, Farifax, VA, 11/2015.

[7] A. Ekelhart, S. Fenz, M. Klemen, E. Weippl, "*Security Ontologies: Improving Quantitative Risk Analysis*", Proceedings of the 40th Hawaii International Conference on System Sciences, 2007.

[8] JEL de Vergara et al., "*A Semantic Web Approach to Share Alerts among Security Information Management Systems*", Communications in Computer and Information Science 72:14-25., 2010

[9] Z. Syed, et al., "*UCO-Unified Cybersecurity Ontology*", The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence. Artificial Intelligence for Cyber Security: Technical Report WS-16-03.

[10] R. Riesco, V. A. Villagrá, "*Leveraging Cyber Threat Intelligence for a Dynamic Risk Framework*", Aceptado para su publicación en International Journal of Information Security, 2019.

[11] G. Goluch, A. Ekelhart, S.Fenz, S. Jakoubi, S. Tjoa y T. Mck, "*Integration of an Ontological Information Security Concepto in Risk-Aware Business Process Management*", Proceedings of the 41st Hawaii International Conference on System Sciences, 2008.

[12] V. Mateos, V. A. Villagrá, F. Romero, "*Ontologies-Based Automated Intrusion Response System*", Computational Intelligence in Security for Information Systems 2010, pp. 99-106, 2010.

[13] I. Romero, R. Riesco, F. Barea, V. A. Villagrá, "*Arquitectura de Gestión Dinámica de Riesgos basada en Ontologías y Reglas de Comportamiento*", XV Reunión Española sobre Criptología y Seguridad de la información (RECSI). Granada (España). Octubre 2018.

[14] Horridge, Matthew, Bechhofer, Sean, *"The OWL API: A Java API for OWL ontologies*", Semantic Web, vol. 2, no. 1, pp. 11-21, 2011.

[15] H. Tsunoda and G. M. Keeni, "Managing syslog," *The 16th Asia-Pacific Network Operations and Management Symposium*, Hsinchu, 2014, pp. 1-4.

Mirror Saturation in Amplified Reflection DDoS

João J. C. Gondim
Depto. de Ciência da Computação
Programa de Pós-graduação em Engenharia Elétrica
Universidade de Brasília
Brasília, Brasil
gondim@unb.br

Robson de Oliveira Albuquerque
Depto. de Engenharia Elétrica
Programa de Pós-graduação em Engenharia Elétrica
Universidade de Brasília
Brasília, Brasil
robson@redes.unb.br

***Abstract*—Over the last six years, there has been two major game changers in DDoS attacks: amplified reflection and IoT. Together, they motivated well-founded security concerns relating to IoT's offered attack surface, and how it could potentialize DDoS. In order to assess those concerns, the feasibility of IoT device abuse as reflectors was evaluated. Attacks abusing two protocols were tested showing a pattern: reflector saturates without sustaining maximum amplification rates, for very low injection rates (between 10 and 100 probe/sec). Hence, if on the one hand IoT devices, in general, would not be good reflectors, they would be good injectors. An attacker could thus use more injectors while maintaining low injection rates. This would certainly require greater coordination from the attacker but tends to hamper detection. It is expected higher sophistication in DDoS attack execution, as in carpet bombing and pulse attacks, with evolution of C2 incorporating orchestration and attack coordination.**

***Index Terms*—DDoS, amplified reflection**

Tipo de contribución: *Investigación original (límite 8 páginas)*

I. Introduction

DoS (denial of service) attacks have been around in the Internet since the mid 70's [4]. Architecturally, the first breakthrough in volumetric DoS was when the attack was distributed and included several sources of traffic [5]. Initially, groups of manually compromised hosts (slaves) under central coordination (of a master) executed attacks of much high volume against targets. However, attack preparation demanded considerable effort. DDoS takes a significant leap in traffic volumes when self propagating malware started being used for building large networks (botnets) of slaves (bots). Then, preparation was much simpler but still involved developing some appropriate malware to infect, control, propagate and execute attacks. The architecture changed and masters become command and control centers with bots reporting to them after infection. Botnets were formed including thousands of bots, in contrast with zombie nets which were much smaller. As a direct consequence, attack volumes lept from $Mbps$ to $Gbps$. It is the motivation for [3].

The next further improvement and first of the game changers in recent years was the introduction of amplified reflection. Paxson described them initially [6] and Rossow revisited them in [7] after their use in the mid 2010's. In an amplified reflection DDoS attack (AR-DDoS) preparation takes virtually minimum effort. There is no need to infect host and install malware on them. It only requires certain particular configurations in hosts to abuse them as reflectors. The effort for probing for a potential reflector is the same as that of triggering an attack.

IoT has already demonstrated its potential for DDoS, with several incidents (e.g. [11], [12], [13]), being Mirai [14] the most remarkable so far. However, Mirai didn't use AR-DDoS [15] generating speculation about possible AR-DDoS attacks abusing IoT devices. [1] and [2] address this question indicating mirror saturation, but the first was a simulation and later tested only one protocol.

Here, the objective is to extend the study of mirror saturation dynamics assessing reflector behavior for other protocol to further evaluate if AR-DDoS can further potencialize the threat posed by IoT. The methodology used is consistent with that applied in [1].

A. Statement of contribution

The main contributions are:

- study mirror saturation dynamics assessing reflector behavior for protocols SNMP and SSDP;
- comparison and characterization of saturation for those; and
- provide further indication that for using IoT devices effectively increases attack complexity.

B. Outline of paper

Section II describes the attacks examined. Section III describes tests executed and the methodology used. Results (Section IV), along with their discussion, are presented in the sequel. Section V present conclusions and further directions.

II. Attack preliminaries

DoS attacks aim at exhausting device or infrastructure resources in order to cause service unavailability. In its volumetric form, attacks basically consist of sending a large number of requests that either overwhelm services when attempting to process them, or exceed traffic limits [19]. One common form of implementing and potentializing DoS attacks is using distribution (distributed DoS, DDoS), in which several nodes send traffic in a coordinated way, resulting in higher attack efficiency and more complex mitigation given source obfuscation. There are several forms of DoS attacks, which can be organized in two major classes: volumetric attacks, and protocol abuse. The latter covers low volume, slow rate attacks where legitimate traffic probes exploiting specific protocol features, characteristics, or implementation details lead to an exhaustion of some of the victim's resources, and consequently, legitimate requests are not properly responded to. The first class, on the other hand, includes attacks where large traffic volumes flood the victim, exceeding its processing

capacity or link bandwidth, so that legitimate requests are not treated. As for volumetric attacks, they can also be divided into direct attacks and reflection attacks. Both forms try to overload some victim resource (usually bandwidth) by sending large traffic volumes to the victim. In direct flooding attacks, compromised nodes send traffic straight to the victim; while in reflection attacks, intermediate nodes (reflectors) are used to flood the victim. For the purposes of the attacker, a reflector is any node that sends an IP datagram in response to one previously received. For AR-DDoS, reflectors of interest are those that amplify; i.e., their response produces more bytes or packets, or both, than the original input request datagram. This behavior is characterized by a reflection factor, indicating how much traffic is generated by the reflector. So, amplifying reflectors potentialize traffic generated by an attacker [6]. A final ingredient common to most DoS attacks, including AR-DDoS, is IP spoofing, which consists of crafting arbitrary source addresses in IP datagrams. In DoS attacks, attackers use this technique to obfuscate the real attack source. Characteristically, in AR-DDoS, attackers send traffic to reflectors using the victim's address as source address [20].

A. Attack with SNMP Protocol

SNMP is the Simple Network Management Protocol [8]. It defines how managed devices and management stations communicate and exchange information. Managed devices run agents and maintain a database, the Management Information Base (MIB), that contains variables representing management data. These variables can be used both for monitoring and for invoking actions on managed nodes. The requests for query or change of variables are made through messages sent via UDP port 161.

In versions 1 and 2c, message authentication is based on the value of the "community" field, which is encoded as full text and transmitted in the clear. There is a "public" community that comes as default [9].

Because SNMP supports messages sent via UDP, IP packet spoofing is trivial. In this way, an SNMP message with IP spoofing and "community" field equal to "public" allows to direct MIB information to a third party.

In version 2 and derivatives a new GetBulkRequest request message was implemented, which serves to retrieve the value of several variables, including lines of tabular variables, in only one message. A tabular variable is one that can have multiple iterations, which represent a row in a table. If the value of the "maxrepetitions" argument of the GetBulkRequest message exceeds the number of rows in the table, the next MIB variables are returned. In this way, an attacker can send a GetBulkRequest with a high number of "max-repetitions", generating a response message with significant size.

Unlike more recent protocols, SNMP has no recommendation for limiting the response interval. However, because of the large processing cost of GetBulkRequest, specially when accessing variable length tables, the responder may not be able to send the responses immediately.

Modus Operandi

Attack takes place according to the following steps:

a) A GetBulkRequest request is generated by the attacker with the identifier of one or more tabular variables. The "non-repeaters" field should be 0, which indicates that there are no scalar variable identifiers in the message. The value of the "max-repetitions" field must be high but chosen such that the answer does not exceed the limit size defined by Internet Protocol version 4 (IPv4), that is, 65,535 bytes;
b) A UDP datagram for the message is generated with port 161, which is reserved for SNMP;
c) An IP packet with spoofed source address is created to match the target's address and destination address equal to that of the reflector's. The UDP datagram is then placed in an IP packet;
d) The UDP datagram is placed in an IP packet and sent normally to the reflector;
e) The reflector receives the packet, processes the SNMP GetBulkRequest, generating a long response with the values for the required variables;
f) Finally, the reflector sends the message to the target.

B. Attack with SSDP Protocol

SSDP is the Simple Service Discovery Protocol and is part of UPnP (Universal Plug ang Play), which was created to extend the concept of "plug-and-play" to devices over a network. The version used in tests was 1.0 (the most used [16]) and is defined in UDA 1.0 (UPnP Device Architeture) specification [17]. SSDP uses a multicast HTTP variant over UDP port 1900.

SSDP has control points which can at any time send an M-SEARCH request to search for devices in the network. Each device then sends messages corresponding to the root device, its embedded devices, and its services. The response, in addition to generating messages larger than an M-SEARCH, generates more than one message, since a device is expected to have minimally one service, which generates a message for the root device and another for the service, evidencing the potential for amplification.

Each root device with d embedded devices and k different kinds of service generates $(3 + 2d + k)$ messages

In order to avoid network congestion. SSDP uses the MX field of the M-SEARCH message to indicate the maximum time for sending the response, as explained above in the discovery step. Although there is a recommendation for MX to be at least 1, the specification itself allows the devices to assume values lower than those specified by MX. This means that it is plausible to generate significant traffic. Another point to note is that although the MX field is mandatory, the responding device behavior in relation to the value of this field is not mandatory. Thus, the ability to generate sustainable amplification depends on implementation. As reported in [18], most reflectors use the same implementations as in [16].

Modus Operandi

The attack using message M-SEARCH takes place according to the following steps:

a) An M-SEARCH message is created with MACM value equal to 1, the lowest recommended, and ST with a value equal to "ssdp:all", so that the message applies to all UPnP devices on the network;

b) A UDP datagram is created with the message and destination port 1900, which is reserved for SSDP;
c) An IP packet with spoofed source address is created to match the target's address and destination address equal to that of the reflector's. The UDP datagram is then placed in a IP packet;
d) The IP packet is sent to the reflector, which can be any UPnP device that responds to discovery messages from outside the local network;
e) The reflector receives the message and inadvertently sends the response to the target.

III. Tests and methodology

The environment used consisted of 4 devices:

- Attacker: the attacker performed the attack sending probes to the reflector. The attacker configuration was a Dell Inspiron 14R 5437 notebook running Windows 10 operating system and hosting an Ubuntu 14.04 LTS virtual machine which effectivelly launched the attack. Configuration used was:
  - i7-4500U @ 1.80GHz processor
  - 8GB DDR3L @ 1600MHz RAM
  - RTL8136 network adapter
  - Windows 10 Home x64 host system Version 1803
  - VirtualBox 5.2.18 virtualization program (virtual network in bridge mode 2)
  - Ubuntu 14.04 guest system (64-bit)
- Reflector: the reflector runs services which on the device that performs programs that use SNMP and SSDP protocols. In tests carried out, a computer, with Windows Operationg System, running Windows Media Player, which uses UPnP for media sharing; and the SNMP service [28] which is Microsoft's implementation of an SNMP agent, an optional feature of Windows OS. The configuration is as follows:
  - i5-660 @ 3.33GHz processor
  - 8GB DDR3 @ 1333MHz RAM
  - RTL8111D Gigabit Ethernet network adapter
  - Windows 10 Pro x64
- Target: the target is the final attack flow destination. It only executes Wireshark to colect attack data. It's configuration was:
  - i3-380M @ 2.53GHz processor
  - 4GB DDR3 @ 1333MHz RAM
  - AR8152 network adapter
  - Windows 10 Operating System Home x64 Version 1803
- Switch: the active network device used was a TP-Link TL-WDR4300 wireless router.

The tool used to perform the attack was developed for Linux. Virtualization was used to make the attacker less powerful than the reflector due to VM overhead. This scenario was not tested in previous work. Since a virtual network card was used in bridge mode 2, the attacker's machine acted exactly as if it were connected to the same network of the reflector and target.

This does not interfere with the results since reflector saturation was achieved before the attacker saturated, even with the use of a reflector having superior hardware compared to a typical SSDP or SNMP device.

For both protocols, test routines consisted of 30 second rounds for each attack level. For each attack level L, the number of probes generated by the attacker tool was given by:

$$Probes = 10^{(L-1)} \tag{1}$$

Wireshark [24] ran on the attacker, reflector and target during attack rounds. From traffic captures collected, flows in bits per second (bps) and packets for second (pkt/s) for reflector inbound and outbound flows were obtained and amplification was calculated as:

$$Amp_{bit} = \frac{bit_{out}}{bit_{in}} \tag{2}$$

for bit amplification, and also in packets:

$$Amp_{pkt} = \frac{pkt_{out}}{pkt_{in}} \tag{3}$$

IV. Results

A. Attack with SNMP

Tables I and II show results for reflector behavior during the attack using SNMP in terms of bit and packet flows respectively.

TABLE I
SNMP Attack (bps)

Level	Reflector inbound	Reflector outbound	Amplification
1	647	394468	609, 1
2	6408	3894071	607, 7
3	63919	38614906	604, 1
4	630051	57614231	91, 4
5	702535	57157632	81, 4
6	719934	59061424	82, 0

TABLE II
SNMP Attack (pkt/s)

Level	Reflector inbound	Reflector outbound	Amplification
1	1	33	33,0
2	10	330	33,0
3	100	3.195	32,0
4	985	4.825	4,9
5	1098	4.815	4,4
6	1125	4.924	4,4

As can be observed, maximum amplification rates, either in bits or in packets, are sustained only until attack level 3. From that point on, it decays but maintains some residual gain. This can be easily verified in Figures 1 and 2.

It can also be observed that inbound traffic also saturates but from level 3 on, instead of level 2, as for the amplification.

B. Attack with SSDP

Results for reflector behavior during the attack using SSDP in terms of bit and packet flows are in Tables III and IV respectively.

Similarly to SNMP, the maximum amplification rates, either in bits or in packets, are sustained only up to attack level 2, in contrast with SNMP which went up to level 3. From that point on, it decays and maintains some residual gain in bits but atenuates in packets (Figures 3 and 4).

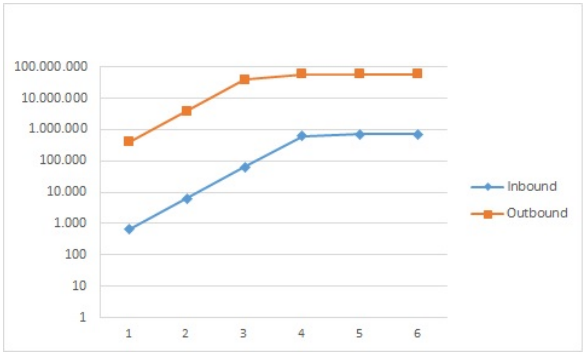


Fig. 1. SNMP Attack (bps)

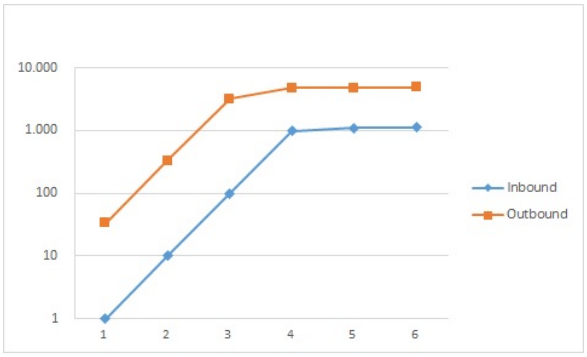


Fig. 2. SNMP Attack (pkt/s)

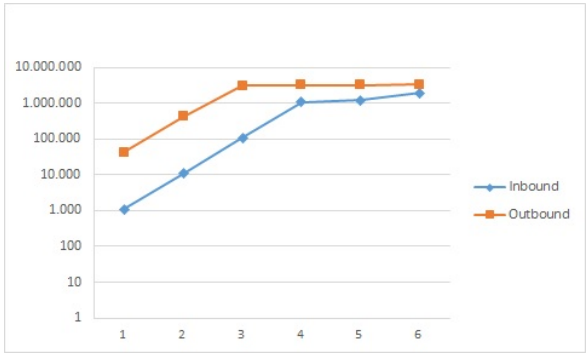


Fig. 3. SSDP Attack (bps)

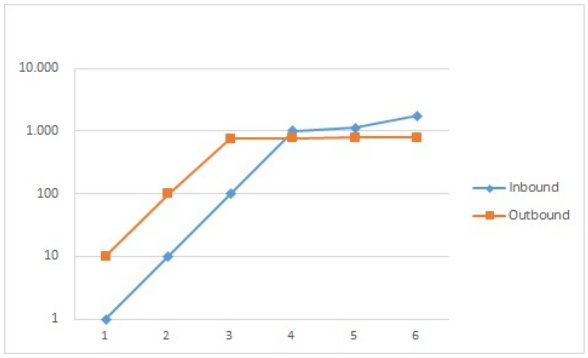


Fig. 4. SSDP Attack (pkt/s)

Also, It can be observed, as for SNMP, that inbound traffic saturates but from level 3 on, instead of level 2, as for the amplification.

C. Amplification

Amplification factor behavior is presented in Figures 5 and 6. As they show, for both attacks saturation occurs. For SNMP, amplification abruptly falls from level 3 on. For SSDP, amplification falls slightly from level 2 to 3 and then falls abruptly.

TABLE III
SSDP ATTACK (bps)

Level	Reflector inbound	Reflector outbound	Amplification
1	1099	41798	38,0
2	10893	416487	38,2
3	108926	3116813	28,6
4	1088113	3253133	3,0
5	1227497	3279005	2,7
6	1914556	3311109	1,7

TABLE IV
SSDP ATTACK (pkt/s)

Level	Reflector inbound	Reflector outbound	Amplification
1	1	10	10,0
2	10	100	10,0
3	100	748	7,5
4	1000	777	0,8
5	1128	784	0,7
6	1758	793	0,5

D. Discussion

SNMP test results are consistent with previous results [1], either in general reflector behavior as in numerical terms. Results diverge slightly for higher attack levels but agree on saturation behavior. In previous tests there was attenuation from level 6 on, in bits and in packets. Here, there was some residual amplification. One possible explanation for this difference is that now the attacker and reflector are more even in terms of computational power than in tests mentioned. So, the attacker does not overwhelm the reflector with traffic.

SSDP, as expected, showed much lower amplification rates than SNMP but produced the same shape in overall behavior. The only difference is some attenuation for higher attack levels in terms of packet amplification.

There two intriguing and suggestive findings when comparing reflector behavior. The first is that for both protocols, reflector behavior is basically the same.

The other finding is that not only did the reflector saturate, in terms of unsustained maximum amplification, but saturation occurred for the same attack levels.

It was expected that for different protocols, providing amplification in very diverse forms, completely dissimilar implementations with contrasting computational demands, saturation would occur but not necessarily at the same attack levels. So this convergence is striking. It is speculated which architectural characteristic common to test setups is determinant to this convergence. The most likely candidate is related to I/O bus standards, but that requires further investigation.

There is yet an even more surprising aspect to point out which may also relate to some architectural feature. Saturation

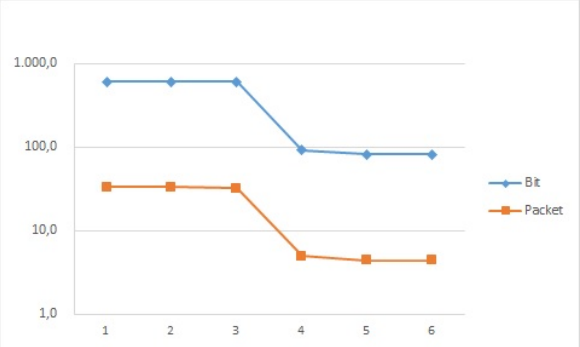


Fig. 5. SNMP Amplification

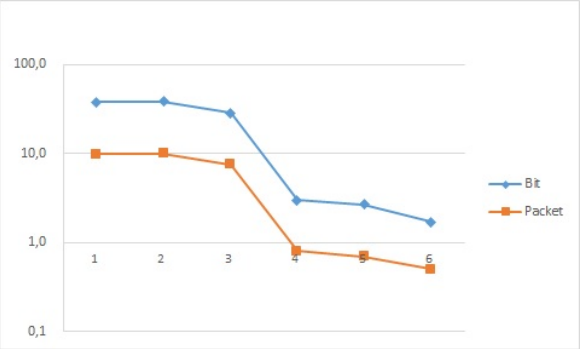


Fig. 6. SSDP Amplification

took place for both attacks, starting at level 2 and confirming on level 3. That represents optimum input probe rates between 10 probe/s and 100 probe/s which are extremely low attack input rates when compared the those commonly used, typically exceeding tens of thousands of probes per second [21]. Consequently, maximum attack efficiency is achieved for low probe input rates, minimizing attack execution effort. An attacker might then involve more reflectors and generate more attack traffic without having to increase his capacity, despite some additional management complexity.

At this point, it is in order to confront test results with data from real events, if available. [13] provides some real attack data and makes some calculations. According to their report, they initially suffered $75Gbps$ a attack which recorded over 30,000 unique reflectors. The reported AR-DDoS attack abused DNS, so that 36 byte long queries amounted to 3000 byte long replies corresponding to an amplification factor of almost 100. Thus, to generate $75Gbps$ of attack traffic, the attacker would have to generate $750Mbps$ of probe traffic (total attack traffic divided by amplification in bits, equation 4):

$$attacker_{vol_{total}} = \frac{vol_{total}}{Amp_{bit}} \quad (4)$$

$$= \frac{75Gbps}{100} \quad (5)$$

$$= 750Mbps \quad (6)$$

That is the attacker effort in terms of traffic volume. The effective injection probe rate per reflector can be then calculated as 87 $probe/s$ dividing the minimum required rate by the number of mirrors (reflectors) used, probe size in bytes and 8 (1 byte = 8 bits) as in equation 7:

$$probe_{rate_{min}} = \frac{attacker_{gen_{traff}}}{(mirrors \times probe_{size} \times 8\ bit)} \quad (7)$$

$$= \frac{750Mbps}{(30000 \times 36 \times 8\ bit)} \quad (8)$$

$$= 87\ probe/s \quad (9)$$

Thus, regardless of the actual injection probe rate used by the real attacker, which is not available, the efective, and minimum, attack effort required is between levels 2 and 3 in tests executed, which is consistent with findings.

However, there are some assumptions which must be clarified. Real attack data obtained refers to an AR-DDoS attack abusing DNS, which was not tested. It is assumed that DNS abuse has similar saturation behavior. Morever, it is assumed that DNS abuse has similar saturation point as that found for tested protocols, *i.e.* between attack levels 2 and 3. So, the figure obtained for saturation point (87 $probe/s$) should be expected when DNS is subject to test. It should be noted, that this lack of attacker data motivated this investigation and its methodology.

V. Conclusion and further directions

Results corroborate previous claims that amplified reflection attacks if directed towards the IoT infrastructure would impact it severely possibly rendering the attack ineffective in hitting the final target. Note that reflector configurations used were much more powerful than that of a typical IoT device, even if only so called smart devices are considered.

This does not totally exonerates the potential threat from IoT infrastructure abuse by AR-DDoS attacks. First, IoT has a huge number of potentially abusable devices. Second, if the attacker modulates probe rates to avoid reflector saturation and increase the number of reflectors involved, high volumes of traffic can be generated. However, more reflectors imply more complex attack management.

In any case, low probe injection rates match IoT device limited resources so that it is feasible to use them as probe injectors for reflection attacks. Again, this would add attack management complexity. But there is already evidence that DDoS is evolving in terms of more refined forms of execution.

Usually, DDoS attacks are execute in a ”brute force” fashion. An attacker uses all capacity needed to hit the target, given prior knowledge of its capacity. However, there is already indication that DDoS execution is changing. Techniques like ”carpet bombing” [22] and pulse wave attacks [23] demonstrate this tendency.

Carpet bombing DDoS attacks scatter traffic within a specific subnet or CIDR block (for example, a /20 block), making it more difficult to detect the attack and also to mitigate it, potentially resulting in outages due to the flood of attack traffic across network devices and internal links. Pulse wave attacks, on its turn, consist of a series of short-lived high intensity traffic pulses occurring in synchronized succession to circumvent hybrid mitigation (in premise jointly with cloud). These techniques target not only the final victim but also mitigating mechanisms used in defense. Consequently, attacks are already security aware.

As already discussed, maximum attack efficiency comes with lower attack effort, *i.e* lower input probe rates. The first

makes live easier for attackers while the latter makes detection and mitigation even more complex.

The additional management burden due to possible inclusion of more reflectors or even injectors might demand better orchestration and coordination from attackers, particularly if using botnets.

Further investigation directions include testing the attack for other protocols (like DNS, NTP, Memcached and CoAP) to verify and possibly corroborate findings related to saturation behavior and saturation point.

VI. Acknowledgement

The authors gratefully acknowledge the support from the Brazilian Research Councils CNPq (Grant 465741/2014-2 INCT on Cybersecurity), CAPES (Grant 23038.007604/2014-69 FORTE), FAP-DF (Grants 0193.001366/2016 UIoT, 0193.001365/2016 SSDDC, and Call 01/2019), as well as the LATITUDE/UnB Laboratory (Grant 23106.099441/2016-43 SDN), the Ministry of the Economy (Grants 005/2016 DIPLA, 011/2016 SEST and 083/2016 ENAP), Project "EAGER: USBRCCR: Collaborative: Securing Networks in the Programmable Data Plane Era" funded by NSF and RNP (Brazilian National Research Network), and the Institutional Security Office of the Presidency of the Republic of Brazil (Grant 002/2017). Authors also thank Henrique Hirata for implementing the SSDP attack in a previously developed tool.

References

[1] Gondim, J. J. C., de Oliveira Albuquerque, R., Clayton Alves Nascimento, A., García Villalba, L., Kim, T. H.: "A methodological approach for assessing amplified reflection distributed denial of service on the internet of things", in *Sensors*, vol. 16, n. 11, p. 1855, 2016.

[2] Pacheco, L. A. B., Gondim, J. J. C., Barreto, P. A. S., Alchieri, E.: "Evaluation of Distributed Denial of Service threat in the Internet of Things"', in *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, pp. 89-92, Oct. 2016.

[3] De Almeida, Marcone Pereira and De Sousa Júnior, Rafael Timóteo and García Villalba, Luis Javier and Kim, Tai-Hoon: "New DoS Defense Method Based on Strong Designated Verifier Signatures", in *Sensors*, vol. 18, n. 9, p. 2813, http://www.mdpi.com/1424-8220/18/9/2813, ISSN 1424-8220, DOI 10.3390/s18092813, 2018.

[4] Radware: "History of DDoS Attacks", https://security.radware.com/ddos-knowledge-center/ddos-chronicles/ddos-attacks-history/, accessed Apr. 2019.

[5] MIT Technology Review: "The first DDoS attack was 20 years ago. This is what we've learned since.",https://www.technologyreview.com/s/613331/the-first-ddos-attack-was-20-years-ago-this-is-what-weve-learned-since, accessed Apr. 2019.

[6] Paxson, V.: "An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks", in *ACM SIGCOMM Computer Commun. Rev.*, vol. 31, pp. 38–47, 2001.

[7] Rossow, C.: "Amplification Hell: Revisiting Network Protocols for DDoS Abuse", in *Proceedings of the 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, CA, USA*, pp. 23–26, February 2014.

[8] Case, J., Fedor, M., Schoffstall, M., Davin, J.: "Simple Network Management Protocol (SNMP) RFC 1157 (Historic)", *Internet Engineering Task Force (IETF)*, 1990.

[9] Case, J., McCloghrie, K., Rose, M., Waldbusser, S.: " Introduction to Community-Based SNMPv2 RFC 1901", *Internet Engineering Task Force (IETF)*, 1996.

[10] Arış, A.; Oktuğ, S.F.; Yalçın, S.B.Ö.: "Internet-of-Things security: Denial of service attacks", In *Proceedings of the 2015 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, Turkey*,pp. 903–906, May 2015.

[11] Goodin, D.: "Record-Breaking DDoS Reportedly Delivered by $> 145k$ Hacked Cameras", *Arstechnica*, http://arstechnica.com/security/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/, accessed on Oct 2016.

[12] Bright, P.: "Spamhaus DDoS Grows to Internet-Threatening Size", *Arstechnica*, http://arstechnica.com/security/2013/03/spamhaus-ddos-grows-to-internetthreatening-size/, 2013 (accessed on 12 May 2016).

[13] Prince, M.: "The DDoS That Knocked Spamhaus Offline (and How We Mitigated It)", *Cloudflare*, https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/, 2013 (accessed on May 2016).

[14] Seaman, C.: "Threat Advisory: Mirai Botnet", Technical report, *Akamai Technologies*, https://www.akamai.com/uk/en/resources/our-thinking/threat-advisories/akamai-mirai-botnet-threat-advisory.jsp, 2016 (accessed on Apr 2019).

[15] Anna-senpai: "Mirai Source Code", https://github.com/jgamblin/Mirai-Source-Code, 2016. (accessed on May 2018).

[16] Moore, H. D.: "Security Flaws in Universal Plug and Play", https://information.rapid7.com/rs/411-NAK-970/images/SecurityFlawsUPnP\%20\%281\%29.pdf, accessed on Sep. 2018.

[17] UPnP Forum: "UPnP Device Architecture 1.0", http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf, accessed on Sep. 2018.

[18] Majkowski, Marek: "Stupidly Simple DDoS Protocol (SSDP) generates 100 *Gbps* DDoS", *Cloudflare*, https://blog.cloudflare.com/ssdp-100gbps/, accessed on Sep. 2018.

[19] McDowell, M.: "Understanding Denial-of-Service Attacks; Technical Report", *US Department of Homeland Security*, 2009.

[20] Ali, F. "IP Spoofing. The Internet Journal", vol. 10, no. 4, *Cisco Press*, http://www.cisco.com/web/about/ac123/ac147/archivedissues/ipj10-4/104ip-spoofing.html, Dec 2007 (accessed on October 2015).

[21] Majkowski, M.: "How to receive a million packets per second", *Cloudflare*, https://blog.cloudflare.com/how-to-receive-a-million-packets/, 2015 (accessed on Dec 2018).

[22] Bjarnason, S.: "DDoS defences in the terabit era: Attack trends, carpet bombing", *APNIC*, https://blog.apnic.net/2018/12/04/ddos-defences-in-the-terabit-era-attack-trends-carpet-bombing/, Dec. 2018 (accessed on Dec. 2018).

[23] Imperva: "Attackers Use DDoS Pulses to Pin Down Multiple Targets, Send Shock Waves Through Hybrids", *Whitepaper*, 2017.

[24] Wireshark Foundation: *Wireshark*, https://www.wireshark.org, Wireshark Foundation. 2015. https://www.wireshark.org, 2019 (accessed on Mar 2019).

SVCP4C: A tool to collect vulnerable source code from open-source repositories linked to SonarCloud

Razvan Raducu, Gonzalo Esteban, Francisco J. Rodríguez, Camino Fernández
Grupo de Robótica. Universidad de León
Av. Jesuitas, s/n. 24007 León (Spain)
{rrad, gestc, fjrodl, camino.fernandez}@unileon.es

***Abstract*—There is a significant body of work to detect Buffer Overflow in the literature. From manual audition of the code to dynamic analyzers, many techniques have been proposed to find vulnerabilities. Particularly one of these techniques is Machine Learning, which has gained ground in this research field lately. By teaching a Machine Learning algorithm what a vulnerability looks like, the result can predict security threats without requiring human intervention. However, fulfilling such task requires the establishment of a proper dataset. The purpose of this paper is to present SVCP4C (SonarCloud Vulnerable Code Prospector for C), a bot written in Python for collecting vulnerable source code. The bot extracts files from open-source repositories linked to SonarCloud—an online tool that performs static analysis—, and tags in such all the lines that are vulnerable. This set of tagged files may later be used to extract features and to create training datasets for Machine Learning algorithms.**

***Index Terms*—Buffer Overflow, Vulnerability, SonarCloud, Bot, Source code, Repository**

Tipo de contribución: *Investigación original (límite 8 páginas)*

I. Introduction

Research on buffer overflow, also known as buffer overrun, has a long tradition since it was first understood and documented in 1972 by James P. Anderson on behalf of the electronic systems division [1]. On November 1988 the, then very small, Internet suffered what is publicly known as "the first successful buffer overflow exploitation", taking advantage of the absence of buffer range checking in one of the functions used by fingerd daemon [2], [3]. Ever since Aleph One published in 1996 the first step-by-step article about stack-based buffer overflow exploitation [4], its popularity kept rising. In 2000, buffer overflow was declared the vulnerability of the decade [5]. Unsurprisingly enough, nowadays there is enough evidence to call it the most occurring vulnerability in the last quarter century [6].

Buffer overflow not only has been the most reported vulnerability in the last 25 years, it is also the most reported vulnerability with high severity and the most reported vulnerability with critical severity as well. The United States Industrial Control Systems Cyber Emergency Response Team stated that two out of four most reported vulnerabilities were buffer overflow, both stack and heap based [7]. Reports like the one published by WatchGuard [8] conclude that out of all the attacks they registered, the top four are all different instances of nothing but buffer overflow. Moreover, there are many types of buffer overflow attacks like write attacks, data manipulation/corruption attacks or read attacks [9], just to mention some. One of the posible reasons behind this is C being inherently unsafe: arrays and pointer references are not automatically bounds-checked thus relegating security to the programmer's skills. Furthermore, many of the standard C library functions, such as `gets()`, `scanf()` or `strcpy()`, just to mention some, are vulnerable as well [10], [11], [12]. All of this leaves us with a single and clear conclusion: buffer overflow is still a ceaseless vulnerability and it seriously jeopardizes nowadays security.

At this point it is clear enough that prevention and defense mechanisms are much needed in order to deal with such a menace to security. Fortunately for security-aware developers as well as for end users, much resources have been destined to research in this field. As a result, several different approaches have been proposed: manual audition of code, static analyzers, compiler and hardware modifications, dynamic analyzers and, more recently, the use of machine learning and derivative techniques.

Manual auditing code in order to find vulnerabilities is a difficult, error-prone and time-consuming task. Besides, it relies on people competent enough to efficiently detect vulnerabilities, which is equally difficult [13]. However, manually reviewing the code can be complemented with static analyzers which automatically review the code identifying potential security holes. One such static analyzer is SonarCloud[1], a collaborative platform that helps developers write secure and clean code. It supports many different languages and it is free if the project to analyze is open-source. There are, of course, many other static analyzers such as ITS4, a static C and C++ source code scanner that splits the code into lexical tokens to further apply pattern matching [14]. The MIT Lincoln Laboratory exhaustively analyzed several static tools [15].

Another way of preventing applications from suffering buffer overflow is using programming languages that natively perform bounds checking such as Java or Pascal. Those languages, on the contrary, lack low-level manipulation that C offers. With these limitations in mind, researchers have developed "safe dialects of C" that natively perform several security maneuvers. Unfortunately, those security checks, like bounds checking, involve up to 100% overhead [16]. Another approach consists in re-compiling it with security-aware modified compilers. Stackguard is one example of it. It prevents stack-based buffer overflow attacks by inserting canaries into the stack [17]. Nevertheless, when source code is not available, the previously mentioned techniques are useless and another approach is needed.

Regarding these approaches, many dynamic analyzers have been proposed—also known as runtime solutions. One of

[1] https://sonarcloud.io/about

these solutions is presented by Fraser, Badger & Feldman [18]. In their work, the authors defined what they call Generic Software Wrappers, which are protected non-bypassable kernel-resident software extensions for augmenting security without modifying the software. On the other hand, Goldberg, Wagner & Brewer [19] proposed Janus, a process that observes and mediates behavior by monitoring system calls. Naccio [20] transforms programs according to a safety policy. Something very similar to Naccio was proposed by Erlingsson and Schneider [21]. The authors called it SASI and it enforces security policies by modifying object code for a target system before that system is executed. There are way more proposed solutions, the previous ones have been mentioned just for the sake of the example.

In the same vein, Prasad and Chiueh [22] proposed a mechanism of re-writing windows portable executable binaries so that they include return address protection mechanisms in order to preserve the integrity of the stack. Another strategy to deter buffer overflow consists in modifying some aspect of the Operating System (OS). We find OS modifications in Unix-like systems with different distributions, the so-called distros. OpenBSD is one example of security-focused distro. In Windows universe we see OS modifications via the release of different Service Packs. Libsafe [23] is another example of modifying some OS aspect, in this case a system library modification in order to secure known vulnerable functions. Yet another approach to prevent buffer overflow is hardware modification. McGregor et al. [24] proposed a modified processor that includes a secure return address stack. It provides built-in, dynamic protection against return address tampering without requiring any effort by users or application programmers. Özdoganoglu et al. [25] proposed SmashGuard that, just like the previous example, includes a small hardware stack in which each function call instruction pushes the return address and the current stack frame pointer. In both cases the performance impact is negligible for most applications. These are mere examples of hardware prevention of buffer overflow.

Last but not least, in recent years the use of machine learning (ML) for vulnerability discovering experienced a pronounced growth. The problem of finding software vulnerabilities seems well-suited for ML systems. Per-line inspection is a tedious and tiresome job. The exact kind of job that computers excel at. With the application of ML we simply have to teach the computer what a vulnerability looks like [26]. A lot of efforts are being directed to this very topic [27], [28], [29], [30], [31], [32]. In addition to all this, recent research on the execution of ML algorithms in GPGPU hardware, and not mainstream CPU, shows a great improvement in performance [33], [34], [35], [36], [37], [38]. Specifically, research on buffer overflow detection using ML was already made and more is in the makings [39], [40], [41]. We believe all the effort that is lately focused on vulnerability detection using ML is enough for us to contribute to the cause.

In light of the above, there is nowadays a lack of publicly available datasets containing snippets of real vulnerable code that are suitable for ML. To fill the gap, this paper introduces SVCP4C (*SonarCloud Vulnerable Code Prospector For C*), a bot written in Python for gathering source code repositories available through SonarCloud. Particularly, it collects files linked to open-source repositories that are both written in C and targeted as vulnerable by the static analyzer.

The paper details how the bot works using the following structure. Section II presents an overview of SVCP4C, i.e. where is framed and how is related to SonarCloud. In Section III the technical behaviour of the bot will be detailed. Then, Section IV discusses the limitations encountered during the development process along with some future enhancments. Finally, Section V concludes summing up the work presented.

II. Methodology

SVCP4C finds its existence in two main reasons. First of all, it is part of a bigger project that is being carried out at the moment this document is written. The idea is to create a tool that automatically parses source code extracting different characteristics from the same. It then exports the data to some concrete file format that is adequate for a ML algorithm in order to predict possible buffer overflow vulnerabilities. Formally it is a static vulnerability analysis tool that, based on Abstract Syntax Trees (AST) and Control Flow Graphs (CFG) generated by Clang, models possible present buffer overflow vulnerabilities via source code inspection. Clang is an LLVM front-end for the C language family [42]. The modelling of the vulnerability is inspired by previous researches such as the works presented in [41], [43], [44], [39], [45]. The resultant tool is foreseen to be called TOOBAD4ML (*TOOl to Buffer overflow Analysis and Description FOR Machine Learning*) and a very simple conceptual diagram can be seen in Figure 1. Further discussion about TOOBAD4ML is outside this paper's scope.

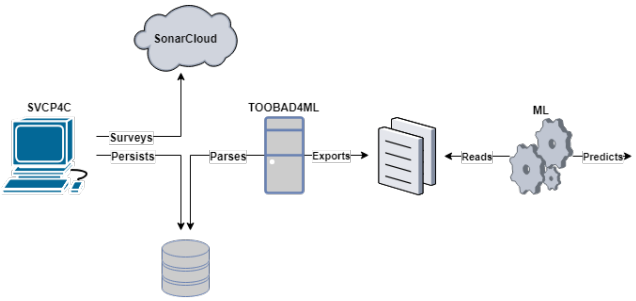


Figure 1. TOOBAD4ML conceptual diagram.

Second, but not less important, we need to somehow gather vulnerable code in order to train the ML algorithm. It is important to have enough balanced data in order to avoid unbalancing the algorithm, overfitting and many other problems that may arise and thus affect the final result [46]. Collecting vulnerable source code via SonarCloud helps us to obtain samples of real vulnerable code and, as we will later discuss, setting the queries parameters accordingly we can also obtain non-vulnerable code. Furthermore, the need of a real vulnerable code repository, that is, vulnerable code dataset, is justified by our prior research. According to our results, only Kratkiewicz and Lippmann [43] offer a vulnerable dataset that is publicly available. The main drawback is its artificial code, which may not be valid for testing with ML algorithms. Even though their dataset is labeled and might be good enough, we need real code from real applications.

A. SonarCloud Web API

SVCP4C is made to completely communicate with and depend on SonarCloud's REST API. The API documentation can be found in SonarCloud's official site[2]. Working with the API is fairly simple: one must make HTTP GET requests to a certain url with certain url parameters and SonarCloud responses with a JSON formatted body. The API is publicly available, i.e., it is free to use. The API offers a lot of functionality and much and diverse information can be obtained. We will focus exclusively on the functionality that serves our purpose, the functionality that is used by SVCP4C. SonarCloud's API has several main services and other internal services, as they call it. Internal services are those who must be used at one's own risk since they are subject to change or removal without previous notification. SVCP4C uses only 3 main services, which are:

1) `/api/components/search_project`
2) `/api/issues/search`
3) `/api/sources/raw`

SonarCloud also offers a Graphical User Interface (GUI) version of which everyone can make use via their website. It is important to mention the GUI because it runs the same API. That is, when somebody uses the GUI, the website is actually requesting to the webservices and parsing the JSON response in order to draw what the user is requesting. Figure 2 shows an example of SonarCloud GUI version reporting a vulnerability type issue because unsafe `strcpy()` function is in use. All SonarCloud responses are GUI oriented. This means, as we will later discuss, when requesting issues via API's HTTP GET requests, the response will locate the issue in the code but only function's name and not full signature, because only function's name would have been drawn. Not reporting the full function's signature it is yet another challenge TOOBAD4ML must deal with in its parsing process since function's arguments are really important. Once again, this discussion is out of the scope of this paper.

```
if(sizeof(argv[1]) < sizeof(buf)){
        strcpy(buf, argv[1]);
```

Figure 2. SonarCloud detecting use of insecure function.

III. Results

Algorithm 1 contains the pseudocode that sums up the steps carried out by the bot. Please keep in mind that if you reproduce the code or execute SVCP4C, results may differ since it entirely depends on SonarCloud's Elastic Search indexing. Some parameters that appear in Algorithm 1 require further contextualization. These parameters are:

- *p*. Represents page number. SonarCloud responds at most 500 results per page. If one query generates more than 500 results, `p` will be pre-incremented and the web service is requested again.
- *ps*. Represents page size. SonarCloud allows users to specify how many results they want to see per page, in our case per HTTP response. `ps` is a constant with 500 as value, since we want to retrieve as much information as possible per request.
- *remainingResults*. Represents how many results are left. That is, if the query generated more than 500 results, `remainingResults` is checked in order to request again.

SVCP4C performs several HTTP requests to SonarCloud's REST API. Thanks to these requests the projects ids, the files ids and, finally, the own source code can be obtained. We consider the performed requests are worth explaining in further detail. The requests we are talking about are represented in steps 5, 16 and 29 from Algorithm 1.

A. Step 5 – Request project ids

The request performed at this point is the one retrieving the ids of the projects that meet our filtering conditions. The filtering, as it was earlier mentioned, is carried out by SonarCloud's API via url parameters. The requested web service is `/api/components/search_projects` and the parameters are:

- *filter*. `security_rating>=2` and `languages=c`
- *p*. `p=i` (ith-page)
- *ps*. `ps=500` (current page size)

As shown in Algorithm 1, `p` is number of page, `ps` is page size and `security_rating` $\geq$ `2` implies a *B* security rating according to the analysis performed by SonarCloud. Different SonarCloud's metrics and ratings can be found at its website [47]. *B* security rating means "at least one Minor Vulnerability". A security rating corresponds to non-vulnerable code and is represented via `security_rating=1` in the HTTP request. This is what must be used in order to obtain non-vulnerable source code according to SonarCloud. The resultant queried url is:

https://sonarcloud.io/api/components/search_projects?ps=500&p=1&filter=security_rating%3E%3D2+and+languages%3Dc

At the time this document was written the query generates 447 results. This is, there are only 447 open-source projects written in C that have at least one minor vulnerability.

B. Step 16 – Request files info

At this point of execution, the HTTP request is performed in order to obtain the unique identifier of every vulnerable source file within each previously queried project. The requested web service is `/api/issues/search` and the parameters are:

- *projects*. `projects=1,2,3` (a list of all project ids previously queried, comma separated).
- *types*. `types=VULNERABILITY` (SonarCloud issue category).
- *languages*. `languages=c` (a list of program languages, comma separated).
- *p*. `p=i` (ith-page).
- *ps*. `ps=500` (current page size).

The types parameter is used to specify what issue we are looking for, i.e. return unique identifier of source files affected only by the specified type of issue. There are four types of issues SonarCloud detects: `CODE_SMELL`, `BUG`, `VULNERABILITY`, `SECURITY_HOTSPOT` [48]. The remaining parameters have already been introduced. There are multiple resultant urls queried.

[2]https://sonarcloud.io/web_api

Algorithm 1. SVCP4C's pseudocode.

```
Check user arguments and options;
If (path from step 1 doesn't exist) then
    Create path;
Otherwise
    Abort with error;
Set p := 1 and remainingResults := 0;
Procedure. Request project ids():
    HTTP GET request (url, params);
    Retrieve all HTTP response payload from step 8 as JSON;
    Update remainingResults and jump to step 12;
end_procedure;
If (remainingResults > ps) then
    If (p == 20) then
        Jump to step 19;
    Pre-increment p;
    Jump to step 7;
Otherwise
    Jump to step 19;
Obtain all project ids from step 9 and set p := 1 and remainingResults := 0;
Procedure. Request files info():
    HTTP GET request (url, params);
    Retrieve all HTTP response payload from step 21 as JSON;
    Write results of step 22 to file;
    Jump to step 34;
    Update remainingResults and jump to step 27;
end_procedure;
If (remaining query results > ps) then
    If (p == 20) then
        Jump to step 50;
    Pre-increment p;
    Jump to step 20;
Otherwise
    Jump to step 50;
Open file from step 23 and parse its JSON formatted content;
For each (issue (key,value) from results of step 34) do:
    Retrieve the value of component key
    HTTP GET request (url,params)
    If (response from step 37 contains errors) then
        Print message notifying the file was skipped because there was an error;
    Otherwise
        Go to step 42;
    Obtain name of file to be persisted based upon the naming policy;
    If (file with name from step 42 does not exist) then
        Create file and append at the end the separator comment line;
    Otherwise
        Jump to step 47;
    Append the vulnerable line from step 35;
    Jump to step 25;
end_foreach;
end_program.
```

C. Step 29 – Retrieve value of component key

This is the last query SVCP4C performs and it is the one that obtains the actual vulnerable source code. For each of the unique source file ids obtained from the previous query, SonarCloud is requested to provide the corresponding source code. The requested web service is `/api/sources/raw` and the parameters is:

- *key*. The unique identifier of the file whose code is about to be retrieved.

After source code is retrieved, a vulnerable line is appended at the end of the code in the format of: "`///sl,so;el,eo`"—without quotes—, where `sl` is starting line, `so` is starting offset, `el` is ending line and `eo` is ending offset.

IV. Discussion

During the development process we have faced many restrictions related to SonarCloud's API, which have raised some future improvements.

A. Constraints related to the SonarCloud web API

One particular constraint SonarCloud's API imposes is what is generally called the 10000 issue limit. That is, every single request made to `/api/issues/search` will be limited to 10000 results. As their prior, and now deprecated, documentation page states: "If the number of issues is greater than 10000, only the first 10000 ones are returned by the web service" [49]. Even though the quoted sentence comes from an older documentation version, the limit still applies nowadays despite being undocumented. There are many questions in different forums and platform from users just like us asking about this very same limit.

Another drawback we found using SonarCloud is that we cannot filter based on vulnerability type; although we can indeed filter based on issue type. We would like to have that second-level filtering feature. That would ease TOOBAD4ML's parsing job because right now, the way it is, we are creating a dataset of all kind of vulnerabilities not only buffer overflow, even though we are only interested in that one.

Up to this point, during development we faced some problems whose solution(s) can be directly seen in Algorithm 1. For example, we found out that we cannot just append every result of the queries asking for vulnerabilities into one single file because the result is a mal-formatted JSON. That is because SonarCloud sends JSON objects as response and, as such, these include the opening and closing square brackets. A JSON file in order to be well-formatted must include a single JSON object, that is, a single pair of closing and opening square brackets. Since we were appending the results into the very same file, we were automatically mal-formatting it. The solution we adopted is quite simple. We request the first 500 results (page 1) and write them to a file. Immediately after, we parse the file and request the corresponding source code. When we got it we request the next 500 (page 2) vulnerabilities, write (not append) them to a file and, once again, request the source code. This loop goes on until we reach the 10000-results limit imposed by SonarCloud. This is reflected in steps 23 and 34 of Algorithm 1.

Another problem we faced was that, after some tests, we found out each vulnerable code line is a different entry from the same JSON list. That is, we could have 13 different results, 13 different issues, with them being just different vulnerable lines of the same source file. If we simply download the file that has a vulnerability, in the previous case we would end with 13 copies of the same file. The solution is, once again, quite simple. We compile all vulnerable lines that refer to the same file, download the file and append the lines as a comment at the end of the file. This represents step 47 in Algorithm 1.

Finally, we encountered problems with nonexistent files. Apparently SonarCloud maintains a list of issues even though the file those issues arise from got deleted long time ago. The result is a file whose sole content is a JSON list called "errors" containing "msg" keys. The solution is pretty straightforward, we inspect the content that is about to be written out in the corresponding file and, if it contains an "errors" JSON list, we skip it. This can be seen in step 39 of Algorithm 1.

B. Future enhancements

First of all, we do not consider as a feasible improvement the imposed 10000 query results by SonarCloud since modifying it is out of our control and it is unlikely to change in the near future. However, there are some things that we can indeed improve, for instance parallelize HTTP requests. As of right now, SVCP4C sends one query, waits for the response and then sends the following one. There is a huge gap in performance when using sequential requests. There are several solutions for parallelizing HTTP requests in Python. On the other hand, we are completely aware of what implications and difficulties may arise. For example, with parallel requests come parallel responses hence persistence becomes a critic operation which shall involve synchronization mechanisms. Moreover, asynchronous requests imply receiving responses in no particular order.

Another improvement we have been thinking of is complementing SonarCloud detection in terms of well-known vulnerable functions that SonarCloud skips. This could be achieved by means of own vulnerable functions dictionaries. In order to illustrate this situation, we have tested some trivial code. SonarCloud successfully detects the possible overflowing of the buffer that may occur in the code shown in Figure 3 by reporting the use of unsafe functions. SonarCloud wisely recommends the use of a width specifier for the corresponding placeholder. It is, in addition, stated in the Common Weakness Enumeration (CWE) [50]. However, as soon as the developer places the width specifier, SonarCloud assumes it is correct without, thus, further inspecting. We consider this assumption both critical and harmful since a self-induced buffer overflow may arise for various reasons as human error, for example. Figure 4 illustrates the situation where the width specifier is bigger than the actual buffer where data will be copied to. In this case the difference in size is evident but there could happen a way more subtle off-by-one buffer overflow.

```
char org[15];
printf ("Enter your organization: ");
scanf ("%s", org);
```

Figure 3. SonarCloud successfully reporting a possible buffer overflow.

```
char org[15];
printf ("Enter your organization: ");
scanf ("%30s", org);
printf("Organization typed: %s\n", org);
```

Figure 4. SonarCloud assumes width specifier is correct.

Yet another improvement we need is detecting full range of function in order to ease TOOBAD4ML function parsing task. This is, to detect complete function's signature, not only function's name and thus appending the corresponding comment at the end of the file. Let us remember that SonarCloud's responses simply consist of starting and ending line and starting and ending offset (column). This way of telling users where the vulnerability is may seem correct but, in reality, it is not since it is incomplete. SonarCloud simply returns what it would have drawn in case the request was made using the GUI. We say this solution is not enough since functions may have variable number of arguments, spread across multiple lines, amongst others. Of course, there is not an easy solution for this problem.

Finally, in order to surpass the 10000 issues limit we could, assuming no project has more than 100000 issues, request the issues project by project. As of right now, SVCP4C retrieves all the project ids that meet our filtering criteria and requests the issues of all the ids altogether because the API allows this operation. This will, obviously, affect the performance since we would go from a single HTTP GET request specifying project ids to a HTTP GET request per project id.

V. Conclusions

Buffer Overflow has been one of the most reported vulnerability for decades. So far, prevention and defense mechanisms are much needed to prevent any security threat. Particularly, auditing code, static analyzers or ML are among the techniques used today to counteract Buffer Overflows. Furthermore, academic literature shows that there are many efforts towards the detection of software vulnerabilities with ML. However, there is a lack of publicly available datasets containing snippets of real vulnerable code.

In this work we presented a tool named SVCP4C. That is, a bot to collect real vulnerable code from open-source repositories available through SonarCloud. The paper has described its technical behaviour along with its constraints and future enhancements. SVCP4C belongs to a greater project named TOOBAD4ML, a tool aimed to extract features of Buffer Overflow vulnerabilities from a given source code. Having said that, SVCP4C plays an initial role on such project, as it is meant to provide a basis for creating datasets of real vulnerable source code. In order to achieve that, the tool tags the lines identified by SonarCloud so they can be quickly retrieved. Although the development is in an early stage, its source code[3] and the corresponding sample output[4] are publicly available on our GitHub repository.

Acknowledgements

This work has been partially funded by the Addendum no. 4 to the Universidad de León-Instituto Nacional de Ciberseguridad (INCIBE) Convention Framework about the "Detection of new threats and unknown patterns", by the Consejería de Educación de la Junta de Castilla y León through the Project LE028P17 about the "Development of reusable software components based on machine learning for the cybersecurity of autonomous robots" and by the Ministerio de Ciencia, Innovación y Universidades through the Project RTI2018-100683-B-I00.

References

[1] J. P. Anderson, "Computer security technology planning study," ESD-TR-73-51, Tech. Rep., 1972.

[2] M. Eichin and J. Rochlis, "With microscope and tweezers: an analysis of the Internet virus of November 1988," in *IEEE Symposium on Security and Privacy*, 2003, pp. 326–343.

[3] E. H. Spafford, "The internet worm program: an analysis," *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 1, pp. 17–57, 2004.

[4] A. One, "Smashing the stack for fun and profit," *Phrack magazine*, vol. 7, no. 49, pp. 14–16, 1996.

[5] C. Cowan, P. Wagle, C. Pu, S. Beattie, and J. Walpole, "Buffer overflows: Attacks and defenses for the vulnerability of the decade," *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, pp. 119–129, 2000.

[6] Y. Younan, "25 Years of Vulnerabilities: 1988-2012," Sourcefire Vulnerability Research Team, Tech. Rep., 2013.

[7] NCCIC, "ICS-CERT Annual Vulnerability Coordination Report," Tech. Rep., 2016.

[8] "Internet Security report," WatchGuard, Tech. Rep., 2017.

[9] N. Tuck, B. Calder, and G. Varghese, "Hardware and binary modification support for code pointer protection from buffer overflow," in *Proceedings of the 37th Annual International Symposium on Microarchitecture, MICRO*, 2004, pp. 209–220.

[10] D. Wagner, J. S. Foster, E. A. Brewer, and A. Aiken, "A First Step Towards Automated Detection of Buffer Overrun Vulnerabilities," in *Network and Distributed System Security Symposium*, 2000, pp. 3–17.

[3] https://github.com/uleroboticsgroup/SVCP4C

[4] https://github.com/uleroboticsgroup/SVCP4CDataset

[11] R. C. Seacord, *Secure Coding in C and C++*. Pearson Education, 2005.

[12] JTC 1/SC 22/WG 14, "ISO/IEC 9899:1999: Programming languages – C," International Organization for Standards, Tech. Rep., 1999.

[13] C. Cowan, "Software security for open-source systems," *IEEE Security and Privacy*, vol. 1, no. 1, pp. 38–45, 2003.

[14] J. Viega, J. T. Bloch, Y. Kohno, and G. Mcgraw, "ITS4: A static vulnerability scanner for C and C++ code," in *Proceedings 16th Annual Computer Security Applications Conference (ACSAC'00)*. IEEE, 2000, pp. 257–267.

[15] M. Zitser, R. Lippmann, and T. Leek, "Testing Static Analysis Tools using Exploitable Buffer Overflows from Open Source Code," in *SIGSOFT '04/FSE-12 Proceedings of the 12th ACM SIGSOFT international symposium on Foundations of software engineering*, 2005, pp. 97–106.

[16] T. Jim, J. G. Morrisett, D. Grossman, M. W. Hicks, J. Cheney, and Y. Wang, "Cyclone: A safe dialect of c." in *USENIX Annual Technical Conference, General Track*, 2002, pp. 275–288.

[17] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, A. Grier, P. Wagle, Q. Zhang, B.-o. Attacks, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, and Q. Zhang, "StackGuard : Automatic Adaptive Detection and Prevention of buffer-overflow attacks," in *Proceedings of the 7th USENIX Security Symposium San Antonio, Texas*, 1998.

[18] T. Fraser, L. Badger, and M. Feldman, "Hardening cots software with generic software wrappers," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, vol. 2. IEEE, 2000, pp. 323–337.

[19] I. Goldberg, D. Wagner, and E. A. Brewer, "A Secure Environment for Untrusted Helper Applications Con ning the Wily Hacker 2 Motivation 1 Introduction," in *Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, no. July, 1996.

[20] D. Evans and A. Twyman, "Flexible policy-directed code safety," in *Proceedings - IEEE Symposium on Security and Privacy*, vol. 1999-January, 1999, pp. 32–45.

[21] Ú. Erlingsson and F. B. Schneider, "SASI enforcement of security policies," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*. IEEE, 2000, pp. 287–295.

[22] M. Prasad and T. C. Chiueh, "A binary rewriting defense against stack based buffer overflow attacks," in *USENIX Annual Technical Conference, General Track*, 2003, pp. 211–224.

[23] B. Arash, S. Navjot, and T. Timothy, "Transparent Run-Time Defense Against Stack-Smashing Attacks," in *Proceedings of the General Track: 2000 USENIX Annual Technical Conference*, 2000, pp. 251–262.

[24] J. P. McGregor, D. K. Karig, Z. Shi, and R. B. Lee, "A processor architecture defense against buffer overflow attacks," in *Proceedings, ITRE 2003 - International Conference on Information Technology: Research and Education*, 2003, pp. 243–250.

[25] H. Özdoganoglu, T. N. Vijaykumar, C. E. Brodley, B. A. Kuperman, and A. Jalote, "SmashGuard: A hardware solution to prevent security attacks on the function return Address," *IEEE Transactions on Computers*, vol. 55, no. 10, pp. 1271–1285, 2006.

[26] B. Schneier, "Machine learning to detect software vulnerabilities," https://www.schneier.com/blog/archives/2019/01/machine_learnin.html, 2019.

[27] J. J. Kronjee, "Discovering vulnerabilities using data-flow analysis and machine learning Demonstrated for PHP applications," Master's thesis, Open Universiteit Nederland, 2018.

[28] L. Zhao, Z. Chen, and Q. Jia, "Summary of vulnerability related technologies based on machine learning," *AIP Conference Proceedings*, vol. 1955, no. April 2018, pp. 1–5, 2018.

[29] J. Jurn, T. Kim, and H. Kim, "An automated vulnerability detection and remediation method for software security," *Sustainability*, vol. 10, no. 5, pp. 1–12, 2018.

[30] J. A. Harer, L. Y. Kim, R. L. Russell, O. Ozdemir, L. R. Kosta, A. Rangamani, L. H. Hamilton, G. I. Centeno, J. R. Key, P. M. Ellingwood *et al.*, "Automated software vulnerability detection with machine learning," *arXiv preprint arXiv:1803.04497*, 2018.

[31] T. Abraham and O. De Vel, "A Review of Machine Learning in Software Vulnerability Research," https://www.dst.defence.gov.au/sites/default/files/publications/documents/DST-Group-GD-0979.pdf, 2017.

[32] B. Chernis and R. Verma, "Machine learning methods for software vulnerability detection," in *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*. ACM, 2018, pp. 31–39.

[33] D. Steinkraus, I. Buck, and P. Simard, "Using gpus for machine learning algorithms," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 2005, pp. 1115–1120.

[34] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[35] P. Li, Y. Luo, N. Zhang, and Y. Cao, "Heterospark: A heterogeneous cpu/gpu spark platform for machine learning algorithms," in *2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*. IEEE, 2015, pp. 347–348.

[36] N. Lopes and B. Ribeiro, "Gpumlib: An efficient open-source gpu machine learning library," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 355–362, 2011.

[37] D. L. Ly, V. Paprotski, and D. Yen, "Neural networks on gpus: Restricted boltzmann machines," http://www.eecg.toronto.edu/moshovos/CUDA08/doku.php, 2008.

[38] G. Wu, J. L. Greathouse, A. Lyashevsky, N. Jayasena, and D. Chiou, "Gpgpu performance and power estimation using machine learning," in *2015 IEEE 21st international symposium on high performance computer architecture (HPCA)*. IEEE, 2015, pp. 564–576.

[39] J. Durães and H. Madeira, "A Methodology for the Automated Identification of Buffer Overflow Vulnerabilities in Executable Software Without Source-Code," in *Lecture Notes in Computer Science (Dependable Computing)*. Springer Berlin Heidelberg, 2005, vol. 3747, pp. 20–34.

[40] G. Grieco and A. Dinaburg, "Toward Smarter Vulnerability Discovery Using Machine Learning," in *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, 2018, pp. 48–56.

[41] Q. Meng, C. Feng, B. Zhang, and C. Tang, "Assisting in Auditing of Buffer Overflow Vulnerabilities via Machine Learning," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–13, 2017.

[42] "Clang c language family frontend for llvm," https://clang.llvm.org, 2019.

[43] K. Kratkiewicz and R. Lippmann, "Using a Diagnostic Corpus of C Programs to Evaluate Buffer Overflow Detection by Static Analysis Tools," in *Workshop on the Evaluation of Software Defect Detection Tools*, Chicago, IL, 2005, p. 19.

[44] B. M. Padmanabhuni and H. B. K. Tan, "Predicting Buffer Overflow Vulnerabilities through Mining Light-Weight Static Code Attributes," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*. IEEE, nov 2014, pp. 317–322.

[45] M. Bishop, S. Engle, D. Howard, and S. Whalen, "A Taxonomy of Buffer Overflow Characteristics," *IEEE Transactions On Dependable And Secure Computing*, vol. 9, no. 3, pp. 305–317, 2012.

[46] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data," *ACM SIGKDD explorations newsletter - Special issue on learning from imbalanced datasets*, vol. 6, no. 1, pp. 20–29, 2004.

[47] SonarSource, "Metric Definitions — SonarCloud Docs," https://sonarcloud.io/documentation/user-guide/metric-definitions/, 2019.

[48] ——, "SonarCloud Web API – /api/issues," https://sonarcloud.io/web_api/api/issues, 2019.

[49] ——, "SonarQube Docs – /api/issues," https://docs.sonarqube.org/pages/viewpage.action?pageId=239218, 2014.

[50] CWE, "CWE – CWE-120: Buffer Copy without Checking Size of Input ("Classic Buffer Overflow") (3.2)," https://cwe.mitre.org/data/definitions/120, 2019.

Cybersecurity on Brain-Computer Interfaces: attacks and countermeasures

Sergio López[1], Alberto Huertas[2], Gregorio Martínez[1]

[1]*University of Murcia*, Murcia, Spain. Email: {slopez, gregorio}@um.es
[2]*Waterford Institute of Technology*, Waterford, Ireland. Email: ahuertas@tssg.org

***Abstract*—In recent years, Brain-Computer Interfaces (BCI) have increased their presence in the medical field as well as in other sectors of the industry such as entertaining or authentication. This expansion has improved not only the subjects' quality of life but also their quality of experience when using entertainment systems. Despite the benefits, new paradigms such as Brain-to-Internet or Brain-to-Brain, together with novel technologies and techniques missing security and privacy by design principles are influencing the emergence of cybersecurity challenges affecting subjects' safety and data privacy. In this context, this line of work aims to review the attacks on BCI disrupting physical safety, data availability, confidentiality and integrity, as well as propose proactive and reactive countermeasures to enable their protection.**

***Index Terms*—Brain-computer Interfaces, BCI, cybersecurity, privacy, safety**

Tipo de contribución: *Investigación ya en desarrollo*

I. Introduction

Brain-computer interfaces (BCI) emerged in the 1970s with the goal of acquiring and processing users' brain activity to later perform specific actions over external machines or devices [1]. Fig. 1 shows the most relevant stages of the common BCI functioning cycle, where the neural activity, often influenced by actions such as body movements, is acquired and processed to later perform particular actions through external applications. However, after several decades of research, this functionality has been extended by enabling not only neural activity recording but also stimulation. One of the first solutions of BCI was developed at the end of the 1990s, producing a major advance in the medical industry, specifically in neurorehabilitation, and bringing to the reality the mental control of prosthetic limbs and wheelchairs. After this notorious achievement, technology and artificial intelligence are playing a key role in the evolution of BCI by providing novel acquisition and stimulation devices as well as intelligent processing platforms. In this context, futuristic applications such as brains connected to the Internet, or interconnected networks of brains, also known as *brainets*, are rising to share knowledge, memories or thoughts between people.

Despite the benefits provided by the previous evolution, the application of BCI in the medical field generate concerns in terms of patients' physical safety, as attacks put their integrity and lives at risk. In addition, the Internet of Things, new applications such as Brain-to-Brain (BtB), and the lack of security-by-design or privacy-by-design approaches have influenced the emergence of critical cybersecurity challenges. Among them, we highlight attacks and concerns affecting the confidentiality of sensitive information managed by BCI like, for example, thoughts, memories or emotions; the integrity of data, decisions, and actions considered by BCI applications such as robotic limbs; the availability of the data and services managed by BCI; the physical safety of patients suffering neurodegenerative diseases treated by BCI; and the automatic detection and mitigation of the previous issues in real time.

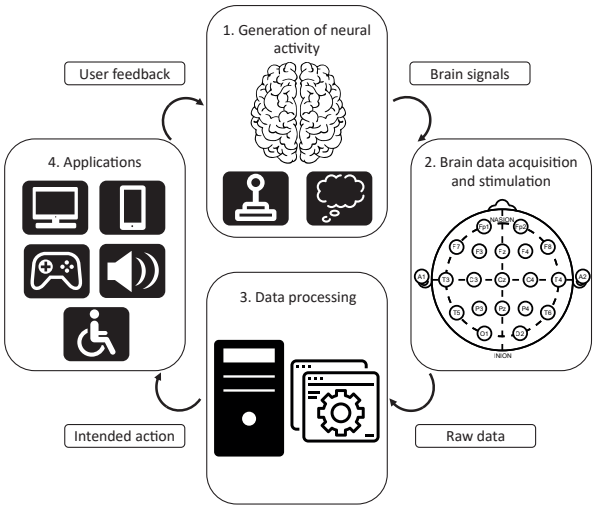


Fig. 1. General functioning of the BCI cycle for information acquisition

In order to deal with the previous challenges, this research line focuses on reviewing the literature and analysing the security threats and countermeasures documented for each phase making up the BCI cycle. In this sense, Section II summarises the most important cyberattacks affecting data integrity, confidentiality, availability, and physical safety, as well as the existing countermeasures to mitigate them. In addition, we have detected new attacks that can result in new opportunities for attackers, proposing countermeasures to reduce their impact. The next steps of this research line will be oriented to propose detection and mitigation mechanisms to improve the protection of sensitive and private data, such as memories, thoughts or feelings.

II. Cyberthreats affecting BCI cycle and countermeasures

The functioning cycle of BCI for the brain acquisition process have been addressed in the literature in a heterogeneous and general fashion. Phase 1, *Signal acquisition*, focuses on the generation and acquisition of neural signals. This generation process is influenced by external stimuli, and it is dependant on the user's intention to perform a given

TABLE I
CYBERSECURITY AND SAFETY ATTACKS FOR EACH PHASE OF THE BCI CYCLE

Phase	Integrity	Confidentiality	Availability	Safety
1. Signal acquisition	(+) Malicious external stimuli: alter the acquired brain waves (-) Replay & spoofing attack: impersonate the legitimate brain waves	(-) Malicious external stimuli: acquire private neural information (e.g. thoughts, emotions or beliefs)	(-) Malicious external stimuli: disrupt brain waves generation (-) Noise attack: disrupt the acquisition process	(+) Integrity and safety attacks of phase 1 (e.g. disrupt the control of a wheelchair)
2. Preprocessing	(+) Malware attack: alter the acquired neural signals	(+) Malware attack: gather the acquired neural information	(+) Malware attack: disrupt the processing	(+) Integrity and safety attacks of phase 2 (e.g. alter neural speech assistants)
3. Feature extraction	(+) Malware attack: use of malicious features	(+) Malware attack: gain access to the features	(+) Malware attack: alter the extraction process to avoid the generation of the action	(+) Integrity and safety attacks of phase 3 (e.g. alter neural monitoring systems)
4. Classification	(+) Malware attack: send malicious actions to the app and to the ML model	(+) Malware attack: gain access to the ML model or the ML software	(+) Malware attack: avoid communication with the app and the ML system	(+) Integrity and safety attacks of phase 4 (e.g. alter mental writing assistants)
5. Output	(-) Man-in-the-middle attack: intercept and modify the action sent to BCI apps	(-) Eavesdropping attack: acquire information sent to external systems or devices	(+) Denial of service attack: suppress the output sent to BCI applications	(+) Malware attack: transmit dangerous actions to BCI applications
6. Applications	(-) Spoofing attack: create malicious applications (+) Security misconfiguration attack: unauthorised access (+) Buffer overflow attack: execute malicious commands	(+) Malware, injection, buffer overflow, security misconfiguration attack: acquire sensitive information managed by BCI applications	(+) Malware, injection, buffer overflow, security misconfiguration attack: denial of service over BCI applications	(+) Malware attack: generate physical damage (e.g. dangerous movements of prosthetic limbs)

task like, for example, the control of an external device. These raw analog signals gathered should be processed in order to identify the user's intention correctly. This process is performed in phase 2, where different steps are carried out: an analog-to-digital conversion to allow later processing of the information, the maximisation of the signal-to-noise ratio and the suppression of undesired artifacts. Subsequently, *Feature extraction* manages the information and extracts features that are classified in phase 4. This classification aims to predict the user's intention by the use of Machine Learning (ML) techniques. The output of this phase is the action intended by the user, and it is finally sent to applications, which can present optional feedback to the user to start new iterations of the cycle. Table I summarises the previous phases of the BCI cycle, highlighting existing integrity, confidentiality, availability and safety attacks and their impact. Attacks detected in the literature are preceded by a hyphen (-), while those identified by us are indicated with a plus symbol (+).

Considering the attacks of Table I and the impacts that they generate, we propose some countermeasures to mitigate them. Replay attacks based on synthetic brain signals can be restricted if BCI implement signal authentication that identify unique characteristics per user. Malware attacks can be mitigated by the use of antivirus solutions, as well as firewalls and IDS. In addition, man-in-the-middle and eavesdropping attacks can be reduced by using strong encryption mechanisms. On the other hand, attacks related to weak logical protection, as is the case of security misconfiguration, buffer overflow and injection attacks, can be avoided using safe APIs and libraries, and implementing access control systems that allow different security groups. Considering confidentiality issues, malicious external stimuli can be reduced if users are educated in the risks of these technologies. Furthermore, BCI devices should impose training sessions before using them, assuring that the users are aware of these risks. To limit confidentiality issues on applications, development APIs should restrict the transmission of raw neural information outside the BCI, avoid a complete control on the device and have limitations in the communication with external services. Availability issues are difficult to mitigate, as an attacker can send energy to the medium that prevent the electrodes to acquire the information properly. However, reactive approaches will be key components against these problems, being able to, for example, detect the attacks, change the frequency used and notify the user. Denial of service (DoS) attacks can also be mitigated by reactive systems, detecting the situation and applying dynamic countermeasures. Finally, a reduction of safety issues can be achieved by the improvement of the previous problems. In addition, the definition and implementation of robust security standards should be taken into consideration, to allow the development of secure and homogeneous devices.

III. CONCLUSIONS AND FUTURE WORK

Despite the great benefits of BCI, they are still immature in terms of cybersecurity protection mechanisms. In this context, we are conducting a comprehensive and systematic review of the literature to identify cyberthreats and concerns, aiming to make a contribution in this field of knowledge. As future work, we will consider the design and implementation of proactive and reactive detection procedures to self-adapt against BCI attacks. To accomplish that, specific security schemes and intelligent techniques based on artificial intelligence will be taken into account, considering necessary the validation of this alternative throw several use cases.

ACKNOWLEDGEMENTS

This work has been supported by the Irish Research Council, under the government of Ireland post-doc fellowship (grant GOIPD/2018/466)

REFERENCES

[1] Q. Li and D. Ding and M. Conti: "Brain-Computer Interface applications: Security and privacy challenges", *IEEE Conference on Communications and Network Security (CNS)*, pp. 663-666, 2015. DOI 10.1109/CNS.2015.7346884

Algoritmo de Interpolación Cromática para la Detección de Zonas Manipuladas de Imágenes Digitales

Esteban Alejandro Armas Vega, Luis Alberto Martínez Hernández, Sandra Pérez Arteaga,
Ana Lucila Sandoval Orozco, Luis Javier García Villalba*
Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: esarmas@ucm.es, {asandoval, javiergv}@fdi.ucm.es

***Resumen*—Aunque históricamente ha habido confianza en la integridad de las imágenes, el avance de la tecnología ha comenzado a erosionar esta confianza. Este documento propone un método de autenticación de imagen digital basado en el error cuadrático medio del patrón de interpolación CFA estimado a partir de la imagen analizada. Los resultados de los experimentos demuestran la eficiencia del método propuesto. Cada uno de estos experimentos se ejecutó utilizando diferentes conjuntos de datos públicos desarrollados con fines de investigación.**

***Index Terms*—Análisis Forense, Crominancia, Copiar-pegar, Detección de manipulaciones, Empalme, Error Cuadrático Medio, Imágenes Digitales, Matriz de Filtros de Color.**

I. Introducción

La popularización de las cámaras en los teléfonos móviles ha provocado una revolución en el mundo de la fotografía. Los teléfonos móviles han puesto al alcance de los apasionados de la fotografía, un medio accesible para iniciarse en este mundo. Sin embargo, ha provocado un gran aumento en el número de contenidos digitales con una mejor calidad. Este exceso de información convierte a la sociedad actual en la generación más informada pero sin los mecanismos necesarios para procesarla y filtrar los elementos no deseados. Debido a esto se ha perdido la capacidad de pensamiento crítico, aceptando cualquier información obtenida como verdadera sin cuestionar su origen.

La tecnología de estas cámaras sigue mejorando y aumentando sus prestaciones, aunque la cámara compacta, o réflex, seguirá manteniendo su uso en los segmentos más profesionales. Las cámaras móviles ha superado la cuota de mercado de las compactas, pero aún no han logrado desplazar a las DSLR (del inglés *Digital Single Lens Reflex*) o una buena CSC (del inglés *Compact System Camera*).

Una de las principales ventajas de las cámaras incrustada en teléfonos móviles sobre las cámaras tradicionales, es la instantaneidad de sus fotografías, pues no siempre se cuenta con una cámara réflex, pero siempre se cuenta con el teléfono móvil [1].

Las mejoras en el segmento de la tecnología de los teléfonos móviles (mejores cámaras, nuevas pantallas, conexión a Internet, etc.) cambian la forma de trabajar de muchos profesionales. Gracias a Internet y a estas tecnologías, estos dispositivos se han convertido en verdaderos centros multimedia, ocio y comunicación. Dentro del ámbito del Periodismo ha democratizado y mejorado su difusión. Gracias a los móviles, los comunicadores han pasado por una cambios significativos como la posibilidad de obtener imágenes captadas por los propios usuarios. Sin embargo, surgen dudas sobre la validez que pueden tener los mensajes de WhatsApp, fotos, SMS y demás contenido presente en un dispositivo móvil. La causa es variable y necesita un análisis en cada caso. Lo que sí se puede afirmar es que dicho contenido es perfectamente válido como prueba en un procedimiento judicial, siempre y cuando, a la hora de obtener dicha prueba se cumpla en todo momento la cadena de custodia. Una vez garantizado esto, es necesario analizar hasta qué punto dicha prueba, especialmente en el proceso penal, es suficiente para demostrar la inocencia o culpabilidad de un acusado.

Para que una prueba pueda ser considerada como evidencia es necesario que se cumpla con las reglas de cadena de custodia. Para ello, el origen de la prueba es fundamental, es decir, solo tendrá validez, aquella prueba obtenida mediante la orden de un juez y que esta sea facilitada por la propia empresa que "almacena" los datos (Twitter, Facebook, etc.) o el dueño del dispositivo que la contiene. No obstante, es necesario que esta disponga de una serie de elementos para comprobar su veracidad. Esto se debe a la facilidad con que dichas pruebas puedan ser modificadas (borrar o agregar algún elemento de la imagen, fusionar dos imágenes, etc.). También es posible que otra persona, haya interceptado la comunicación o dispositivo móvil para suplantar la identidad del usuario o que simplemente dichas pruebas hayan podido ser modificadas por un técnico informático.

Según STS 1415/ 2003, del 29 de Octubre, el derecho a la presunción de inocencia del art. 24.2 CE exige al Tribunal de instancia lo siguiente: Que exista una prueba con un contenido de cargo, que dicha prueba haya sido obtenida y aportada al proceso siguiendo las normas de la Constitución y de la Ley procesal y que la prueba de cargo sea razonable y considerada como suficiente para justificar la condena penal. Por tanto, cualquier medio de "prueba tecnológica" puede ser utilizado en un procedimiento judicial. Sin embargo, la misma puede no ser suficiente para condenar a un acusado. Por tanto, para utilizar cualquier medio de prueba disponible es tan importante como, demostrar que dichos medios de prueba, tengan validez suficiente para conseguir una condena

penal. Para conseguirlo, es fundamental que dicha prueba haya sido obtenida por la autoridad judicial correspondiente o que esté avalada por un perito forense. De ahí la importancia del análisis forense de imágenes digitales de dispositivos móviles en la actualidad.

El presente trabajo pretende de dar respuesta al problema de verificación de manipulación de imágenes digitales obtenidas con los dispositivos móviles. La gran cantidad de información digital dificulta enormemente verificar su autenticidad. Este problema, unido a la falta de pensamiento crítico, conlleva a que acepte como verdadero todo lo que se ve, sin cuestionar siquiera la legitimidad de dicha información. Esto permite que una persona sea fácilmente manipulable a voluntad de aquellos que buscan obtener un beneficio con dicho engaño. Por ello, la principal motivación de este trabajo es tratar de minimizar su impacto, creando una herramienta que realiza un estudio de imágenes digitales que permita obtener un rápido resultado y verifique la autenticidad de esta información.

El resto del trabajo se divide como sigue: En la Sección II se presenta los trabajos relacionados con las técnicas de detección de Manipulaciones. Seguidamente, se propone una técnica de estimación del algoritmo de interpolación cromática para detectar la zona modificada de una imagen en la Sección III. La sección IV describe los experimentos realizados para evaluar la eficiencia de la técnica propuesta. Finalmente, la Sección V muestra las principales conclusiones y el trabajo futuro.

II. Técnicas Detección de Manipulaciones

La detección de manipulaciones busca verificar la autenticidad de las imágenes. Existen distintos métodos para la autenticación, aunque están divididos en dos tipos, los activos y los pasivos [2]. Esta división se basa en si la imagen se encuentra disponible o no. Esta clasificación se muestra en la Figura 1.

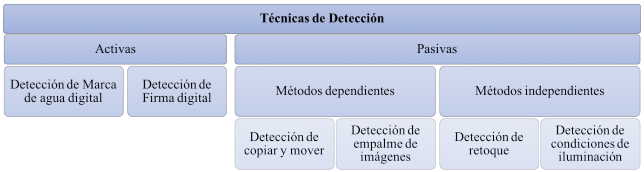


Figura 1. Clasificación de las técnicas de detección de manipulaciones

II-A. Técnicas Activas

En las técnicas de autenticación activas, la información previa de la imagen es indispensable para comprobar su autenticidad. Esta información por lo general esta oculta y son códigos que se incrustan en la imagen al momento de creación de la imagen. Para comprobar la autenticidad de la imagen es necesario verificar que esos códigos son los originales de la imagen. Ya que con un post procesamiento se pueden extraer los códigos y ser cambiados por otros [3]. Las principales técnicas activas son:

- Marca de Agua Digital: Las marcas de agua digital son insertadas en la imagen al momento de su adquisición o en la etapa de procesamiento donde se incorpora cierta información secundaria. El proceso de detección es el siguiente:
  1. Se realiza un análisis inverso sobre la estructura para localizar regiones alteradas de la imagen.
  2. Se calcula el código de autenticación de mensaje.
  3. Se calcula el Hash.
  4. Se realiza una suma de comprobación de la imagen.
  5. Se realiza un blindaje de la imagen.
- Firma Digital: Incrustan cierta información secundaria, usualmente extraída de la imagen, en el extremo de la adquisición en la imagen.

Estos métodos tienen grandes limitaciones, entre las cuales se encuentran [4]: (1) La marca de agua tiene que ser incrustada por el dispositivo de captura (cámara) o por la persona autorizada que procesa la imagen, lo cual es una aproximación poco práctica debido a la indisponibilidad de realizar las marcas de agua en la mayoría de dispositivos de captura de imágenes. (2) La calidad de la imagen, puede ser degradada durante el proceso de la marca de agua.

II-B. Técnicas Pasivas

La autenticación pasiva es el proceso de autenticación de imágenes sin requerir información previa. Estas técnicas se basan en la premisa de que a pesar que la manipulación no deje rastro visual, es probable que se alteren las estadísticas subyacentes. Son estas inconsistencias las que se usan para detectar la manipulación [5].

Estas técnicas realizan un análisis de la información binaria de la imagen digital sin ninguna información externa. Los algoritmos y métodos varían dependiendo del tipo de construcción de seguridad utilizada. Sin embargo, la detección de manipulación apunta a la localización de la manipulación indebida en la imagen.

Existen técnicas de manipulación eficientes para ocultar información en una imagen a nivel de contenido de la imagen. En este caso, el objetivo principal de las técnicas de detección pasiva es clasificar una imagen dada como original o alterada. La Figura 2 presenta la estructura general de este proceso.

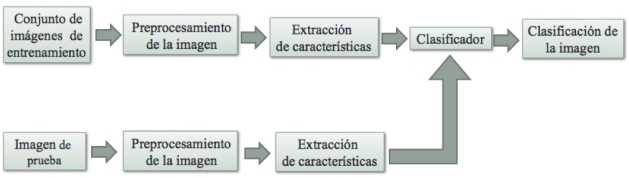


Figura 2. Técnicas de Detecciones de Falsificaciones de Imágenes

Las técnicas pasivas a su vez se subdividen en: métodos dependientes e independientes.

Los métodos dependientes se clasifican en:

- Detección Copiar y Mover: Es la técnica más popular y común de manipulación de fotos debido a la facilidad con la que se puede llevar a cabo. En este tipo de falsificación, una parte de la imagen se copia y se pega en otra parte de la misma imagen normalmente con la intención de ocultar un objeto o una región.
- Detección Empalme: En la técnica de empalme de imagen hay composición de dos o más imágenes, que se combinan para crear una imagen falsa. El empalme de la imagen supone cortar una región de una imagen y pegar en la otra.

Los métodos independientes, por su parte, se subdividen en:

- Detección de Retoque: Se utiliza con mayor frecuencia para aplicaciones comerciales y estéticas. Se usa principalmente para mejorar o reducir las características de la imagen. La detección de retoque de la imagen se lleva a cabo tratando de encontrar la difuminación, las mejoras, los cambios de color y los cambios de iluminación en la imagen falsificada. La detección es fácil si la imagen original está disponible, de lo contrario, la detección ciega es una tarea difícil.
- Condiciones de Iluminación: Las imágenes que se combinan durante la manipulación se toman en diferentes condiciones de iluminación. Es difícil combinar la condición de iluminación con la combinación de fotografías. Esta incoherencia de iluminación en la imagen compuesta puede utilizarse para la detección de manipulación de imágenes [6].

En [7] se propone un método para detectar si una imagen dada contiene regiones duplicadas apoyándose en el uso de la transformada de Gabor. El método sigue los siguientes pasos: (1) la imagen se convierte a escala de grises y se divide en bloques superpuestos de un tamaño fijo y (2) se extraen características locales de cada bloque utilizando los descriptores HOGM (del inglés *Histogram of Orientated Gabor Magnitude*) que representen el bloque entero. Finalmente, cada vector de características es lexicográficamente ordenado, y las regiones falsificadas de la imagen son detectadas a través de la identificación de pares de bloques similares. Esta técnica no es eficaz cuando la falsificación incluye post-procesamiento como: pequeñas rotaciones de imagen, escalado, compresión JPEG, difuminado, y ajuste de brillo.

Debido a que la detección de imágenes compuestas no tiene ninguna región de referencia para comprobar regiones duplicadas, en [8] se utiliza la incongruencia en las características de la varianza del ruido restante para detectar regiones modificadas y definir claramente sus contornos. Esta varianza es una clase de patrón de ruido del sensor SPN (del inglés *Sensor Pattern Noise*) resultado de las imperfecciones en la adquisición de una imagen digital y es relativamente estable. Con este método se obtienen buenos contornos de regiones modificadas. Más detalladamente, los componentes de la imagen se extraen para realizar una detección adaptativa. Luego, se calcula la varianza del ruido que queda después de reducir el ruido en cada componente de la imagen. Finalmente, las regiones modificadas son detectadas utilizando la varianza del ruido restante de los componentes. Los resultados de los experimentos realizados muestran que el método propuesto tiene una buena tasa de detección de imágenes compuestas.

Cuando una imagen es falsificada por la combinación de varias imágenes es necesario realizar modificaciones como por ejemplo, cambiar el tamaño de las regiones de las imágenes que serán combinadas, con el objetivo de que la falsificación sea convincente. Sin embargo, aunque no es posible detectar las modificaciones visualmente, a través de las correlaciones introducidas en la imagen al modificarlas se puede detectar la falsificación.

En [9] se presenta una técnica de detección de falsificaciones que utiliza el algoritmo de Expectación / Maximización (EM). La propuesta se basa en las correlaciones introducidas por el re-muestreo, donde se supone que cada muestra pertenece a una de las siguientes opciones: a) Muestras que estén correlacionadas con sus vecinos; y b) Muestras que no tienen correlación con sus vecinos.

En [10] se propone un método de detección de falsificaciones en una imagen digital usando DyWT (del inglés *Undecimated Dyadic Wavelet Transform*), una variación de la transformada wavelet. El sistema completo se conoce como DyWT, similar a DWT (del inglés *Discrete Wavelet Transform*), pero sin presentar algunas carencias como: DWT no es invariante respecto a traslaciones causando una excesiva cantidad de coeficientes que dificultan la estimación de ruido. El uso de la transformada wavelet se prefiere sobre la transformada de Fourier en cuanto a procesamiento de imágenes, pues no sólo extrae información de escalado, sino también información de localización. La transformada wavelet descompone una imagen en su representación media y en distintas representaciones de detalles direccionales.

Para el reconocimiento de patrones en la imagen, es preciso que la técnica sea inmune a rotaciones, pues a veces se copian elementos rotados. Si hay elementos parecidos en una imagen, tratar sólo con la subbanda LL1 los identifica como objetos copiados (falsos positivos). La subbanda HH1 los distinguiría por el nivel de ruido de cada uno. Por lo tanto, ambos elementos deben usarse a la vez. También es necesario convertir las imágenes a escalas de grises antes de usar el método propuesto. La aplicación del método es bastante complejo y tiene la dificultad de que al formar bloques en la imagen no se sabe si hay rotaciones, por ejemplo. Aun así, es un sistema que produce unos resultados cercanos al 100 %.

III. Descripción de la Técnica Propuesta

En esta sección se describirá la técnica propuesta para la detección de falsificaciones en imágenes a color. En la técnica se procesa la imagen completa sin ningún tipo de entrenamiento previo.

El primer paso de la técnica es estimar el patrón de interpolación de la matriz de filtro de color de la cámara digital que capturó la imagen. Para dicho proceso la imagen se re-interpola con varios patrones CFA. Para cada patrón se obtiene su Error Cuadrático Medio (MSE, del ingles *Mean Square Error*) entre la imagen original y la imagen re-interpolada.

A continuación, se analizan los resultados obtenidos del MSE para determinar si la imagen ha sido modificada. Se espera que uno de los valores del MSE calculado para cada patrón CFA sea mucho más pequeño que los otros tres. Si ninguno de los cuatro valores es significativamente más pequeño que los demás, se puede deducir que la imagen puede haber sido sometida a un pos-procesamiento. Sin embargo, en este punto no se puede asegurar que tipo de modificación ha sido realizada o si ha sido retocada.

Siendo $L_c(x, y)$ la intensidad de la imagen del canal de color c en una localización espacial (x, y) y $c \in \{R, G, B\}$, el siguiente paso es definir la máscara del filtro de color que se realiza como se muestra en la Ec. 1.

$$\theta_{k,c}(x, y) = \begin{cases} 1, \ (x, y) \in \psi_{k,c} \\ 0, \text{ otro caso} \end{cases} \quad (1)$$

donde, $\psi_{k,c}$ representa la localización del conjunto de la matriz de filtros de color del canal c para un particular tipo de patrón CFA denotado por k y $\theta_{k,c}(x, y)$ la máscara del filtro de color correspondiente de $\psi_{k,c}$

La técnica utiliza bloques de tamaño $W \times W$, donde $W = 8$ píxeles, para dividir la imagen teniendo en cuenta solo bloques no lisos. Cada bloque no liso es denotado como B_i donde $i = 1, ..., N.$, siendo N el número de bloques no lisos que contiene la imagen. Los bloques reinterpolados con el filtro k se denotan como $\hat{B}_{i,k}$. Estos bloques son calculados mediante una convolución entre el kernel bilinear y el bloque re-muestrado B_i con el kth patrón CFA definido con la Ec. 2.

$$\hat{B}_{i,k} = f(B_i, \theta_k) \qquad k = 1, ..., 4 \tag{2}$$

Seguidamente, se calcula el error de MSE entre los bloques de B y $\hat{B}$ en regiones no lisas sobre toda la imagen mediante la Ec. 3.

$$E_i(k, c) = \frac{1}{W \times W} \sum_{x=1}^{W} \sum_{y=1}^{W} (B_i(x, y, c) - \hat{B}_{i,k}(x, y, c))^2 \tag{3}$$

donde, E_i es una matriz que contiene los errores cuadráticos medio por cada canal de color.

Para detectar las distancias relativas entre los canales de color se crea una nueva matriz de error E_i^2. La normalización de todas las filas de la matriz E_i se realiza con la Ec. 4.

$$E_i^{(2)}(k, c) = 100 \times \frac{E_i(k, c)}{\sum_{l=1}^{3} E_i(k, l)}, c = 1, \cdots, 3 \tag{4}$$

Debido a que existe un menor número de píxeles interpolados en el canal verde, se toman los valores de la columna del canal verde $V_i(k)$ para determinar si existe algún tipo de modificación. Este proceso se realiza con la Ec. 5.

$$V_i(k) = 100 \times \frac{E_i^{(2)}(k, 2)}{\sum_{l=1}^{4} E_i^{(2)}(l, 2)} \tag{5}$$

Mediante la uniformidad del vector V_i se puede indicar una posible operación de pos-procesamiento. La uniformidad del vector del canal verde es definido con la Ec. 6.

$$U(i) = \sum_{l=1}^{4} |V_i(l) - 25| \tag{6}$$

Finalmente, se calcula la mediana del vector U como una métrica de seguimiento del filtro CFA como se muestra en la Ec. 7.

$$F = median(U) \tag{7}$$

Cuanto más alta es la métrica del filtro CFA (F), es más probable que la imagen pueda interpolarse con el filtro CFA. Por tanto, se puede deducir que no se sometió ningún tipo de procesamiento o alteración significativa.

Otra manera de medir los artefactos del algoritmo de interpolación cromática CFA es observando los cambios de la potencia del ruido del sensor en la imagen dada. Si una imagen es interpolada se espera que el ruido del sensor en los píxeles interpolados se suprima. Esto se debe a la naturaleza del paso bajo de la interpolación. La varianza del ruido del sensor en píxeles interpolados se vuelve significativamente más baja que la potencia de ruido del sensor en píxeles no interpolados. Por tanto, los artefactos del algoritmo de interpolación se pueden medir comparando la relación de varianzas de ruido de píxeles interpolados y no interpolados. Si esta relación es cercana a 1, se puede suponer que la imagen de entrada fue manipulada.

Una manera típica de obtener el ruido del sensor es mediante el algoritmo de eliminación de ruido basado en wavelet presentado en [11], [12]. Este proceso se realiza sobre el canal verde de una imagen separando los píxeles interpolados de los no interpolados mediante máscara del filtro del canal verde $\theta_{k,c}$, donde $k = 1$ y $c = 2$.

Los píxeles no interpolados se dividen en 2 vectores A_1 y A_2 para obtener la relación de las variaciones del ruido del sensor con la Ec. 8.

$$F_2 = max(\frac{var(A_1)}{var(A_2)}, \frac{var(A_2)}{var(A_1)}) \tag{8}$$

donde var representa la varianza del vector y max devuelve el valor mas alto entre x y y.

IV. Experimentos y Resultados

Para evaluar la eficiencia del método descrito, se utilizaron imágenes de los datasets [13] y [14] denominados D1 y D2, respectivamente.

Las imágenes del dataset D1 tienen las siguientes características: Imágenes de alta resolución (3000x2000 o 2000x3000 píxeles mínimo). con falsificaciones realistas de copiar y mover (realista"se refiere a la cantidad de píxeles copiados, el tratamiento de los píxeles del borde de la región copiada y del contenido de la región). La resolución mínima promedio de una imagen es de aproximadamente .

Las imágenes del dataset D2 tienen las siguientes características: La resolución de las imágenes son de tamaño mediano (1000x700 o 700x1000), con imágenes no comprimidas con regiones copiadas y movidas simplemente, imágenes no comprimidas con escenas simples (un objeto, fondo simple) en lugar de escenas complejas, ya que el dataset se utiliza para estudiar principalmente la robustez contra algunos ataques específicos.

Las características del equipo con el que se han realizado los experimentos se presentan en la Tabla I. Es un factor importante a tener en cuenta ya que los tiempos de ejecución de las diferentes pruebas varían según los recursos computaciones disponibles.

Tabla I
Características del equipo de experimentación

Recursos	Características
Sistema operativo	Ubuntu 18.04
Memoria	4 GB
Procesador	Intel® Core™ 2 Quad CPU Q8200 @ 2.33GHz x 4
Gráficos	NV96
Tipo de SO	64 bits
Disco	100 GB

Para evaluar la eficiencia del método descrito, se utilizaron imágenes de alta resolución (superiores a 1500×1500 píxeles) con y sin alteraciones en diferentes áreas de la imagen [13] [14]. Además se midió el tiempo que le toma al método mostrar el área donde se tiene modificación. En la Figura 3 se puede observar que el resultado obtenido.

En la Figura 3.a se muestra la imagen original, la Figura 3.b se observa modificación realizada y en la Figura 3.c se representa el resultado obtenido al aplicar la técnica propuesta. En ella se hace evidente la región donde se aplicó la alteración resaltando el área modificada. Cabe destacar que las dimensiones de la imagen son de 2000×3008.

Sin embargo, existen casos donde los resultados no son tan claros debido a las condiciones de la imagen por ejemplo cuando existen fondos muy claros como cielos ocasionando que sean marcadas zonas donde no existe una modificación.

La Figura 4 da muestra de ello, si bien hace la delimitación del área modificada correctamente, son mostradas zonas en la parte del cielo (c) donde los cálculos indican que existe una falsificación.

Al analizar los resultados con imágenes pequeñas del dataset D2 (ver Figura 5) se pudo observar que el método no es preciso debido a la baja resolución de la imagen y que al procesar la imagen y formar los bloques de tamaño WxW, descritos en secciones anteriores, la falta de información de la imagen hace que todas las varianzas sean bajas no habiendo una diferencia significativa entre ellas. En la Tabla II se muestra el tiempo que le tomo al método analizar imágenes de diferentes resoluciones. El tiempo empleado para el procesamiento de imágenes de resoluciones altas fue de $24{,}2959$ segundos lo que demuestra que el método es eficiente y muy preciso con imágenes grandes.

Tabla II
TIEMPO DE EJECUCIÓN DEL MÉTODO

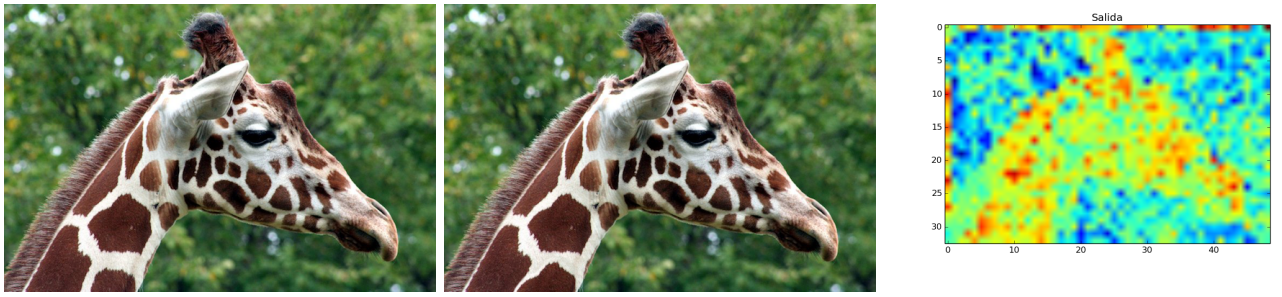
Resolución	**Tiempo (s)**
2000x3008	20,3427
2014x3038	25,9153
2304x3072	22.6366
2448x3264	28.2889

(a) Imagen Original (b) Imagen Modificada (c) Falsificación Detectada

Figura 3. Resultados óptimo

(a) Imagen Original (b) Imagen Modificada (c) Falsificación Detectada con Falsos Positivos

Figura 4. Resultados con Errores



(a) Imagen Original (b) Imagen Modificada (c) Falsificación Detectada con Falsos Positivos

Figura 5. Resultados Obtenidos con Imagenes de Baja Resolución

El método propuesto fue desarrollado en el lenguaje de programación python ya que cuenta con librerías que facilitan el procesamiento de la imagen. El tiempo de procesamiento de cada imagen es bajo teniendo en cuenta que se utiliza toda la información de la imagen sin ningún procesamiento previo y que las imágenes de prueba son imágenes de alta resolución y a color.

V. Conclusiones

Este trabajo presenta una técnica para detectar manipulaciones en una imagen a color mediante algoritmos de interpolación cromática. En el desarrollo del trabajo se pudo observar que mediante una estimación del patrón de interpolación y el error cuadrático medio de bloques de la imagen se puede determinar si existe o no una modificación en una imagen dada.

El método descrito obtuvo resultados satisfactorios cuando se ingresaban imágenes con dimensiones superiores a $1500x1500$ píxeles delimitando la zona modificada. Un punto a considerar son imágenes que tienen cielos con colores blancos ya que marca como modificación debido a que la varianza calculada en dichas secciones es muy baja al resto de la imagen. Así mismo, los mejores resultados son obtenidos con imágenes grandes ya que la información de la imagen es suficiente para calcular de manera correcta la varianza de la imagen y se puede hacer una distinción de éstas. Sin embargo, con imágenes menores a 700×700 píxeles el método tiene dificultades para detectar la zona con modificaciones ya que la información de la imagen no es suficiente para hacer notoria la diferencia entre las varianzas.

Una ventaja que tiene la técnica sobre otras propuestas es el tiempo de procesamiento de una imagen, ya que éste no supera un minuto. Esta rapidez la hace eficaz para utilizarse en en secuencias de vídeo.

Agradecimientos

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326. Website: http://ramses2020.eu

Referencias

[1] J. Bañuelos Capistrán and F. Mata, "Fotografía y Dispositivos Móviles," ITESM/Porrúa Print, Technical Report, November 2014.

[2] G. K. Birajdar and V. H. Mankar, "Digital image forgery detection using passive techniques: A survey," *Digital Investigation*, vol. 10, no. 3, pp. 226–245, 2013.

[3] Z. Zhang, Y. Ren, X.-J. Ping, Z.-Y. He, and S.-Z. Zhang, "A survey on passive-blind image forgery by doctor method detection," in *Machine Learning and Cybernetics, 2008 International Conference on*, vol. 6. IEEE, 2008, pp. 3463–3467.

[4] B. Malcolm, *El Libro Completo de la Fotografía*. Boca Raton, FL, USA: Ediciones AKAL, 1999.

[5] Z.-p. Zhou and X.-x. Zhang, "Image splicing detection based on image quality and analysis of variance," in *Education Technology and Computer (ICETC), 2010 2nd International Conference on*, vol. 4. IEEE, 2010, pp. V4–242.

[6] S. Mushtaq and A. H. Mir, "Digital image forgeries and passive image authentication techniques: A survey," *International Journal of Advanced Science and Technology*, vol. 73, pp. 15–32, 2014.

[7] J.-C. Lee, "Copy-Move Image Forgery Detection Based on Gabor Magnitude," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 320–334, August 2015.

[8] W.-C. Hu, J.-S. Dai, and J.-S. Jian, "Effective Composite Image Detection Method Based on Feature Inconsistency of Image Components," *Digital Signal Processing*, vol. 39, pp. 50–62, April 2015.

[9] A. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Resampling," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 758–767, February 2005.

[10] P. Muhammad, M. Hussain, and G. Bebis, "Passive Copy Move Image Forgery Detection using Undecimated Dyadic Wavelet Transform," *Digital Investigation*, vol. 9, no. 1, pp. 49–57, June 2012.

[11] A. E. Dirik and N. Memon, "Image tamper detection based on demosaicing artifacts," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1497–1500.

[12] A. L. Sandoval Orozco, L. J. García Villalba, D. M. Arenas González, J. Rosales Corripio, J. Hernandez-Castro, and G. S. J., "Smartphone Image Acquisition Forensics using Sensor Fingerprint," *IET Computer Vision*, vol. 9, no. 5, pp. 723–731, September 2015.

[13] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An Evaluation of Popular Copy–Move Forgery Detection Approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[14] E. Ardizzone, A. Bruno, and G. Mazzola, "Copy–move forgery detection by matching triangles of keypoints," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2084–2094, 2015.

Forensic Analysis Overview in the IoT Environment. A Windows 10 IoT Core Approach

Juan Manuel Castelo Gómez
Albacete Research Institute of Informatics
Albacete, Spain
juanmanuel.castelo@uclm.es

José Luis Martínez Martínez
Albacete Research Institute of Informatics
Albacete, Spain
joseluis.martinez@uclm.es

***Abstract*—The huge development of the Internet of Things (IoT) and the sudden incursion of this network into our everyday world have drastically changed the application of technology in our lives. One of the main concerns arising from the IoT is security; the way the devices have been conceived has turned out to include a massive underestimation of the security requirements, which has led to a large-scale problem. In this article, a review of the state of security on the IoT is carried out, focusing on the forensics aspect. In addition, a case study is presented on how to perform a forensic analysis in an IoT-based operating system, namely Windows 10 IoT Core.**

***Index Terms*—Cybersecurity, Forensics, IoT, Windows 10 IoT Core.**

Type of contribution: *Original research*

I. Introduction

The word *thing* covers the entirety of items in the world and, although it might not seem so, that is quite accurate when used to describe this new paradigm. Almost anything is possible. IoT devices can be found everywhere, even in places where nobody expects. Nowadays, it is very common to talk about smart homes, eHealth or smart industries, among other topics. We are using IoT devices without even noticing it. Everyday technology users will find themselves using smart TVs, smart washers, smart watches or simply sensors in their home that measure temperature or the amount of light in a room, or detect presence. In 2018 there was an estimation of more than 11 thousand million IoT devices installed, with 63% of them corresponding to the consumer segment. In 2020 it is predicted that this value will double and reach 20.4 thousand million units [1]. The fact that this huge number of devices coexist in the IoT environment is a great advantage for consumers, as they can choose from a wide range of options, but it is a big inconvenience for developers, since facing such a heterogeneous platform makes it more difficult to establish common ground to be shared by all the systems.

Regarding security, the main concern when operating with IoT devices is that they are not secure enough, especially the ones that were designed when the IoT was starting. This was due to the lack of attention paid to security at that time, with developers focused on usability and offering a wide range of products instead of implementing appropriate security measures. Furthermore, almost nobody who was designing devices expected that something as simple as a smart switch could be compromised and that this could lead to an incident that could affect the whole network in a home. Nowadays, companies have acknowledged this issue and are starting to add security features to ensure that the devices and the information that is handled by them is protected, but there is still a long way to go. As stated above, we are now surrounded by IoT devices, and most of them handle very sensitive information, such as that related to our health or our home, so it is crucial to ensure that the data that they store is only accessed by the right people. Moreover, it is an issue that is currently affecting us, not something that will only happen in the future. It is very common to see an industry with multiple sensors that capture information about the state of a machine in order to control it. Any of those sensors is susceptible to security threats, especially if their measures are almost non-existent, and the consequences of them behaving erratically can have a huge impact on a company. The need to implement proper security on these devices is imperative.

A glance at the number of malware samples detected in recent years shows the magnitude of the problem. In 2018 it was more than 120 thousand, almost four times more than the figure for the year 2017 [2]. 20.9% of those samples belonged to the Mirai botnet family, a piece of malware that in its very first version affected more than 600 thousand IoT devices, and, with its different variations, went on to infect millions. Almost three years later, new versions of Mirai still appear every day. This malware took control of a system by using a login and password dictionary and, in most cases, the devices were still using the default values [3], confirming what was stated above, namely that security measures in IoT devices are not strong enough to protect users from the simplest attacks. If we focus on the purpose of the malware samples that were detected in 2018, the three most common types used were DDoS (Distributed Denial of Service) attacks, cryptocurrency mining and data theft.

In this article a study of the state of IoT security is carried out, highlighting the main requirements and challenges encountered by the community when working in this new environment. The forensic point of view is addressed too, explaining how different an investigation when IoT devices are present is, compared with a traditional one, and how it should be approached. Also, the IoT-based operating system developed by Microsoft, namely Windows 10 IoT Core, is analyzed and used as a case study to extract what data stored on it is useful from a forensic perspective.

The rest of the paper is organized as follows. Section II provides an analysis of Windows 10 IoT, Section III discusses the related work in IoT security and forensics and Section IV an overview of how to perform a forensic analysis on Windows 10 IoT Core is presented and the evidence found in it is listed in Section V. Finally, our conclusions are presented in Section VI.

II. Background

Windows 10 IoT is the Internet-of-Things-based operating system developed by Microsoft, which was launched in 2015. It is a free version of the desktop Windows 10 version, optimized for ARM and x64/86 devices such as Raspberry Pi, Dragon Board or Minnow Board. There are two editions: Windows 10 IoT Core, the free small version that runs a single app to interact with the system; and the Enterprise version, a full version of Windows 10 only supported by x86/64 devices and focused on providing features to create services and devices [4]. The applications are developed with the Universal Windows Platform (UWP), supporting the languages C++, C#, JavaScript and Visual Basic. To set up a device and connect to it, the Windows 10 IoT Core Dashboard application must be used. This tool allows you to execute PowerShell commands and change multiple settings, as can be seen in Figure 1.

Some main features of this system are:

- Secure Boot: UEFI located security feature to only allow the execution of trusted applications signed by known authorities.
- Bitlocker Encryption.
- Device Guard: allows the execution of only trusted code, identifying the firmware, drivers and applications that should run on the device [5].
- Cortana (no longer available since version 1809).
- PowerShell.
- Windows Update.
- Bluetooth.
- Web, SSH and FTP Server.
- Compatibility with Arduino boards.
- Miracast.
- WiFi Direct.
- Other hardware compatibility such as WiFi Adapters, Ethernet Adapters, Cameras, NFC, RFID and multiple sensors.

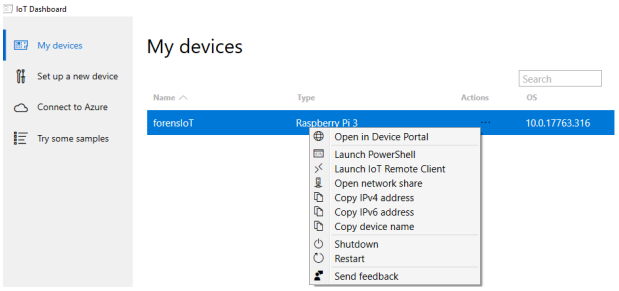


Figure 1. Windows 10 IoT Core Dashboard.

III. Related Work

A. IoT Security

The first security concerns arising from the IoT environment can be found in [6]. It presents several differences that the IoT architecture has compared with the traditional ones, such as the formation of larger networks and the lack of a unified structure. The main resulting security problems are that data transmission is via wireless networks, meaning that signals are publicly exposed; the environment is very heterogeneous; and there are no universal standards for the development of IoT applications. This analysis is supported by [7], which also states that the approach for developing security mechanisms in the IoT has to be different from the one used in classical systems, due to the new features and characteristics of this new paradigm. In addition, a model based on nodes is proposed to represent the interaction between the main actors in the system and security practices. An interesting statement is made in [8] regarding the computational power of IoT devices, which causes, among other things, the need for a reduction in the computational requirements for cryptosystems or security applications, such as antivirus, in order to be able to use them. Supporting that idea, [9] adds that hardware-based security is the best approach for the IoT, taking into account that computational power limitation, and reviews the existing physical unclonable functions and their potential to be used as a security protocol.

A different perspective is offered by [10], in which IoT security concerns are classified regarding the different layers that form the general IoT architecture, and the authors also present a detailed analysis of how each layer should be protected. In addition, the most common threats for each layer are described, specifying what kind of attack they could individually suffer. The same standpoint is held by [11], but, instead of focusing on the security needs of each layer, it offers a more general perspective, evaluating security measures that affect all layers and reviewing the main measures taken by the community that affect authentication, trust establishment and security awareness.

In [12] a secure execution environment is proposed in which a processing unit can execute applications in a protected manner, securing the device physically and not depending on a software solution that controls the processes in the system, adapting safety solutions to the characteristics of IoT devices and making the design of security systems a key task in the development process. Another interesting proposal is [13], which is focused on improving secure communications in IoT networks. A new routing protocol is introduced to authenticate devices when forming a network or joining an existing one, carrying out several tests to demonstrate that the security of the network has not been compromised by a malicious device, and that the overhead added by the protocol is almost insignificant. [14] tackles the problem of having unpatched and unupdated firmwares on devices by developing a system that identifies the devices present in a network and makes a vulnerability assessment of each one of them. The communications established by every device are monitored and analyzed to decide whether they are a potential vulnerability or a harmless connection. For this purpose, a security gateway is used to monitor and control traffic and then, using a machine learning classification model, an evaluation is made to determine the isolation level required by a device, depending on the known CVEs (Common Vulnerabilities and Exploitations) for it.

Focusing on the area in which IoT devices are used, we can also see how every context requires different security measures. In [15] a comprehensive analysis is performed, thoroughly studying two different IoT devices, namely a smart home sensor and an industrial smart meter. The security measures implemented on them were proven to be insufficient

by carrying out several attacks that could cause a huge impact in a real scenario. Regarding the smart home sensor, they gained access to the root account of the device, its password and the boot parameters, as well as being able to obtain the binary update file. Something similar occurred with the industrial meter, for which a modification of ID of the device was successfully performed, which lead to the possibility of making the device identify itself as if it were another. In relation to smart home security, [16] presents the requirements that devices should meet to provide a trustworthy service, describing different components that can be found in the typical smart home infrastructure and highlighting, for each one of them, the security functions that they are supposed to provide.

Another interesting context in which IoT devices can be used is in the eHealth domain. In [17] the development of a remote monitoring device for elderly people is presented, explaining how security measures have been implemented to ensure that the data handled by the application is protected. Although its perspective is software development centered, it is very useful to understand how security is one of the key features when developing an IoT solution, especially when dealing with such sensitive data as that of the health of people.

B. IoT Forensics

A good starting point for understanding the current state of forensics research, is [18], in which the problems that arise when using IoT devices are described and other useful topics are mentioned. A key issue is highlighted, and that is the relationship between IoT devices and the cloud, which is an important feature when working in this environment. Another interesting article with a similar perspective is [19], which presents the different parameters of IoT Forensics, such as the sources of evidence, the number of devices, the quantity and type of data, comparing this with traditional scenarios. In addition, two approaches are proposed on how to perform an IoT forensic analysis, stating the most relevant points to focus on. To appreciate the wide range of IoT applications and what scenarios an investigator could face [20] is very useful; it also presents the taxonomy of IoT forensics as well as its requirements, offering a very complete analysis of the situation.

With these challenges in mind, solutions are proposed to facilitate analysis when dealing with IoT devices. One such solution can be found in [21], where a system is proposed to autonomously perform forensic tasks in an IoT environment, helping investigators to save time and automatize the analysis, allowing them to focus on obtaining information rather than spending time on trivial tasks such as parsing data, managing storage or creating time lines. In the quest for processing data more efficiently, the cloud emerges as an interesting possibility, as is stated in [22], which proposes a cloud-based service to perform forensic operations, allowing investigators to collaborate in an easier way and perform tasks more quickly, automatizing non-forensic actions such as resource management. Something similar is suggested in [23], where a model for performing IoT forensic investigations is designed, and guidelines are given to investigators on how to approach the analysis. Opting for a centralized model seems to be the most frequently selected approach, as is the case in [24], which presents a forensic-aware IoT idea that securely preserves evidence and assures the chain of custody.

The immense diversity that characterizes the IoT environment leads to researchers focusing on studying specific devices. In [25] an investigation is carried out in order to determine what information stored in a smart TV can be important when performing a forensic analysis on it. Other relevant devices are smart watches, which contain a considerable amount of sensitive information, as is shown in [26], in which two models are examined and a forensic analysis is performed on them, explaining the acquisition process and the tools that are used. The information obtained from them is not very relevant for an investigation, but the process followed is very interesting and significant in helping explain how to manage this kind of devices. Due to the wide implementation of the IoT, we also find research regarding Smart Cities; in [27] recommendations are made on how to acquire and analyze the information that can be found in the electronic control unit of a car. Another vehicle-related study is [28], in which a useful term related to the IoT is introduced, the IoV (Internet of Vehicles). In this research, a framework is proposed for the recovery and storage of evidence that has been created in an environment which involves vehicles, networks, IoT devices and cloud computing.

Another important environment created using IoT devices is the smart home, which, for ordinary technology users, is probably the most interesting and common implementation of this kind of technology. A framework is proposed in [29] to explain the phases that an analyst needs to follow in an investigation, and the authors also address the challenges associated with smart homes. Three interesting case studies are carried out, in which different scenarios that could arise when analyzing a home environment can be seen, demonstrating that the framework is useful and how different an investigation can be when working in these types of situations.

One of the major changes in digital forensics when dealing with IoT investigations is that the importance of the environment surrounding the device is far greater than in traditional analysis. The lack of computational process on IoT devices is balanced by the ability to exchange information with other similar systems, which greatly extends the range of forensic analysis. For this reason it is very useful to study an environment as a whole and not to focus only on examining devices individually. An interesting study is [30], in which an analysis of the Amazon Alexa ecosystem is made, examining the interaction of all the interconnected devices in that environment, such as mobile phones, computers and smart speakers, and what data can be extracted from them and be used in a forensic analysis.

Not only is it relevant to determine how to analyze IoT devices, it is also important to understand how a system behaves. In [31] a perfect example is presented, with the authors proposing a method for monitoring the energy consumption pattern of the processes that are running on three different Android devices, and using a classification algorithm to be able to detect crypto-ransomware malware in IoT networks. Another relevant operation is to determine what threat is affecting the device depending on the forensic information obtained from it, so investigators have an idea about how to

approach the analysis. An interesting proposal is [32], which lists the possible attacks that an IoT network can face, and develops a model that recognizes them depending on the actions performed by the devices of a smart home.

The integration of the IoT with the cloud, as well as the limited amount of storage of the devices means that most of the data stored on them will be transferred to other device or saved by a cloud storage provider. In [33] a framework is proposed to identify when evidence belongs to a local artifact or has been synced from another device. The evaluation is carried out in a Windows scenario, but it can be very useful to extrapolate that to the IoT environment, where connectivity with other devices and the exchange of information between them is essential. In addition to the information that can be found on a device, it is also important to understand how an attacker would act when trying to breach the security of a smart home installation. With this in mind, [34] introduced a model of what actions can be performed by an attacker and studied them on two IoT devices, a switch and a bulb, successfully completing several attacks, exposing the low security level of these devices.

IV. Windows 10 IoT Forensic Overview

As it can be extracted from the Related Work, the high heterogeneity of the IoT environment is one of the main challenges that investigators have to face so, to address this issue, researchers study specific devices or operating systems to help understand what evidence is relevant in certain contexts and how to extract them. In this case, we focus on the Windows 10 IoT Core operating system, with the goal of understanding how it works and offering helpful information for investigators about what data stored in it is useful when performing a forensic analysis on this operating system.

Before starting with the forensic analysis, it is very important to understand how the system that we are going to analyze behaves. In this case, we are dealing with an operating system that derives form Windows 10, one of the most used OS in the world and, as a result, a well-known system forensically speaking. However, Windows 10 IoT Core incorporates new features and, what is more important, is designed for completely different purposes than the desktop version. For these reasons, a system overview is mandatory before addressing the forensic analysis. As we are focusing on investigating the data stored in non-volatile memory, this task consists in determining how the storage is structured.

A. Test Environment

Before carrying out the analysis, we need to prepare and configure the environment in order to make sure that the experiment is performed properly. In our case, the components used are the following:

- Raspberry Pi Model 3 B.
- 32 Gigabyte microSD Card.
- Windows IoT Core Build 17763.

In order to prepare the microSD card to perform the installation of the operating system on the device, a desktop computer with Windows 10 running is needed with the Windows 10 IoT Core Dashboard application. This device also acts as the forensic computer, on which the acquisition and the analysis are performed.

B. Procedure

We are dealing with an extraordinary forensic analysis, that is, we are not basing the investigation on an incident; our purpose is to obtain the useful information that is available on the device, regardless of the circumstances that surround the examination. This means that a different approach is needed to ensure that no information is ignored, requiring a more general way of tackling the analysis. Specifically, it is essential to understand how the operating system behaves. With this in mind, two different acquisitions of the device are carried out:

- Raw installation. Once the microSD Card has been sanitized we launch the program created by Microsoft to manage the IoT device (Windows 10 IoT Core Dashboard) and install the operating system on it. When the installation has finished, we proceed to create an image of the microSD Card and start analyzing its content. The purpose of this acquisition is to understand the system in its conception, before any usage data is injected into it.
- First boot. The next step is booting the system for the first time, accepting all the terms and setting all the privacy options by default. After the boot is completed and the main screen is shown, we turn off the device and create an image of it. In this case, we are trying to understand what information the operating system contains once it is ready for the user to work with.

As it can be inferred from the way the acquisitions have been carried out, no standard forensic methodology has been followed, although the conventional phases in which a forensic analysis is divided into can be clearly identified. The main difference is that the acquisition and analysis phases have no specific ending strictly speaking, as we are studying how the actions that are performed in the operating system, meaning the installation of the system and it first boot, are reflected in the data contained in the non-volatile memory. So, once the identification phase has ended, in which we established that the data stored in the microSD card was the representative source of non-volatile information, the acquisition and analysis phases are performed twice.

C. Forensic Tools

The tools needed to acquire and analyze the data acquired are listed below. The decision to use these tools is made on the basis of the knowledge that they are compatible with different operating system versions and, in particular, with the desktop version of Windows 10.

- FTK Imager: used for image creation of the microSD Card [35].
- Autopsy: analysis tool for exploring purposes. It allows us to browse through the storage and recover deleted files [36].
- QPhotorec: data carving tool that enables recovering files from an image file [37].
- Registry Explorer: analysis tool for obtaining information from the Windows registry [38].
- RegRipper: Windows registry extraction tool that interprets the data stored in the registry hives [39].

D. Data Acquisition and Analysis

The data acquisition is performed using FKT Imager. As the non-volatile memory is stored on the microSD card, the process is quite simple, as is shown from the description below:

- The microSD card is extracted from the Raspberry Pi board and plugged into a microSD to SD card, which allows us to write-block the storage.
- The adaptor is inserted into the forensic computer that is running the Windows 10 desktop version and FTK Imager.
- The image file of the SD card is created and stored on the forensic computer.

Once the image file is created, it can be mounted on the system using the same tool. This allows us to browse through the directories of the file system and extract the files that seem relevant and recover the deleted files using QPhotorec. Also, the image file format is supported by other analysis tool such as Autopsy, which offers an interface thath is easy to use and several features that allow us to filter the data contained in the image file.

V. Forensic Evidence Found in the System

After analyzing the image file acquired, several useful pieces of evidence are found in the system. This evidence is listed below, and for every item an explanation about why it could be useful in an investigation is given.

A. Partitions

Three different partitions can be found in the system, as is shown in Table I. It is important to understand how the operating system distributes the information among all the partitions available, especially on these devices, where storage space is limited.

Table I
Partitions found on the SD Card.

Partition	Description
EFIESP	FAT 32 Extensible Firmware Interface system partition used by the device to boot and which stores boot loaders, applications and drivers that are launched by the UEFI firmware. Its size is 63,7 Megabytes, 47,9 remaining free.
MainOS	NTFS partition acting as the system root directory. Its size is 1,39 Gigabytes and it is the one launched by Windows Boot Loader when the device is powered on.
Data	NTFS partition that is used by the system to store most of the information, as it is the largest of all three available. Its size depends on the microSD card capacity, since the partition takes all the remaining space available after the "EFIESP" and "MainOS" partitions have been created.

B. Apps

The different programs that can be installed on Windows 10 IoT Core are presented in the form of Apps, which are similar to the ones that are used on smart phones. They are the ones that provide meaning to a system, so they are crucial in a forensic analysis, firstly since they help investigators to understand what the purpose of the device is and, secondly, because they contain a lot of usage data. They are stored in the "Data" partition, specifically in the `Programs\WindowsApp` route.

C. Registry

The registry is one of the main sources of information that can be found in the operating system developed by Microsoft. It contains data regarding the system and user configuration, hardware devices and applications installed. The information is stored in the form of a hive, which contains the registry keys, sub-keys and values. The same registries that can be found in the desktop version are found in the IoT version. In Table II the system registries are listed and described, and Table III shows the user registries, which are also the same as the ones available in the desktop version.

Table II
System Registry Hives .

Registry Hive	Description
COMPONENTS	Holds data associated with Windows Update configuration and status [40].
DEFAULT	Profile for the Local System account. Used by programs and services that run as Local System such as winlogon or logonui [41]
DRIVERS	Stores the drivers installed on the machine and their dependencies.
SAM	Contains information used by the Security Accounts Manager. Among other data, it contains data regarding usernames and passwords.
SECURITY	Collects local security information used by the system and network.
SOFTWARE	Stores program variables and settings that apply to all the device users.
SYSTEM	Contains device drivers and service configurations, which are stored in control set form [42][43].

Table III
User Registry Hives.

Registry Hive	Description
NTUSER.dat	Stores personal files, preferences and settings for each user. Very useful to obtain data about shellbags, the configuration stored for each directory in Windows Explorer that provides us with information about the content that the user has visited using the file explorer [44].
Usrclass	Used to record configuration information from user processes that do not have write permission to the standard registry hives. Information regarding shellbags is also stored here [45].

As is well known, a lot of useful data can be found in the Windows registry, some examples of which are the following:

- Mounted Devices.
- Default Application Path.
- Data Directory.
- Program Files directory.
- Common Files directory.
- USB connected.
- Dlls in the system.
- Event logs information.
- System services.
- Drivers installed.
- Policy control.
- Digital certificates.

D. Users

Knowing what users coexist in the system what their purposes are, and what permissions they have, allow investigators to understand how the different changes that a system has undergone could have been made. In Windows 10 IoT Core we find up to seven different users: DefaultAccount, DevToolsUser, System, Administrator, Guest, WDAGUtilityAccount and sshd.

E. Bluetooth and WiFi Connections

This is probably the most relevant data that can be found in an IoT environment, as these devices are desgined to exchange information at all times. In order to communicate with each other, technologies such as WiFi and Bluetooth are used. The best location to look for such evidence is the Windows registry, since, as stated above, it is the best place to collect data in Microsoft systems. This also applies to Bluetooth and WiFi connections. With respect to WiFi, data such as Network Interface Cards available, interface configuration or Wireless profile settings can be extracted. In the case of Bluetooth data, the IDs and names for the devices connected are stored.

F. Browser

The browser is one of the mandatory sources of evidence to be analyzed in a forensic investigation. When studying a desktop system or a smart phone, the relevance of this data is much greater than in an IoT system. This is due to the fact that IoT devices are not intended to be used for browsing the web, but they are still provided with a browser, so it is mandatory to analyze it. After studying the registry, we have found out that the User Agent used in the native Windows 10 IoT browser is Mozilla/5.0 (compatible; MSIE 9.0; Win32). Also, evidence regarding the web pages visited, cookies and cache can be extracted from the registry and the App folder for the browser.

G. System Events

All the relevant actions that occur on a Windows device are stored in the form of events. The information that is saved is classified into four different categories, depending on what component of the system has been affected. These categories are:

- Application: incidents with the software and components installed on the system.
- Security: data regarding the Windows system audit policies.
- Setup: data regarding the control of domains.
- System: mainly events related to the Windows system files [46][47].

Depending on the impact that an action has had on a system, the event is also categorized in an error, warning or information message. Logs for each category can be found in the `Windows/system32/winevt/Logs/` route.

H. Pagefile and hiberfil

The interaction between the physical memory and the persistent storage is basic in the functioning of a system. This exchange of data leaves very useful sources of information for forensic investigators, such as the hiberfil and pagefile files. Both of them are available in Windows 10 IoT Core, and are stored in the "MainOS" partition. In order for the system to create them, the option has to be enabled in the registry, which does not happen in the case of the hiberfil file, as the hibernation option is not active. They contain the following information:

- Hiberfil: file that is created when the system is put in hibernation mode, saving the state of the device. It contains volatile data that, instead of being stored in RAM memory, is saved temporarily in non-volatile memory before shutting down the system and then recovered when the device is restarted. User passwords, deleted files, connections established or information about processes that were running in the system can be found in this file, among other data.
- Pagefile: well-known file that is used to temporarily exchange data between RAM memory and persistent storage. This virtual memory is created through paging, in order to have more space available in physical memory. A piece of information that is stored in RAM memory will also be in the pagefile file, so it is a very useful source of evidence.

VI. Conclusions and Future Work

A. Conclusions

We have addressed IoT security, describing the challenges that it involves and the requirements that this new paradigm and its users make, showing the need to improve the security measures on existing devices, as well as the obligation for developers to prioritize security over other characteristics, due to the sensitivity of the information that is currently being handled by these systems. From the forensic perspective, we have highlighted the drastic change that the IoT environment has made to how to approach an analysis and what information should be extracted from a system, outlining the challenges that investigators have to face in order to carry out a successful investigation.

Regarding the analysis of the Windows 10 IoT Core operating system, it has been shown, from a forensic point of view, what data is more useful to be acquired and analyzed when dealing with IoT devices, and when they are running this operating system. The information extracted from our device has allowed us to acknowledge that the desktop Windows 10 version and the IoT-based one share interesting evidence, which makes the analysis easier and allows investigators to understand that it is very useful to study other similar systems when performing an investigation, particularly when they are based on the same concept. In addition, it has been proven that some forensic tools that are currently used to carry out analysis in Windows 10 desktop operating systems are compatible and very useful when working with Windows 10 IoT Core.

From a methodological perspective, the usefulness of having different approaches for multiple types of devices has been confirmed, especially when working in the IoT environment, where there is a high degree of heterogeneity. Every investigation has its peculiarities and having research to rely on can make a huge difference when studying a device. Knowing how a system behaves before analyzing it allows investigators to

be aware of what to expect and be ready in case a problem arises.

B. Future work

This work has been an introduction to the IoT forensic world. The need for guidelines on how to approach a forensic analysis of IoT devices has proven to be fundamental in the security field, so there is a wide spectrum of research that needs to be carried out to ensure that investigators have the right tools and knowledge to address this new paradigm. Some of these projects could be the following:

- Broaden the analysis of the Windows 10 IoT Core operating system, focusing on other useful data such as that included in volatile memory and that regarding traffic connections.
- Perform further research on how similar the desktop version of Windows 10 and the IoT-based one are, focusing on their behaviour and the evidence that can be acquired from them.
- Automatize the process, developing tools that allow us to automatically capture data present on the device, facilitating the analysis for investigators.
- Expand the forensic analysis to other IoT-based operating systems, so the community has guidelines on how to approach an investigation on the most commonly used IoT systems.
- Understand the interaction between IoT devices and reflect that behaviour from a forensic point of view. When dealing with heterogeneous IoT infrastructure new problems arise for an investigator, so comprehending how devices interact with each other is essential to be able to perform a good analysis.

ACKNOWLEDGMENTS

This research was supported by the University of Castilla La Mancha under the contract 2018-CPUCLM-7476, by the Ministry of Science, Innovation and Universities under the project RTI2018-098156-B-C52 and by the Regional Government of Castilla-La Mancha under the project SBPLY/17/180501/000353.

REFERENCES

[1] Gartner Inc., "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016," https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016.

[2] Mikhail Kuzin and Yaroslav Shmelev and Vladimir Kuskov, "New trends in the world of IoT threats - Securelist," https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/.

[3] Denis Makrushin, "Is Mirai Really as Black as It's Being Painted? - Securelist," https://securelist.com/is-mirai-really-as-black-as-its-being-painted/76954/.

[4] Windows Dev Center, "Overview of Windows 10 IoT Core - Windows IoT- Microsoft Docs," https://docs.microsoft.com/es-es/windows/iot-core/windows-iot-core.

[5] ——, "Enabling Secure Boot, BitLocker, and Device Guard on Windows 10 IoT Core - Windows IoT - Microsoft Docs," https://docs.microsoft.com/es-es/windows/iot-core/secure-your-device/securebootandbitlocker.

[6] K. Zhao and L. Ge, "A survey on the internet of things security," in *2013 Ninth International Conference on Computational Intelligence and Security*, Dec 2013, pp. 663–667.

[7] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou, and A. Bouabdallah, "A systemic approach for iot security," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, May 2013, pp. 351–355.

[8] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen, and S. Shieh, "Iot security: Ongoing challenges and research opportunities," in *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Nov 2014, pp. 230–234.

[9] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of iot systems: Design challenges and opportunities," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2014, pp. 417–423.

[10] M. U Farooq, M. Waseem, A. Khairi, and P. Sadia Mazhar, "A critical analysis on the security concerns of internet of things (iot)," *International Journal of Computer Applications*, vol. 111, pp. 1–6, 02 2015.

[11] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (iot) security: Current status, challenges and prospective measures," in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec 2015, pp. 336–341.

[12] A. Ukil, J. Sen, and S. Koilakonda, "Embedded security for internet of things," 04 2011, pp. 1 – 6.

[13] P. L. R. Chze and K. S. Leong, "A secure multi-hop routing for iot communication," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, March 2014, pp. 428–432.

[14] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 2177–2184.

[15] J. Wurm, K. Hoang, O. Arias, A. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial iot devices," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2016, pp. 519–524.

[16] J. Han, Y. Jeon, and J. Kim, "Security considerations for secure and trustworthy smart home system in the iot environment," in *2015 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2015, pp. 1116–1118.

[17] R. Giaffreda, L. Capra, and F. Antonelli, "A pragmatic approach to solving iot interoperability and security problems in an ehealth context," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 547–552.

[18] D. Lillis, B. Becker, T. O'Sullivan, and M. Scanlon, "Current challenges and future research areas for digital forensic investigation," *CoRR*, vol. abs/1604.03850, 2016. [Online]. Available: http://arxiv.org/abs/1604.03850

[19] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, "Internet of things forensics: Challenges and approaches," 10 2013.

[20] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. A. Kazmi, and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265 – 275, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18315644

[21] E. Oriwoh and P. Sant, "The forensics edge management system: A concept and design," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, Dec 2013, pp. 544–550.

[22] R. van Baar, H. van Beek, and E. van Eijk, "Digital forensics as a service: A game changer," *Digital Investigation*, vol. 11, 05 2014.

[23] S. Perumal, N. M. Norwawi, and V. Raman, "Internet of things(iot) digital forensic investigation model: Top-down forensic approach methodology," in *2015 Fifth International Conference on Digital Information Processing and Communications (ICDIPC)*, Oct 2015, pp. 19–23.

[24] S. Zawoad and R. Hasan, "Faiot: Towards building a forensics aware eco system for the internet of things," in *2015 IEEE International Conference on Services Computing*, June 2015, pp. 279–284.

[25] I. Sutherland, H. Read, and K. Xynos, "Forensic analysis of smart tv: A current issue and call to arms," *Digital Investigation*, vol. 11, no. 3, pp. 175 – 178, 2014, special Issue: Embedded Forensics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287614000620

[26] I. Baggili, J. Oduro, K. Anthony, F. Breitinger, and G. McGee, "Watch what you wear: Preliminary forensic analysis of smart watches," in *2015 10th International Conference on Availability, Reliability and Security*, Aug 2015, pp. 303–311.

[27] X. Feng, E. S. Dawam, and S. Amin, "A new digital forensics model of smart city automated vehicles," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, June 2017, pp. 274–279.

[28] M. Hossain, R. Hasan, and S. Zawoad, "Trust-iov: A trustworthy forensic investigation framework for the internet of vehicles (iov)," in

2017 IEEE International Congress on Internet of Things (ICIOT), June 2017, pp. 25–32.
[29] A. Goudbeek, K.-K. R. Choo, and N.-A. Le-Khac, "A forensic investigation framework for smart home environment," 08 2018, pp. 1446–1451.
[30] H. Chung, J. Park, and S. Lee, "Digital forensic approaches for amazon alexa ecosystem," *Digital Investigation*, vol. 22, pp. S15 – S25, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287617301974
[31] A. Azmoodeh, A. Dehghantanha, M. Conti, and K.-K. R. Choo, "Detecting crypto-ransomware in iot networks based on energy consumption footprint," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1141–1152, Aug 2018. [Online]. Available: https://doi.org/10.1007/s12652-017-0558-5
[32] N. Akatyev and J. I. James, "Evidence identification in iot networks based on threat assessment," *Future Generation Computer Systems*, vol. 93, pp. 814 – 821, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17300857
[33] J. Boucher and N.-A. Le-Khac, "Forensic framework to identify local vs synced artefacts," *Digital Investigation*, vol. 24, pp. S68 – S75, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287618300410
[34] Q. Do, B. Martini, and K.-K. R. Choo, "Cyber-physical systems information gathering: A smart home case study," *Computer Networks*, vol. 138, pp. 1 – 12, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618301440
[35] AccessData Corp. Command Line Imager Support, "Using Command Line Imager."
[36] Brian Carrier. Sleuthkit.org, "Autopsy - The Sleuth Kit," http://www.sleuthkit.org/autopsy/.
[37] CGSecurity. CGSecurity.org, "PhotoRec ES - CGSecurity," http://www.cgsecurity.org/wiki/PhotoRec_ES.
[38] Eric Zimmerman. Github.com, "Eric Zimmerman's tools," https://ericzimmerman.github.io/#!index.md.
[39] Harlan Carvey. Github.com, "RegRipper," https://github.com/keydet89/RegRipper2.8.
[40] niemiro - Sysnative Forums, "Restoring a Backup of the COMPONENTS Hive - What are the issues?" https://www.sysnative.com/forums/threads/restoring-a-backup-of-the-components-hive-what-are-the-issues.11691/.
[41] Raymond Chen, "The .Default user is not the default user - The Old New Thing," https://devblogs.microsoft.com/oldnewthing/20070302-00/?p=27783.
[42] Lih Wern Wong, "Forensic Analysis of the Windows Registry - ForensicFocus.com," https://www.forensicfocus.com/Content/pid=73/page=1/.
[43] Windows Dev Center, "Predefined Keys - Windows applications - Microsoft Docs," https://docs.microsoft.com/en-us/windows/desktop/sysinfo/predefined-keys.
[44] Ed Tittel, "Understanding NTUser.dat in Windows 10 - Windows Enterprise Desktop," https://searchenterprisedesktop.techtarget.com/blog/Windows-Enterprise-Desktop/Understanding-NTUserdat-in-Windows-10.
[45] Chad Tilbury, "SANS Digital Forensics and Incident Response Blog - Computer Forensic Artifacts: Windows 7 Shellbags - SANS Institute," https://digital-forensics.sans.org/blog/2011/07/05/shellbags.
[46] Margaret Rouse, "What is Windows event log? - Definition from WhatIs.com," https://searchwindowsserver.techtarget.com/definition/Windows-event-log.
[47] Chris Hoffman, "What Is the Windows Event Viewer, and How Can I Use It?" https://www.howtogeek.com/123646/htg-explains-what-the-windows-event-viewer-is-and-how-you-can-use-it/.

Análisis de la Estructura de los Contenedores Multimedia de Vídeos de Dispositivos Móviles

Carlos Quinto Huamán, Daniel Povedano Álvarez, Ana Lucila Sandoval Orozco, Luis Javier García Villalba
Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: {cquinto, dpovedano}@ucm.es, {asandoval, javiergv}@fdi.ucm.es

***Resumen*—En la actualidad, los dispositivos móviles se han convertido en el sustituto natural de la cámara digital, ya que capturan situaciones cotidianas de forma fácil y rápida, promoviendo que los usuarios se expresen a través de imágenes y vídeos. Estos vídeos pueden ser compartidos a través de diferentes plataformas quedando expuestos a cualquier tipo de manipulación, lo que compromete su autenticidad e integridad. Es común que los fabricantes no cumplan al 100 % con las especificaciones del estándar, dejando características intrínsecas del dispositivo que generó el vídeo. Las investigaciones de los últimos años se centran en el análisis de contenedores AVI, siendo muy limitado, la literatura en el caso de contenedores MP4, MOV y 3GP. En este trabajo se realiza una técnica de análisis de la estructura de los contenedores de vídeos generados por dispositivos móviles y su comportamiento al ser compartido por las redes sociales ó manipulados por programas de edición. Como resultado del análisis se tienen los siguientes resultados: verificación de la integridad de los vídeos, identificación de la fuente de adquisición y diferenciación entre vídeos originales y manipulados.**

***Index Terms*—Análisis Forense de Vídeos, Análisis de Estructuras de Contenedores, Átomos, Autenticación, Integridad.**

I. Introducción

En la actualidad el uso de los teléfonos móviles ofrece a los usuarios realizar múltiples actividades con un único dispositivo, como acceder a Internet, usar la cámara integrada, usar las múltiples aplicaciones que solucionan operaciones que antes demandaba mucho más tiempo. Esto lo convierte en uno de los dispositivos más demandados en los últimos años y se prevé un crecimiento en los próximos. Según Cisco [1], en 2022, el tráfico IP global alcanzará los 396 Exabytes mensuales (4,8 Zettabytes anuales), los usuarios de Internet aumentarán a 4.800 millones de los 3.400 millones del 2017 y habrá 28.500 millones de conexiones de dispositivos personales fijos y móviles de los 18.000 millones del 2017. Asimismo, el tráfico de vídeo IP representará un 82 % del tráfico global IP, comparado con el 75 % que se alcanzó en 2017.

Por otro lado, en [2] manifiesta que actualmente un usuario medio de Internet pasa más de 6 horas y media en línea cada día, lo que significa que la comunidad digital del mundo pasará más de 1.200 millones de años utilizando Internet en 2019. También señalan que este año existe alrededor de 3.484 millones de usuarios activos de redes sociales, que representa un 9 % más que en 2018. De estos, 3.256 millones acceden a estas a través de los teléfonos móviles. También indican que la red social Facebook es la más popular con 2.120 millones de usuarios activos mensualmente en todo el mundo, seguida por Youtube con 1.900 millones y por Whatsapp con 1.500 millones. En cuanto aplicaciones de mensajería instantánea Whatsapp es la preferida por 133 países del mundo y Facebook messenger en 75 países.

Como se ha podido notar, la tecnología provee de múltiples beneficios a la cotidianidad de la sociedad actual, sin embargo, estos beneficios se pueden convertir en un puente o conexión para que personas malintencionadas aprovechen los recursos que se encuentran sin la protección debida y realicen algún tipo de fraude o falsificación. Los vídeos capturados por dispositivos móviles no están exentos a este tipo de amenazas, porque es común compartir este tipo de ficheros por redes sociales, dispositivos de almacenamiento, incluso por la pérdida del dispositivo móvil. Por lo anterior, los vídeos están cada vez más propensos a ser manipulados y ser presentados como pruebas digitales en ámbitos procesales para evadir responsabilidades sobre acciones delictivas como pornografía infantil, tráfico de personas, etc. En este sentido, es necesario investigar sobre diferentes métodos para verificar la autenticidad e integridad de un vídeo. En esta investigación, se presenta una técnica basada en el análisis de la estructura del contenedor para autenticar y verificar la integridad de vídeos de dispositivos móviles.

Este trabajo está estructurado en 5 secciones, siendo la primera la presente introducción. En la sección 2 se describe brevemente algunos conceptos sobre vídeos digitales y su proceso de generación. La sección 3 estudia los trabajos relacionados con el análisis de estructuras de contenedores existentes en la literatura. En la sección 4 se realiza el análisis propuesto. Por último en la sección 5 se presentan las conclusiones del presente trabajo y los trabajos futuros.

II. Vídeo Digital

Un vídeo digital esta compuesto por una secuencia de imágenes, que son previamente codificados y posteriormente encapsulados en un contenedor multimedia [3]. En los dispositivos móviles, generalmente, ésta secuencia de imágenes es capturado conjuntamente con una secuencia de audio digital. El proceso de generación de un vídeo digital, también llamado *pipeline* [4] [5], es similar entre los fabricantes de dispositivos móviles, solo se aprecia una diferencia en las prestaciones adicionales como la cámara incorporada y otros detalles propios de cada fabricante. La cámara digital está compuesta por un sistema de lentes, un grupo de filtros, una matriz de filtro de colores (*Color Filter Array* (CFA)), un sensor de imagen, un procesador digital de señales (*Digital Signal Processor* (DSP)) que contiene un procesador de imagen

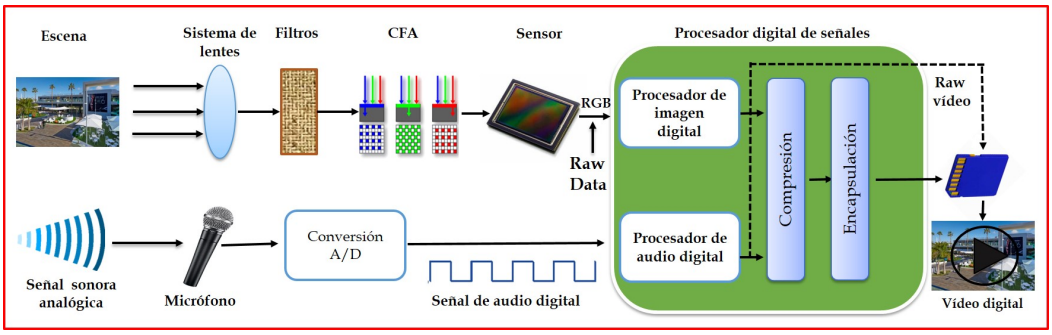


Figura 1. Proceso de generación de un vídeo digital

(*Digital Image Processor* (DIP)) y un procesador de audio digital. Adicionalmente, posee un micrófono y un conversor de señal analógica a digital [6]. En la figura 1, se muestra gráficamente el proceso de generación de un vídeo digital y su interacción con los componentes.

Para generar un vídeo, se procesa la secuencia de imágenes y audio de forma paralela [7]. El procesamiento de la secuencia de imágenes inicia cuando el sistema de lentes captura la luz de la escena controlando la exposición, el foco y la estabilización de la imagen. Esta luz ingresa a la cámara a través del sistema de lentes y aplica varios filtros para mejorar la calidad visual de la imagen (infrarrojo y anti-aliasing). Después la luz pasa al sensor de la imagen a través del CFA, que son elementos sensibles a la luz llamados píxeles. Esta señal se convierte en una señal digital, que se transmite al DSP y en concreto al DIP, que realiza diferentes procesos de cámaras para estabilizar la señal y corregir alteraciones *(artifacts)* como eliminar el ruido y otras anomalías introducidas [8] [9]. Después, la señal estabilizada pasa al proceso de compresión mediante un códec, previa sincronización y se encapsula en un contenedor multimedia. Finalmente el fichero generado se almacena en el dispositivo.

El procesamiento del audio se inicia cuando la señal sonora, transmitida por el aire, es capturada a través del micrófono que funciona como un sensor electro acústico. El micrófono transforma las ondas sonoras en una señal eléctrica para aumentar su intensidad y transmitirla a un conversor analógico digital (*Analog Digital Conversion* (ADC)), que a su vez, la convierte a una señal digital. La señal de audio digital es capturada por el DSP y en concreto por el procesador de audio digital, que la procesa para mejorar la calidad del audio antes de la compresión (control de volumen y frecuencia). Después la señal es comprimida con un algoritmo de codificación, luego se encapsula en un contenedor multimedia y se almacena. Generalmente las cámaras de los dispositivos móviles utilizan el algoritmo de compresión con pérdida (*Advanced Audio Coding* (AAC)), establecido en el estándar MPEG-4 parte 3.

II-A. Contenedores Multimedia de Vídeos

Generalmente el vídeo, audio, subtítulos y metadatos son encapsulados en un mismo contenedor multimedia. Este contenedor sigue un formato preestablecido de acuerdo a su especificación. En [10] [11] indican que un contenedor multimedia está constituido por una pista de vídeo y una de audio. Ambas pistas son comprimidas con un códec propietario de cada fabricante.

En [12] se indica que la compresión de datos se consigue mediante la eliminación de la redundancia, es decir, los componentes que no son necesarios para una reproducción fiel de los datos. Existen diferentes estándares de compresión de vídeos, pero en la actualidad los más usados por los dispositivos móviles son dos: 1) H264/AVC o MPEG-4 Parte 10 [13] y 2) H265/HEVC o MPEG-H Parte2 [14], ambos desarrollados por el ITU-T (*International Telecommunication Union* (ITU)) y ISO/IEC (*International Organization for Standardization* (ISO)), (*International Electrotechnical Commission* (IEC)).

Los contenedores más usados en la actualidad son: MP4 y MOV. El contenedor MP4, forma parte del estándar (*Moving Picture Experts Group* (MPEG))-4 parte 14, normalmente es utilizado por fabricantes que introducen el sistema operativo Android en los dispositivos móviles. El contenedor MOV, del estándar *QuickTime*, fue desarrollado por *Apple* [15], que generalmente utiliza iOS como sistema operativo de sus dispositivos. Por último, también los contenedores (*Audio Interleave de Microsoft* (AVI)) y (*Matroska* (MKV)) son usados por dispositivos más específicos.

III. Trabajos Relacionados

En los últimos años han surgido investigaciones que analizan la estructura interna de los contenedores multimedia, pero en su mayoría se centran en formatos AVI, siendo muy limitado la literatura para contenedores MP4, MOV y 3GP. Asimismo, otras analizan las estructuras de un número reducido de redes sociales y de software de edición. Se debe tener en cuenta que los fabricantes de dispositivos móviles continuamente van añadiendo nuevos átomos, etiquetas, valores y otras características relacionadas a cada marca, modelo, sistema operativo e inclusive a la versión de este último. Las redes sociales añaden características diferentes según las versiones de sus aplicaciones y según el dispositivo que se usa para compartir el vídeo. De igual forma los programas de edición dejan estructuras diferentes de acuerdo al tipo de manipulación que se aplica al vídeo y también por las versiones de los mismos.

En [16] [17], se realiza un estudio de las características que pueden ser objeto de análisis forense en dispositivos móviles. El mayor problema con este enfoque es que los diferentes modelos de las cámaras digitales usan componentes de un número reducido de fabricantes y que los algoritmos que usan para la generación de las imágenes y vídeos también son muy similares entre modelos de la misma marca.

En [18] se realiza una comparación minuciosa de los principales grupos de técnicas de identificación de la fuente de adquisición. Estas se dividen en cinco grupos y están basadas en: metadatos, características de la imagen, defectos de la matriz CFA e interpolación cromática, imperfecciones del sensor y las transformadas wavelet.

En [19] los autores realizan una técnica para verificar la integridad de vídeos con formato AVI, generados por grabadores de datos de eventos (*Video Event Data Recorders* (VEDRS)), se realizó un análisis de la estructura de 296 vídeos originales que posteriormente fueron editados por 5 programas de edición. Los resultados del análisis demostraron que los editores cambian notablemente la estructura y los valores de los metadatos con respecto los originales. Cada programa de edición incrusta una estructura específica que ayuda al analista forense a detectar si un vídeo ha sufrido algún tipo de manipulación.

En [20] se realizó un análisis de las estructuras de vídeos con formato AVI y MP4, agrupados en 19 modelos de cámaras digitales, 14 modelos de teléfonos móviles y 6 programas de edición. Después de analizar los vídeos originales los autores determinaron que las estructuras de cada tipo de contenedor no está estrictamente definida como se especifica en los estándares. Se encontraron diferencias considerables entre vídeos generados por dichos dispositivos. Asimismo, los vídeos AVI después de ser manipulados con los programas de edición, cambiaron la estructura interna incluyendo los valores de los metadatos, características esenciales para saber el origen de los vídeos.

En [21], los autores implementaron un método no supervisado para verificar la integridad de vídeos basado en la disimilitud entre vídeos originales y editados. Asimismo, desarrollaron un método para identificar la fuente de adquisición del vídeo mediante el análisis de los contenedores. Para lograr este objetivo utilizaron la librería *MP4Parser*, obteniendo ficheros de lenguaje de marcado extensible (*Extensible Markup Language* (XML)) para un posterior análisis, logrando buenos resultados en sus experimentaciones, sosteniendo que la solución utiliza un mínimo de recurso computacional a comparación de otras alternativas.

IV. Análisis de la estructura de contenedores

En este trabajo se propone una técnica para la verificación de la integridad de los vídeos, identificación de la fuente de adquisición y diferenciación entre vídeos originales y manipulados. La técnica esta compuesta por 3 fases: 1) Se extrae la información contenida de cada vídeo con el algoritmo de extracción de átomos. 2) Se analiza cada vídeo por cada dispositivo, marca, modelo, red social y programa de edición, la estructura y contenido de cada uno de los átomos extraídos. 3) Se comparan las estructuras.

En los vídeos, los átomos son características propias de cada marca, modelo, red social, programas de edición e incluso del tipo de contenedor multimedia y la versión del sistema operativo del dispositivo móvil. Los átomos están organizados de forma jerárquica *(p.ej. /moov/)*, a su vez estos átomos tienen átomos hijos *(p.ej. /moov/mvhd/)* y etiquetas *(p.ej. moov/mvhd/version)*. Asimismo, estas etiquetas contienen valores *(p.ej. Path-tag: /moov/mvhd/version, **value**: 0)*.

IV-A. Extracción de Átomos

La extracción de átomos consiste en almacenar los átomos, etiquetas, valores, orden de aparición de átomos y todo tipo de información relevante del vídeo generado por un dispositivo móvil. Este procesamiento se realiza mediante el Algoritmo de extracción de átomos 1. El proceso se inicia obteniendo el byte inicial del átomo, tamaño del átomo y el tipo de átomo con una longitud de 4 bytes, representado por una cadena de caracteres. Seguidamente, se verifica la duplicidad de átomos y la existencia de átomos hijos. Finalmente, se obtiene un diccionario de un conjunto de átomos y etiquetas (*Path-tag*) con sus respectivos valores y orden de aparición.

Algorithm 1: Extracción de Átomos de Vídeos

Input: $Video_{multimedia}$: Es el vídeo multimedia
Result: $Atomos_{extraidos}$: Son los átomos extraídos

```
1  procedure EXTRAERATOMOS(Video_multimedia)
2      DiccParam ← (Byte_ini, Tam_atom, Tipo_atom);
3      Obtener Pos_Lec;
4      Valor_Tipo ← Tipo_atom;
5      Num_Ent ← 0;
6      while Valor_tipo in Diccatomos do
7          if Valor_tipo in Diccatomos then
8              Num_Ent = Num_Ent + 1;
9              Valor_tipo = Tipo_atom + Num_Ent;
10     Diccatomos[Valor_Tipo] ← DiccParam;
11     Obtener Ruta_atomo;
12     Obtener Orden_atomo;
13     if Tipo_atom in lista_atomos then
14         if Tam_atom > 0 then
15             Obtener EtiquetaValor_atom;
16             while Byte_ini + Tam_atom > Pos_lec do
17                 Regresar paso 2;
18             if Byte_ini + Tam_atom = Pos_lec then
19                 Atomo leído correctamente
20             else
21                 Error de lectura
22         else
23             Atomo no cumple con el estándar
24     else
25         Procesar Atomo_desconocido;
26 end procedure
```

IV-B. Análisis de Estructuras de Vídeos

Para realizar el análisis se utilizó un conjunto de datos con las siguientes características: 1109 vídeos originales generados por 74 dispositivos diferentes, agrupados en 14 marcas y 57 modelos. 596 vídeos compartidos por 8 redes sociales y 276 vídeos manipulados por 4 editores más comunes.

En Tabla I se detallan los vídeos originales de los dispositivos móviles usados en el análisis, cuyo contenido parcial es propio y otros del trabajo [22].

En la Tabla II se detallan los vídeos compartidos a través de las redes sociales y editados por los programas de edición con las operaciones realizadas.

Tabla I
VÍDEOS DE DISPOSITIVOS MÓVILES USADOS PARA EL ANÁLISIS

ID Marca	Marca	ID Modelo	Modelo	Dispositivo	Contenedor	SO/Versión	Códec Vídeo	Cant
		M01	Ipad 2	D01	MOV	iOS 7.1.1	H.264	16
			Ipad 2	D02	MOV	iOS 9.3.5	H.264	10
		M02	Ipad Air	D03	MOV	iOS 11.3	H.264	10
		M03	Ipad Mini	D04	MOV	iOS 8.4	H.264	16
		M04	Iphone 4	D05	MOV	iOS 7.1.2	H.264	19
		M05	Iphone 4S	D06	MOV	iOS 7.1.2	H.264	13
			Iphone 4S	D07	MOV	iOS 8.4.1	H.264	15
		M06	Iphone 5	D08	MOV	iOS 7.0.4	H.264	10
			Iphone 5	D09	MOV	iOS 9.3.3	H.264	19
			Iphone 5	D10	MOV	iOS 8.4 (decia8.3)	H.264	32
		M07	Iphone 5C	D11	MOV	iOS 7.0.3	H.264	19
			Iphone 5C	D12	MOV	iOS 8.4.1	H.264	13
			Iphone 5C	D13	MOV	iOS 10.2.1	H.264	19
B01	APPLE	M08	Iphone 5S	D14	MOV	iOS 9.2	H.264	10
		M09	Iphone 6	D15	MOV	iOS 8.4	H.264	17
			Iphone 6	D16	MOV	iOS 9.2	H.264	10
			Iphone 6	D17	MOV	iOS 10.1.1	H.264	18
			Iphone 6	D18	MOV	iOS 11.2.0	H.264	10
		M10	Iphone 7	D19	MOV	iOS 11.2.6	H.264	10
			Iphone 7	D20	MOV	iOS 10.2.6	H.264	10
			Iphone 7	D21	MOV	iOS 11.0.3	H.264	10
		M11	Iphone 7 Plus	D22	MOV	iOS 11.0.3	H.264	10
		M12	Iphone 8	D23	MOV	iOS 12.1.4	H.264	10
		M13	Iphone 8 Plus	D24	MOV	iOS 11.2.5	H.265	9
			Iphone 8 Plus	D25	MOV	iOS 11.2.6	H.265	7
		M14	Iphone 6 Plus	D26	MOV	iOS 10.2.1	H.264	19
		M15	Iphone XR	D27	MOV	iOS 12.1.1	H.264	10
			Iphone XR	D28	MOV	iOS 12.1.4	H.265	10
B02	ASUS	M16	Zenfone2 Laser	D29	MP4	–	H.264	19
B03	BQ	M17	Aquaris E5	D30	MP4	Android	H.264	10
		M18	Aquaris E4.5	D31	MP4	Android	H.264	10
		M19	Ascend G6-U10	D32	MP4	–	H.264	19
		M20	P8 GRA-L09	D33	MP4	Android 6.0/GRA-L09C55B330	H.264	19
		M21	P9 EVA-L09	D34	MP4	Android 6.0/EVA-L09C55B190	H.264	19
B04	HUAWEI	M22	P9 Lite VNS-L31	D35	MP4	Android 6.0/VNS-L31C02B125	H.264	19
		M23	Y635-L01	D36	MP4	Android	H.264	10
			Y635-L01	D37	MP4	Android	H.264	11
		M24	Honor 5C NEM-L51	D38	MP4	Android 6.0/NEM-L51C432B120	H.264	19
B05	LENOVO	M25	P70A	D39	3GP	–	H.264	19
		M26	D290	D40	MP4	–	H.264	19
B06	LG	M27	Nexus 5	D41	MP4	Android	H.264	10
			Nexus 5	D42	MP4	Android	H.264	26
B07	MICROSOFT	M28	Lumia 640 LTE	D43	MP4	Windows Phone	H.264	10
		M29	Moto G1	D44	MP4	Android	H.264	10
B08	MOTOROLA		Moto G1	D45	MP4	Android	H.264	10
		M30	Moto G2	D46	MP4	Android	H.264	10
		M31	Nexus 6	D47	MP4	Android	H.264	10
B09	NOKIA	M32	808 Pureview	D48	MP4	Symbian	H.264	10
		M33	A0001	D49	MP4	Android	H.264	20
B10	ONE PLUS	M34	A3000	D50	MP4	Android 7.0/NRD90M 15 dev-keys	H.264	19
		M35	A3003	D51	MP4	Android 7.0/NRD90M 138 dev-keys	H.264	19
		M36	Galaxy A6	D52	MP4	Android	H.264	20
		M37	Galaxy Nexus	D53	MP4	Android	H.264	20
		M38	Galaxy S III Mini GT-I8190N	D54	MP4	I8190NXXAML1, I8190NXXALL6	H.264	22
		M39	Galaxy S III Mini GT-I8190	D55	MP4	I8190XXAMG4	H.264	16
		M40	Galaxy S3	D56	MP4	Android	H.264	10
		M41	Galaxy S3 GT-I9300	D57	MP4	–	H.264	19
		M42	Galaxy S3 Neo GT-I93011	D58	MP4	Android	H.264	9
		M43	Galaxy S4 Mini GT-I9195	D59	MP4	Android	H.264	20
B11	SAMSUNG		Galaxy S4 Mini GT-I9195	D60	MP4	I9195XXUCNK1	H.264	19
		M44	Galaxy S4	D61	MP4	Android	H.264	10
		M45	Galaxy S5	D62	MP4	Android	H.264	10
		M46	Galaxy S5 SM-G900F	D63	MP4	Android 6.0.1/G900FXXS1CQAA	H.264	19
		M47	Galaxy S6	D64	MP4	Android	H.264	10
		M48	Galaxy S9 Plus	D65	MP4	Android	H.264	20
		M49	Galaxy J5 2016	D66	MP4	Android 6.0.1/J510MNUBU2AQI1	H.264	12
		M50	Galaxy Tab 3 GT-P5210	D67	MP4	P5210XXUBNK2	H.264	37
		M51	Galaxy Trend Plus GT-S7580	D68	MP4	S7580XXUBOA1	H.264	16
B12	SONY	M52	Xperia M2 D2303	D69	MP4	Android	H.264	20
		M53	Xperia Z1 Compact D5503	D70	MP4	14.5.A.0.270_6_f100000f	H.264	19
B13	WIKO	M54	Ridge 4G	D71	MP4	–	H.264	11
		M55	Mi3	D72	MP4	Android	H.264	13
B14	XIAOMI	M56	Redmi Note 3	D73	MP4	Android 6.0.1/MNEXUS 5MB29M	H.264	19
		M57	Redmi Note 5	D74	MP4	Android 8.1.0/OPM1.171019	H.264	10

Tabla II
VÍDEOS COMPARTIDOS Y EDITADOS USADOS PARA EL ANÁLISIS

ID Modelo Original	ID Red Social	Red Social	Versión	Opciones		Contenedor Editado	Cant. Vídeos
				Subida	Descarga		
D02,D03,D08,D14,D16,D20,D24	R01	Whatsapp	2.19.20	–	Compartidos	MP4	69
D02,D03,D08,D14,D16,D20,D24	R02	Facebook HD	Web Page	Max 4gb, 240min	Inspect element(Firefox)	MP4	69
D02,D03,D08,D14,D16,D20,D24	R03	Facebook SD	Web Page	Max 4gb, 240min	Inspect element(Firefox)	MP4	69
D02,D03,D08,D14,D16,D20,D24	R04	Telegram	5.0 M2	Max 1.5gb	Compartidos	MP4	69
D02,D03,D08,D14,D16,D20,D24	R05	Youtube	Web Page	Max 128gb,12hrs	Youtube studio beta	MP4	69
D02,D03,D08,D14,D16,D20,D24	R06	Flickr	Web Page	max 1gb	Web page(save as)	MP4	69
D02,D03,D08,D14,D16,D20,D24	R07	Linkedin	Web Page	Max 6gb, min 75kb	Web Page(save as)	MP4	69
D02,D03,D08,D41,D62,D72	R08	Twitter	Web Page	Max 500mb, 2.20 min	Firefox(twitervideodownloader)	MP4	63
D03,D18,D19,D52	R09	Instagram	Web Page	Max 10 min, ratio 9:16	Inspect element(Firefox)	MP4	50
ID Modelo Original	**ID Soft.**	**Programa**	**Versión**	**Opciones**	**Contenedor**	**Cantidad**	
D02,D03,D08,D14,D16,D20,D24	S01	Adobe Premier	2018-12.0	Abrir/Guardar	MP4	69	
D02,D03,D08,D14,D16,D20,D24	S02	Camtasia	2018.0.1	Abrir/Guardar	MP4	69	
D02,D03,D08,D14,D16,D20,D24	S03	Ffmpeg	3.4.4	Simple (Copy -vcodec)	MOV	69	
D02,D03,D08,D14,D16,D20,D24	S04	Lightworks	14.5	Importar/Exportar	MP4	69	

En la Figura 2, se observa un DataFrame con 15 registros de un vídeo original generado por un dispositivo móvil de marca apple y modelo Iphone 6 (X) y en la Figura 3, se muestra otro DataFrame con 13 registros del mismo vídeo X, después de ser subido a Youtube (X'). Al compararse los registros de X y X' se detectan las siguientes diferencias:

	File Name	Maker	Model	Path-tag	Value	Reading Orders
0	IMG_0348.MOV	Apple	iPhone 6	/ftyp/	ftyp	1
1	IMG_0348.MOV	Apple	iPhone 6	/ftyp/majorBrand	qt	1
2	IMG_0348.MOV	Apple	iPhone 6	/ftyp/minorVersion	0	1
3	IMG_0348.MOV	Apple	iPhone 6	/ftyp/compatibleBrands	qt	1
4	IMG_0348.MOV	Apple	iPhone 6	/wide/	wide	2
5	IMG_0348.MOV	Apple	iPhone 6	/mdat/	mdat	3
6	IMG_0348.MOV	Apple	iPhone 6	/moov/	moov	4
7	IMG_0348.MOV	Apple	iPhone 6	/moov/mvhd/	mvhd	5
25	IMG_0348.MOV	Apple	iPhone 6	/moov/trak/	trak	6
170	IMG_0348.MOV	Apple	iPhone 6	/moov/trak1/	trak	34
293	IMG_0348.MOV	Apple	iPhone 6	/moov/trak2/	trak	60
450	IMG_0348.MOV	Apple	iPhone 6	/moov/trak3/	trak	108
602	IMG_0348.MOV	Apple	iPhone 6	/moov/meta/ilst/data1/value	Apple	148
608	IMG_0348.MOV	Apple	iPhone 6	/moov/meta/ilst/data2/value	iPhone 6	149
658	IMG_0348.MOV	Apple	iPhone 6	/moov/free1/trex3/default_sample_flags	0	157

Figura 2. DataFrame original de *X*, Marca: Apple, Modelo: Iphone6.

	File Name	Maker	Model	Path-tag	Value	Reading Orders
0	IMG_0348.mp4	Unknown	Unknown	/ftyp/	ftyp	1
1	IMG_0348.mp4	Unknown	Unknown	/ftyp/majorBrand	mp42	1
2	IMG_0348.mp4	Unknown	Unknown	/ftyp/minorVersion	0	1
3	IMG_0348.mp4	Unknown	Unknown	/ftyp/compatibleBrands	isommp42	1
4	IMG_0348.mp4	Unknown	Unknown	/moov/	moov	2
5	IMG_0348.mp4	Unknown	Unknown	/moov/mvhd/	mvhd	3
6	IMG_0348.mp4	Unknown	Unknown	/moov/mvhd/version	0	3
7	IMG_0348.mp4	Unknown	Unknown	/moov/mvhd/flags	0	3
23	IMG_0348.mp4	Unknown	Unknown	/moov/trak/	trak	4
126	IMG_0348.mp4	Unknown	Unknown	/moov/trak1/	trak	23
218	IMG_0348.mp4	Unknown	Unknown	/moov/udta/	udta	41
236	IMG_0348.mp4	Unknown	Unknown	/moov/udta/meta/ilst/too/data/value	Google	46
249	IMG_0348.mp4	Unknown	Unknown	/mdat/	mdat	49

Figura 3. DataFrame de X después de pasar por Youtube X'.

En X, se observa la existencia del atomo */ftyp/* con las etiquetas */ftyp/majorBrand* y */ftyp/compatibleBrands* con el mismo valor *"qt"*. También contiene los átomos */wide/*, */mdat/* y */moov/* en el orden 2, 3 y 4 respectivamente. Además, tienen cuatro átomos */moov/trak*. Después aparecen los átomos */data1/* y */data2/* que contienen las etiquetas */moov/meta/ilst/data1/value* y */moov/meta/ilst/data2/value*, siendo el primer valor *"Apple"*, que indica la marca del dispositivo que generó el vídeo y el segundo valor *"Iphone 6"*, que indica el modelo del dispositivo. Por último, se observa que contiene 659 registros entre átomos y etiquetas.

En cambio, X', tiene en el átomo */ftyp/* dos etiquetas */ftyp/majorBrand* y */ftyp/compatibleBrands* con los valores *"mp42"* y *"isommp42"* respectivamente. El orden de aparición del átomo */moov/* varía a 2 y en */mdat/* a 49. El átomo */wide/* desaparece y se reduce los átomos */moov/trak* de cuatro a dos respectivamente. Asimismo, desaparecen los átomos */data1/* y */data2/* que normalmente proveen la marca y modelo del dispositivo. Se adiciona el átomo */data/* que contiene la etiqueta */moov/udta/meta/ilst/too/data/value* con el valor *"Google"* que es la marca propietaria de la plataforma Youtube.

Por último, se observa que se reduce de 659 a 250 el número de registros entre átomos y etiquetas con respecto al vídeo X. En esta comparación inicial se aprecia que la red social Youtube cambia drásticamente el contenido del vídeo original. Este procedimiento lo efectúan generalmente casi todas las redes sociales y editores de vídeo para reducir el tamaño del fichero, facilitar su transferencia y optimizar la gestión de almacenamiento en sus plataformas.

En el análisis anterior, se observó que la estructura jerárquica del contenedor multimedia tiene la forma: *Path-tag:* "/moov/mvhd/version", que incluye átomos y etiquetas. Esta etiqueta posee un valor: "0". Por tanto, un vídeo posee una lista de N *Path-tag*.

Para consolidar una estructura única por dispositivo, modelo, marca, red social y programa de edición, se utiliza la presencia o ausencia de los *Path-tag*. Para realizar un análisis masivo se utiliza, las variables binarias (0 ó 1), asignando el valor 1 para cada *Path-tag* que se encuentre presente y el valor de 0 para cada *Path-tag* que se encuentre ausente en el vídeo, obteniendo estructuras con igual cardinalidad.

El primer objetivo es verificar si los fabricantes incrustan un patrón único de *Path-tag* a dispositivos del mismo modelo y marca, pero con características distintas.

En la Tabla III, se observa que los dispositivos D01 y D02, ambos de marca Apple y modelo Ipad 2, tienen estructuras considerablemente diferentes.

Tabla III
ANÁLISIS DE DISPOSITIVOS ENTRE MODELOS SIMILARES

ID Dispositivo	Estructura	Átomos	Etiquetas	Total	Dif
D01	379	82	297	684	305
D02	651	155	496	684	33
D06	387	84	303	387	0
D07	357	74	277	387	36
D08	364	77	287	636	272
D09	613	147	466	636	23
D10	343	72	271	636	293
D11	379	82	297	659	280
D12	351	74	277	659	308
D13	621	149	472	659	38
D15	351	74	277	660	309
D16	659	157	502	660	1
D17	613	147	466	660	47
D18	659	157	502	660	1
D19	629	151	478	783	154
D20	641	153	488	783	142
D21	673	160	513	783	110
D24	685	161	524	685	0
D25	647	153	494	685	38
D27	639	152	487	734	95
D28	690	161	529	734	44
D36	225	43	182	225	0
D37	225	43	182	255	0
D41	250	50	200	250	0
D42	248	49	199	250	2
D44	226	44	182	226	0
D45	226	44	182	226	0
D59	221	42	179	221	0
D60	221	42	179	221	0

Los vídeos del dispositivo D001 poseen 379 *Path-tag* considerados como la estructura, divididos en 82 átomos y 397 etiquetas. Por el contrario, los vídeos del dispositivo D02 poseen 651 *Path-tag* considerados como la estructura del dispositivo, divididos en 155 átomos y 496 etiquetas. Ambos dispositivos tienen 684 *Path-tag* únicos para el modelo, eso

hace notar que el dispositivo D01 tiene una diferencia de 305 *Path-tag* con respecto al total consolidado, representando también el número de ceros asignados por la ausencia de estos *Path-tag*. El dispositivo D02 tiene una diferencia de 33 *Path-tag* con respecto al total consolidado, representando también el número de ceros asignados por la ausencia de estos *Path-tag*. Este fenómeno se debe a que el dispositivo D01 tiene el sistema operativo iOs con versión 7.1.1 y el dispositivo D02 tiene la versión iOs 9.3.5. Es evidente que los fabricantes controlan la codificación del vídeo en el sistema operativo del dispositivo, entre más reciente versión, mayor probabilidad de aparición de nuevos átomos y etiquetas. Los tres dispositivos (D19, D20, D21) de marca Apple y modelo Iphone 7, también tienen diferencias de aparición de *Path-tag*, el dispositivo D19 tiene 619 *Path-tag*, divididos en 151 átomos y 478 etiquetas. Mientras que D20 tiene 641 *Path-tag*, divididos en 153 átomos y 488 etiquetas. En cambio, el dispositivo D21 tiene 673 *Path-tag*, divididos en 160 átomos y 513 etiquetas. Estos últimos dispositivos consolidan un total de 783 *Path-tag* únicos para el modelo, demostrando que el D19 tiene una diferencia de 154 átomos con respecto al total consolidado, el D20 tiene 142 y el D21 posee 110. Estas diferencias son en número de ceros asignados a cada dispositivo por la ausencia de *Path-tag*. Esta última verificación demuestra que los dispositivos de marca Apple con similar modelo pero con diferente versión de sistema operativo, poseen diferente estructura. De forma análoga pasa con los demás dispositivos (D24, D25), (D27, D28), (D41, D42) que son marcas que generan vídeos con formato MP4. No obstante, los dispositivos (D36, 37), (D44, D45), (D59, D60) mantienen una estructura similar a pesar de ser dispositivos diferentes.

Para realizar el análisis de los vídeos compartidos por las redes sociales, se seleccionaron 13 dispositivos y el total de vídeos de cada dispositivo fueron compartidos ó subidos y descargados por las redes sociales de acuerdo a las opciones detalladas en la Tabla II. Asimismo, se consideró a la red social Facebook dos veces debido a que dependiendo a la forma de descargar los vídeos, las estructuras cambian considerablemente. Por tanto, en la Tabla IV se observa que red social Facebook HD mantiene una estructura de 267 *Path-tag*, dividido en 55 átomos y 212 etiquetas, excepto el dispositivo D08 de marca Apple y modelo Iphone 5, que presenta una estructura de 274 *Path-tag* dividido en 57 átomos y 217 etiquetas. Este último dispositivo tiene 7 *Path-tag* adicionales con respecto a los demás, y son los siguientes:

- /moov/udta/meta/ilst/day/
- /moov/udta/meta/ilst/day/data/
- /moov/udta/meta/ilst/day/data/country_indicator
- /moov/udta/meta/ilst/day/data/language_indicator
- /moov/udta/meta/ilst/day/data/typeIndicator_field1
- /moov/udta/meta/ilst/day/data/typeIndicator_field2
- /moov/udta/meta/ilst/day/data/value

La red social Facebook SD posee una estructura de 262 *Path-tag*, dividido en 54 átomos y 208 etiquetas, excepto el dispositivo D08, que cuenta con 269 *Path-tag*, divididos en 56 átomos y 213 etiquetas. Este último, nuevamente posee los 7 *Path-tag* insertados por Facebook HD. La red social Flickr mantiene el 100 % de *Path-tag* de los dispositivos originales, inclusive mantienen el mismo tamaño del fichero. La red social Linkedin mantiene una estructura de 258 rutas, 53 átomos y 205 etiquetas, excepto el dispositivo D08, que tiene 265 rutas representado por 55 átomos y 210 etiquetas. Igualmente, este último tiene los 7 *Path-tag* insertados por Facebook HD. La red social Telegram tiene una estructura fija de 251 *Path-tag* divididas en 49 átomos y 202 etiquetas. La red social Whatsapp también posee una estructura fija de 237 *Path-tag* divididos en 46 átomos y 191 etiquetas. Finalmente, la red social Youtube mantiene una estructura constante de 250 *Path-tag* divididas en 51 átomos y 199 etiquetas.

Para el análisis de vídeos de la red social Instagram se seleccionaron 4 dispositivos que produjeron vídeos con características que requería la red social (ver la Tabla II). En la Tabla IV se observa que esta red social inserta una estructura similar a vídeos de tres dispositivos (D03, D18, D19); contiene 255 *Path-tag* que se dividen en 52 átomos y 203 *Path-tag*. Por el contrario, los vídeos del dispositivo D52 tienen 260 *Path-tag*, es decir 5 más. Los *Path-tag* adicionales son:

- /moov/trak/mdia/minf/stbl/ctts/
- /moov/trak/mdia/minf/stbl/ctts/entries
- /moov/trak/mdia/minf/stbl/ctts/entryCount
- /moov/trak/mdia/minf/stbl/ctts/flags
- /moov/trak/mdia/minf/stbl/ctts/version

Para el análisis de la red social Twitter se usaron vídeos de 7 dispositivos móviles. Esta red social inserta una estructura fija a todos los dispositivos, teniendo un total de 260 *Path-tag*, divididos en 53 átomos y 207 etiquetas.

Después realizar este análisis se concluye que todas las redes sociales, excepto Flickr, aplican sobre los vídeos una compresión adicional para reducir su tamaño, facilitar la transferencia

Tabla IV
ANÁLISIS DE VÍDEOS COMPARTIDOS POR REDES SOCIALES

Redes/S	**FacebookHD**			**FacebookSD**			**Flickr**			**Linkedin**			**Telegram**			**Whatsapp**			**Youtube**			**Instagram**			**Twitter**		
Dispositivo	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**
D02	55	212	267	54	208	262	155	496	651	53	205	258	49	202	251	46	191	237	51	199	250	–	–	–	53	207	260
D03	55	212	267	54	208	262	165	538	703	53	205	258	49	202	251	46	191	237	51	199	250	52	203	255	53	207	260
D08	57	217	274	56	213	269	77	287	364	55	210	265	49	202	251	46	191	237	51	199	250	–	–	–	53	207	260
D14	55	212	267	54	208	262	157	502	659	53	205	258	49	202	251	46	191	237	51	199	250	–	–	–	–	–	–
D16	55	212	267	54	208	262	157	502	659	53	205	258	49	202	251	46	191	237	51	199	250	–	–	–	–	–	–
D18	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	52	203	255	–	–	–
D19	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	52	203	255	–	–	–
D20	55	212	267	54	208	262	153	488	641	53	205	258	49	202	251	46	191	237	51	199	250	–	–	–	–	–	–
D24	55	212	267	54	208	262	161	524	685	53	205	258	49	202	251	46	191	237	51	199	250	–	–	–	–	–	–
D41	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	53	207	260
D52	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	53	207	260	–	–	–
D62	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	53	207	260
D73	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	53	207	260

y así optimizar la gestión de almacenamiento en sus plataformas.

Para realizar el análisis de vídeos manipulados por programas de edición, se seleccionaron 7 dispositivos. Los vídeos de cada dispositivo fueron manipulados por cada programa de edición, según las opciones detalladas en la Tabla II. Se observa en la Tabla V, que todos los programas de edición insertan una estructura fija. El programa Lightworks inserta 239 *Path-tag* divididas en 47 átomos y 192 etiquetas. La librería Ffmpeg incrusta 283 *Path-tag* que representa 56 átomos y 227 etiquetas. El programa Camtasia inserta 264 *Path-tag* divididas en 54 átomos y 210 etiquetas. Finalmente, Adobe Premier introduce 398 *Path-tag* divididas en 77 átomos y 321 etiquetas.

Al realizar este análisis preliminar se ha logrado agrupar las características o *Path-tag* únicos por modelos, marcas, redes sociales y programas de edición a fin de lograr los objetivos de este trabajo: verificar la integridad de los vídeos, identificar la fuente de adquisición y distinguir entre vídeos originales y manipulados.

Tabla V
ANÁLISIS DE VÍDEOS MANIPULADOS CON PROGRAMAS DE EDICIÓN

Editor	**Lightworks**			**Ffmpeg**			**Camtasia**			**Adobe Premier**		
Marca	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**	**C/A**	**C/E**	**E**
D02	47	192	239	56	227	283	54	210	264	77	321	398
D03	47	192	239	56	227	283	54	210	264	77	321	398
D08	47	192	239	56	227	283	54	210	264	77	321	398
D14	47	192	239	56	227	283	54	210	264	77	321	398
D16	47	192	239	56	227	283	54	210	264	77	321	398
D20	47	192	239	57	231	288	54	210	264	77	321	398
D24	47	192	239	57	231	288	54	210	264	77	321	398

IV-C. Comparación de Estructuras

Para la comparación entre modelos, marcas, redes sociales y programas de edición se recurre a las características consolidas de presencia o ausencia de *Path-tag* (1 ó 0). Para ello, se utilizó como medida de similitud el coeficiente de correlación de pearson que posee propiedades para variables binarias y toma un valor en el rango [-1,1]. Si su valor es 1 indica que las dos variables están perfectamente relacionadas; si es 0 no hay relación lineal entre ellas y si es negativo es que existe una correlación negativa. En este trabajo, la comparación entre modelos, marcas, redes sociales y programas de edición; mientras la correlación de las variables sean negativas, existe una mayor diferencia entre ellas.

En la Figura 4 se muestra la matriz de correlación por modelos, agrupados por dispositivos que generan vídeos con formato MOV, donde se aprecia que los modelos Iphone 4 y Iphone 4S tienen como valor 1, eso indica que ambos modelos están perfectamente relacionados. También los modelos Iphone 5S y Iphone 6 tienen una correlación positiva muy alta. Por el contrario los modelos Iphone 8 y Iphone XR alcanzan una correlación negativa moderada. En este análisis por modelos se puede concluir que las estructuras de los vídeos MOV agrupados por modelos tienen más diferencias que relaciones entre ellos. Estas diferencias son elementos sustanciales que permiten verificar la integridad del vídeo analizado y conocer el modelo del dispositivo que lo generó.

En la Figura 5 se muestra la matriz de correlación por redes sociales, donde se aprecia que la red social Flickr alcanza una correlación negativa moderada frente a las demás redes sociales, esto debido a que los vídeos no sufren una compresión adicional, es decir, son similares a los originales. Por otro lado, Facebook HD y Facebook SD, alcanzan una correlación positiva moderada. Después de este análisis se puede concluir también que existen diferencias en la estructura de vídeos que fueron compartidos por las redes sociales que ayuda notablemente a la verificación de la integridad, detección la manipulación por este tipo de plataformas.

En la Figura 6 se muestra la matriz de correlación por programas de edición de vídeos, donde se aprecia claramente que existe una correlación positiva perfecta solo entre mismos programas de edición. Estas diferencias notables entre distintos programas sirven definitivamente para identificar que el vídeo fue modificado por un programa de edición específico.

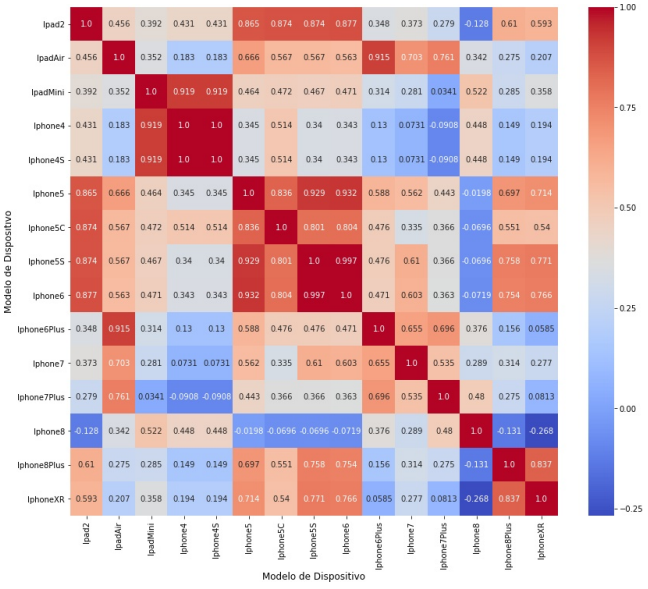


Figura 4. Matriz de correlación por modelos de dispositivos (MOV)

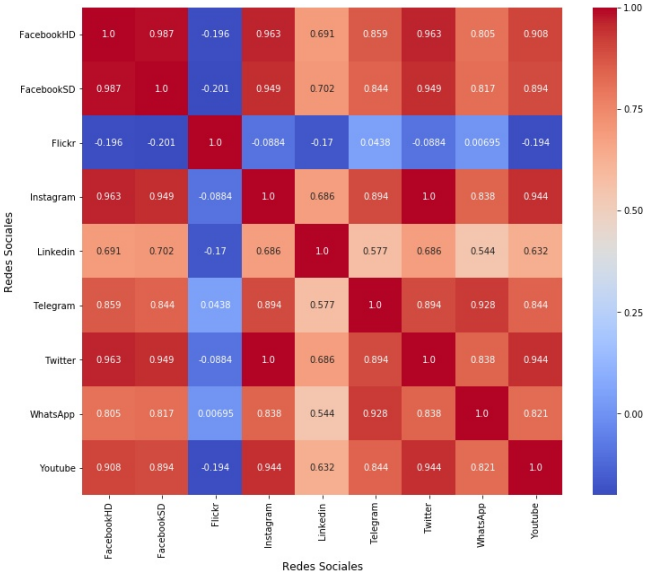


Figura 5. Matriz de correlación por redes sociales

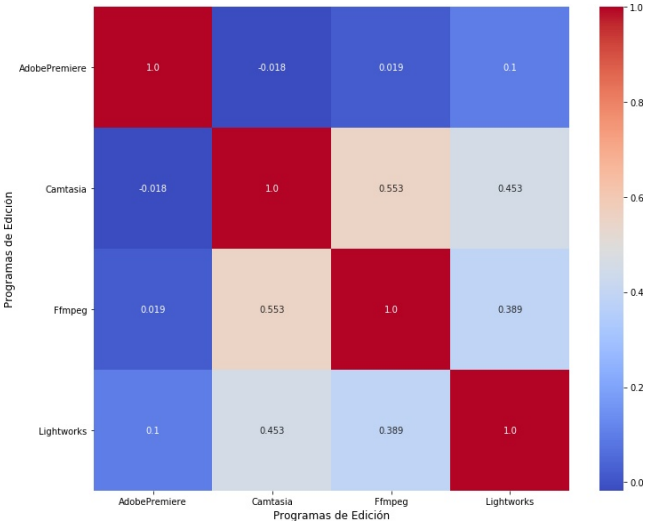


Figura 6. Matriz de correlación por programas de edición

V. Conclusiones

En este trabajo se presenta una técnica para realizar el análisis de la estructura intrínseca de los contenedores de vídeos generados por dispositivos móviles, compartidos en redes sociales y editados por programas de edición, con el objetivo de verificar la integridad de los vídeos, identificar la fuente de adquisición y diferenciar entre vídeos originales y manipulados.

Para lograr este objetivo se realizó 3 pasos: primero la extracción del contenido de los vídeos con el algoritmo de extracción de átomos, segundo, el análisis del contenido extraído y por último la comparación de estructuras de vídeos por modelo, red social y editor.

En la comparación entre modelos, solo los modelos Iphone 4 y Iphone 4S tienen una estructura similar, mientras que los demás modelos tienen diferencias notables entre ellas.

En la comparación entre redes sociales, se observó que cada red social mantiene un patrón único y tienen diferencias considerables entre ellos. No obstante, Flickr es la única red social que no vuelve a comprimir el vídeo, manteniendo su estructura original.

En la comparación entre programas de edición, se observó también que cada programa de edición deja una estructura fija en los contenedores multimedia.

Estas diferencias y patrones únicos son fundamentales para que el analista forense determine si un vídeo fue o no manipulado.

Como trabajos futuros se han planificado los siguientes:

- Analizar el orden de aparición de los átomos y de los valores que contienen cada una de la etiquetas para realizar una verificación más exacta de la estructura de los vídeos.
- Identificar la fuente de adquisición de vídeos mediante un algoritmo no supervisado basado al análisis de la estructura del contenedor.
- Realizar la identificación de la fuente de adquisición de vídeos mediante un algoritmo supervisado basado al análisis de la estructura del contenedor.

Agradecimientos

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326. Website: http://ramses2020.eu

Referencias

[1] CISCO, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper," February 2019.

[2] HOOTSUITE, " Global Digital Reports 2019," pp. 1–221, February 2019.

[3] A. Murat Tekalp, *Digital Video Processing*. Massachusetts, USA: Prentice Hall, June 2015.

[4] M. Tennoe, E. Helgedagsrud, M. Naess, H. Kjus Alstad, H. Kvale Stensland, V. Reddy Gaddam, D. Johansen, C. Griwodz, and P. Halvorsen, "Efficient Implementation and Processing of a Real-Time Panorama Video Pipeline," in *Proceedings of the 2013 IEEE International Symposium on Multimedia*, Washington, DC, USA, December 2013, pp. 76–83.

[5] Texas Instruments Incorporated, "Digital Still Camera," 2012. [Online]. Available: http://www.ti.com/solution/digital_still_camera

[6] S. Bayram, H. T. Sencar, and N. Memon, "Classification of Digital Camera-Models Based on Demosaicing Artifacts," *The International Journal of Digital Forensics & Incident Response*, vol. 5, no. 2, pp. 49–59, September 2008.

[7] Q. H. C., "Técnicas Antiforenses para Vídeos de Dispositivos Móviles," Facultad de Informática, Universidad Complutense de Madrid, Tesis de Máster, Setiembre 2016.

[8] P. Corporation, "Lumix Digital Camera Know-Hows," 2012. [Online]. Available: http://av.jpn.support.panasonic.com/support/global/cs/dsc/knowhow/index.html

[9] J. Nakamura, *Image Sensors and Signal Processing for Digital Still Cameras*. Boca Raton, FL, USA: CRC Press, August 2005.

[10] J. Kaur and N. Sharma, "Survey on the General Concepts of MPEG Moving Picture Experts Group," *Paripex: Indian Journal of Research*, vol. 5, no. 2, pp. 252–255, February 2016.

[11] B. G. Haskell, P. A., and N. A. N., *Digital Video: An Introduction to MPEG-2 Digital Multimedia Standards*. Orlando, FL, USA: Springer US, 2007.

[12] S. Dhanani and M. Parker, *Digital Video Processing for Engineers: A Foundation for Embedded Systems Design*. Newton, MA, USA: Newnes, 2012.

[13] I. T. Union, "Advanced Video Coding for Generic Audiovisual Services H.264," 2016. [Online]. Available: http://www.itu.int/

[14] International Telecommunication Union, "High Efficiency Video Coding," 2018. [Online]. Available: http://www.itu.int/

[15] Q. F. Format, "QuickTime File Format Specification," 2016. [Online]. Available: https://developer.apple.com

[16] V. Thing, K. Ng, and E. C. Chang, "Live Memory Forensics of Mobile Phones," *Digital Investigation*, vol. 7, pp. 74–82, September 10.

[17] T. Van Lanh, K. S. Chong, S. Emmanuel, and M. S. Kankanhalli, "A Survey on Digital Camera Image Forensic Methods," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Beijing, July 2007, pp. 16–19.

[18] A. Sandoval Orozco, D. Arenas González, J. Rosales Corripio, L. García Villalba, and J. C. Hernandez-Castro, "Techniques for Source Camera Identification," in *Proceedings of the 6th International Conference on Information Technology*, Amman, Jordan, May 2013, pp. 1–9.

[19] J. Song, K. Lee, W. Y. Lee, and L. H., "Integrity Verification of the Ordered Data Structures in Manipulated Video Content," *Digital Investigation*, vol. 18, no. C, pp. 1–7, Septiembre 2016.

[20] T. Gloe, A. Fisher, and M. Kirchner, "Forensic Analysis of Video File Formats," in *Proceedings of the First Annual DFRWS Europe*, Munster, Germany, May 2014, pp. 68–76.

[21] M. Iuliani, D. Shullani, M. Fontani, M. S., and A. Piva, "A video forensic framework for the unsupervised analysis of mp4-like file container," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 635–645, March 2018.

[22] D. Shullani, M. Fontani, M. Iuliani, O. A. Shay, and A. Piva, "VISION: A Video and Image Dataset for Source Identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 15, October 2017.

Improving Speed-Accuracy Trade-off in Face Detectors for Forensic Tools by Image Resizing

Deisy Chaves
Departamento IESA
Universidad de León
Researcher at INCIBE
dchas@unileon.es

Eduardo Fidalgo
Departamento IESA
Universidad de León
Researcher at INCIBE
efidf@unileon.es

Enrique Alegre
Departamento IESA
Universidad de León
Researcher at INCIBE
enrique.alegre@unileon.es

Pablo Blanco
Departamento IESA
Universidad de León
Researcher at INCIBE
pblanm@unileon.es

***Abstract*—During forensic material analysis, accurate and fast face detection is required prior to facial recognition of fugitives or children in sexual abuse scenes. However, this is not easy due to common limitations in the image quality or face pose. Moreover, real-time performance is expected in some applications as the forensic ones. There are several methods to address the face detection problem, but most of them are not suitable for real-time applications due to its computational complexity. In this work, we propose a strategy based on image resizing especially valid for Child Sexual Abuse crimes, oriented to improve the trade-off between the speed and the performance of three deep-learning-based face detectors. The results showed that the proposed approach is able to speed up face detection with a small reduction in accuracy. The best speed-accuracy trade-off is achieved using images resized to 50% of the original image size.**

***Index Terms*—Face detection, Forensic images, CSA, Deep learning**

Contribution type: *Ongoing Research*

I. Introduction

Face detection has applications in different fields such as security, bio-metrics and health-care. This process is crucial in some forensics and law enforcement [1] activities, since an accurate and fast face detection is required as an previous step in other tasks such as surveillance, fugitives recognition, sexual abuse detection, among others. However, several factors difficult a correct face detection, such as are variations in pose, expression, image resolution, and illumination [2], which are common issues found in some forensic images.

There are several approaches to address the face detection problem. In [3], Viola and Jones presented the first framework for real-time face detection based on a sliding window search using the AdaBoost algorithm with hand-crafted features (Haar descriptors). Recently, most face detectors focus on using features learnt from a Convolutional Neural Network (CNN) [4], [5], [6], [7], [8], which increase performance significantly in complex detection conditions, e.g. low/high illumination, blur and face occlusion. Nevertheless, most of these face detectors are focused on improving detection accuracy under challenging conditions without taking into account the processing time. Thus, their application for analysing large amounts of data where real-time performance is desired, e.g. forensics images, requires the development of strategies to speed up detection.

In this work, we evaluated the trade-off between processing time and accuracy detection through an image resizing strategy. We assessed three of the best, in terms of speed or accuracy, face detection methods presented in the last three years —Multi-Task Cascade CNN (MTCNN) [4], PyramidBox [8] and Dual Face Shot Detector (DSFD) [7]— using a set of images selected from the dataset UFDD [9] with similar characteristics in regards to the number of people found in Child Sexual Abuse (CSA) images. We are interested in this problem because this work is framed on the European project Forensic Against Sexual Exploitation of Children (4NSEEK) and in the research lines defined by the Framework agreement between INCIBE and the University of León. Preliminary results showed that our strategy improves the processing time of the evaluated face detector methods with a moderate reduction in accuracy.

II. Methodology

Fig. 1 shows the data flow of the strategy used to improve face detection processing time. First, images are resized to a percentage of their original size to keep the proportions of faces and objects contained on the image. Second, face detection is performed on the resized image using a state-of-art method. Finally, detected bounding boxes containing face locations are scaled back to the original image dimensions and returned as output.

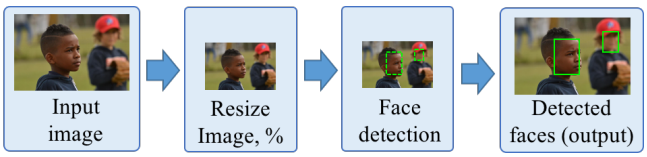
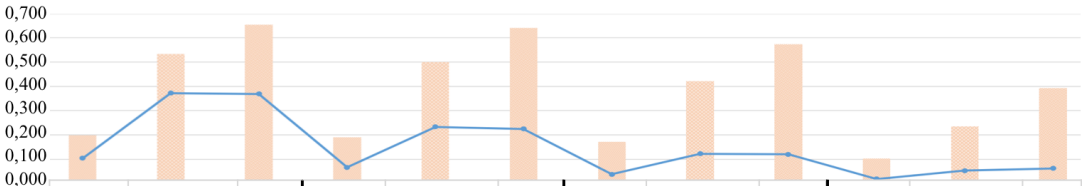


Figure 1. Proposed face detection strategy

We selected the three deep-learning-based methods indicated previously for face detection. The first method, MTCNN [4] performs both face detection and alignment using a multi-task training. Custom networks are used for proposing and refining regions that contain faces. MTCNN detects effectively faces not initially aligned, and it is widely used due to their fast detection performance. In particular, this method is the one currently integrated on the Evidence Detector software, provided from INCIBE to the *Policia y Guardia Civil Española* (Police and Law Enforcement of Spain). The second method, PyramidBox [8] combines context semantic with hierarchical features from an extended VGG16 [10] architecture and an anchor scale design strategy. This method performs better at detecting small faces. The last method used, DSFD [7] integrates features obtained from a VGG16 architecture with enhanced features to detect faces accurately.



	MTCNN	PyramidBox	DSFD	MTCNN	PyramidBox	DSFD	MTCNN	PyramidBox	DSFD	MTCNN	PyramidBox	DSFD
mAP	0,199	0,534	**0,654**	0,190	0,500	**0,641**	0,171	0,422	**0,573**	0,103	0,236	**0,392**
Speed (sec)	0,104	0,372	0,369	0,066	0,234	0,225	0,038	0,123	0,120	0,019	0,053	0,062
Resized image	**100%**			**75%**			**50%**			**25%**		

Figure 2. Speed and accuracy (mAP) trade-off results for MTCNN, PyramidBox and DSFD face detection methods using four different image resolutions

PyramidBox and DSFD methods were selected due to their accurate performance on complex detection conditions.

III. Experimental Results

The evaluation of the face detectors —MTCNN, DSFD and PyramidBox— was carried out using a subset of 5672 images chosen from the UFDD dataset [9], containing less than four people per scene which is the average maximum number of individuals observed on images of CSA. These images were taken considering several acquisition conditions to evaluate the robustness of the detectors: rain, snow, haze, blur, high/low illumination, lens distortion and distractors, i.e. images without any human faces. Moreover, four image sizes —original resolution, and images resized to 75%, 50% and 25% of the original size— were considered to determine the best trade-off between speed and accuracy in face detection. The accuracy was evaluated using the mean Average Precision (mAP) metric [11], which combines the precision and the recall measures by summarising the shape of the precision-recall curve considering different overlapping thresholds. All the experiments were performed on a GNU/Linux machine box running Ubuntu 16.04, with 32GB RAM, using a 6Gb GTX-1060 NVIDIA Card.

Fig. 2 presents the mAP values and processing times computed for the input images with the four evaluated sizes. As it can be observed, MTCNN is the faster detector, while DSFD is the most accurate for the evaluated image sizes. Furthermore, the use of resized images improves detection speed with a reduction of the mAP values related to the percentage of image resizing. A large resized image percentage leads to a substantial mAP decrease in comparison to the mAP values obtained using original images.

The use of images resized to 75% improved the detection speed more than a 36,37% when compared to the original images, with a slight reduction of mAP values —a maximum decrease of 6,38%. The analysis of images resized to 25% increased the detection speed more than 82,05% in comparison to using original images, but there is a significant reduction of mAP values —a maximum decrease of 55,80%. Finally, the best trade-off between speed and mAP is achieved using images resized to 50%. In this case, the detection speed improved more than 63,44% in comparison to the original images with a maximum decrease of 20,97% for mAP values.

IV. Conclusions

In this work, we proposed an image resizing strategy to speed up three face detector methods —MTCNN, DSFD and PyramidBox— while minimizing the performance drop. The experimental results showed that the use of resized images to 75% and 50% of the original image size allows for a significant improvement in processing time with a small reduction of the mAP values. However, further image resizing decreases face detection performance. All in all, the proposed strategy makes it possible to use complex face detectors such as DSFD and PyramidBox on real-time forensic applications such as CSA detection.

As future work, a dataset with problem domain images will be created to fine-tune face detectors and improve accuracy (mAP) performance.

Acknowledgement

This work was supported by the framework agreement between the Universidad de León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01. Also, this research has been funded with support from the European Commission under the 4NSEEK project with Grant Agreement 821966. This publication reflects the views only of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein.

References

[1] A. Gangwar, E. Fidalgo, E. Alegre, and V. González-Castro, in *ICDP*, 2017, pp. 37–42.

[2] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: Past, present and future," *Comput Vis Image Underst.*, vol. 138, pp. 1–24, 2015.

[3] P. Viola and M. J. Jones, "Robust real-time face detection," *Int J Comput Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[4] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Proc Let*, vol. 23, no. 10, pp. 1499–1503, 2016.

[5] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S^3FD: Single shot scale-invariant face detector," in *ICCV*, 2017, pp. 192–201.

[6] J. Zhang, X. Wu, J. Zhu, and S. C. H. Hoi, "Feature agglomeration networks for single stage face detection," *CoRR*, vol. abs/1712.00721, pp. 1–12, 2017.

[7] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang, "DSFD: dual shot face detector," *CoRR*, vol. abs/1810.10220, pp. 1–10, 2018.

[8] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *ECCV*, 2018, pp. 1–17.

[9] H. Nada, V. A. Sindagi, H. Zhang, and V. M. Patel, "Pushing the limits of unconstrained face detection: a challenge dataset and baseline results," *CoRR*, vol. abs/1804.10275, pp. 1–10, 2018.

[10] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," in *ACPR*, 2015, pp. 730–734.

[11] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserma, "The pascal visual object classes (VOC) challenge," *Int J Comput Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

Localización de Manipulaciones en Imágenes Analizando Artefactos de Interpolación

Edgar González Fernández, Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco, Luis Javier García Villalba*

Grupo de Análisis, Seguridad y Sistemas (GASS),
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
Emails: {edggonza, esarmas}@ucm.es, {asandoval, javiergv}@fdi.ucm.es

***Resumen*—Diversos trabajos han abordado el problema de detección de manipulaciones en imágenes adquiridas desde dispositivos que emplean matrices de filtro de color, comunes en el mercado debido a los bajos costes de producción. Estos dispositivos emplean algoritmos de interpolación cromática durante el proceso de formación de la imagen, lo que permite realizar un análisis estadístico en los elementos generados a partir de este proceso. La mayoría de los trabajos centran sus esfuerzos en analizar la banda verde del filtro de Bayer, ya que contiene más información. La falta de métodos para analizar eficazmente las demás bandas o distintos filtros de color reduce la capacidad de detección de las herramientas conocidas. La finalidad principal de este trabajo es proveer una metodología general para la detección de manipulaciones en este tipo de dispositivos, además de proporcionar nuevas técnicas que permiten generalizar el análisis en una gran diversidad de sensores.**

***Index Terms*—Análisis Forense de Imágenes, Artefactos de Interpolación, Filtro de Bayer, Filtros de Color.**

Tipo de contribución: *Investigación original*

I. Introducción

Con el fin de economizar la producción de dispositivos fotográficos en el mercado, una gran cantidad de ellos utiliza sensores que filtran la información de luminosidad en distintos colores para generar imágenes digitales. Estos sensores capturan de forma parcial la información en cada una de las bandas de color y por medio de algoritmos de interpolación cromática (comúnmente conocidos como *demosaicing* o *demosaicking*) se estima la información necesaria para obtener una imagen a color. Esto genera diversas inconsistencias en los valores estimados en la imagen, como el efecto Moiré, colores falsos, o artefactos en bordes, entre otros.

En este trabajo se propone una metodología general para el análisis de imágenes generadas a partir de una Matriz de Filtro de Color (CFA por sus siglas en inglés), examinando las correlaciones generadas entre los valores interpolados a partir de los píxeles capturados por el sensor. Estas correlaciones se hacen presentes durante el proceso de formación de la imagen, ya que se deben generar los datos necesarios para agregar la información ausente en cada una de las bandas a partir de la información capturada por el sensor, 3 en el caso de los filtros RGB y CYYM, o 4 para el filtro RGBE y CYGM, como se observa en Fig. 1.

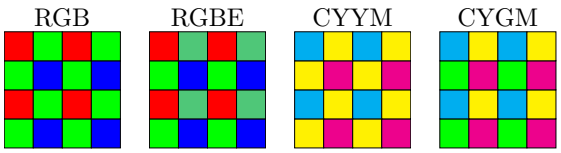


Figura 1: Matrices de Filtro de Color.

Se analizan los artefactos que resultan de la interpolación cromática, que consisten de la discrepancia entre las distribuciones estadísticas de los valores en píxeles *adquiridos*, que son obtenidos originalmente por el dispositivo, e *interpolados*, calculados a partir de la información observada. Una de las características que hacen que esta alteraciones sean observables es la disposición específica con la que estos artefactos aparecen en toda la imagen. Esto es considerado en diversas técnicas existentes en la literatura utilizando distintas propuestas, que serán examinadas y ajustadas para la estrategia que aquí se propone. Se plantearán adicionalmente técnicas que generalicen el análisis de estos artefactos para configuraciones del filtro de color arbitrarias y sus respectivas bandas.

El resto del trabajo está estructurado como sigue: en la Sección II se revisan algunas de las operaciones más comunes para la modificación de imágenes. En la Sección III se abordan brevemente algunos de los trabajos más relevantes relacionados con el objetivo de la investigación. En la Sección IV se analizan con mayor detalle los pasos efectuados en trabajos previos y se establece la metodología a seguir de acuerdo a estas observaciones. Así mismo, se proponen herramientas novedosas para un análisis más flexible, que permite detectar manipulaciones en distintos filtros de color. En la Sección V se presentan los resultados obtenidos en los experimentos usando datasets públicos. Se observará que las técnicas aquí propuestas no se limitan únicamente a artefactos CFA, sino que pueden aplicarse a distintos casos en los que se generen artefactos periódicos durante el proceso de formación de la imagen. Finalmente, en la Sección VI se menciona el trabajo futuro derivado de esta investigación. A continuación se detalla la notación matemática utilizada en este trabajo.

I-A. Notación

A lo largo de este trabajo, se utiliza la siguiente notación. Se denota por $\mathbf{X} = (x_{ij})$ a un elemento del espacio de matrices de tamaño $M \times N$, que usualmente representa una imagen. Se define un *píxel* como la pareja de elementos (i, j) que identifica al elemento x_{ij} de la matriz $\mathbf{X}$, mientras que el valor x_{ij} es su *intensidad*. $\mathcal{A}_g$ denota las posiciones de los píxeles adquiridos en la banda verde, que se define (sin pérdida de generalidad) como $\mathcal{A}_g = \{(i, j) \mid i + j \mod 2 = 1\}$, esto suponiendo una configuración del filtro de Bayer (RGB) como se muestra en Fig. 1. Análogamente, se define el conjunto de píxeles interpolados con $\mathcal{I}_g = \{(i, j) \mid i + j \mod 2 = 0\}$. Los conjuntos $\mathcal{I}_r$ y $\mathcal{A}_r$ para la banda roja, y $\mathcal{I}_b$ y $\mathcal{A}_b$ para la banda azul pueden definirse de forma similar. Dado que la banda verde será analizada más frecuentemente, por simplicidad se denotará por $\mathcal{I}$ y $\mathcal{A}$ a los píxeles interpolados y adquiridos de esta banda. Dada una matriz $\mathbf{X}$ de tamaño $M \times N$ y un conjunto de índices $\mathcal{J} \subseteq [1, M] \times [1, N]$, se denota por $\mathbf{X}(\mathcal{J}) = \{x_{ij} | (i, j) \in \mathcal{J}\}$. La cardinalidad de un conjunto $\mathcal{J}$ se denota por $|\mathcal{J}|$.

II. Manipulaciones comunes

Algunas de las manipulaciones más comunes que pueden ser localizadas utilizando las técnicas aquí propuestas son las siguientes:

- **Filtros de difuminado**. Estos consisten en realizar un suavizado de la imagen utilizando filtros de difuminado o de desenfoque. En las zonas de la imagen en las que algún filtro ha sido aplicado, las diferencias entre el valor de las varianzas de píxeles interpolados y adquiridos disminuye.
- **Copiado-pegado**. Consiste en copiar una región de la imagen y colocarla en una zona distinta dentro de la misma imagen. La región original puede sufrir transformaciones adicionales, como rotaciones, cambios de tamaño o filtros de difuminado para disfrazar la modificación efectuada. Estos cambios suelen destruir el patrón CFA al introducir correlaciones distintas.
- **Enpalme de imágenes**. Consiste en insertar imágenes provenientes de fuentes externas. En este caso, la modificación suele cambiar el patrón CFA por dos motivos:
  - El cambio de posición de la rejilla característica del color verde, lo que puede ocurrir al mezclar imágenes de un mismo dispositivo. Esto puede verse como un caso más general de la alteración de copiado y pegado.
  - Pueden introducirse inconsistencias estadísticas adicionales, ya que al mezclar imágenes de dispositivos distintos las correlaciones entre píxeles que hemos mencionado pueden diferir, por ejemplo al insertar imágenes generadas por computadora o provenientes de dispositivos con diferentes filtros de color.

Mencionamos a continuación como algunos trabajos abordan la detección de estas modificaciones y explicamos brevemente las deficiencias de cada uno.

III. Trabajos previos

En [1], [2] se muestra que al aplicar métodos de interpolación básicos (e.g., bilineal y bicúbica) en la rejilla de la banda verde, los píxeles adquiridos siguen una distribución normal $\mathcal{N}(0, \sigma^2)$, mientras que los píxeles interpolados siguen, aproximadamente, una distribución normal $\mathcal{N}(0, \sigma^2/4)$ (ver Fig. 2). No obstante, aún desconociendo el método de interpolación utilizado es posible identificar dos distintas distribuciones. Este caso será analizado en la Sección IV-B.

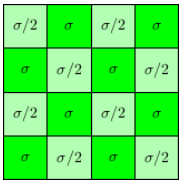


Figura 2: Diferencias de varianza en píxeles interpolados y adquiridos.

Esta diferencia entre distribuciones de los píxeles es utilizada en diversos métodos de detección de manipulaciones buscando la ausencia de los artefactos de interpolación en imágenes provenientes de filtros de Bayer. En [3] se evidencia la manipulación siguiendo los siguientes pasos:

1. Se realiza una estimación del kernel de interpolación de la banda verde mediante el algoritmo de Esperanza-Maximización (EM). Iterativamente se calcula una estimación del kernel de interpolación minimizando el error con el método de Mínimos Cuadrados con Pesos (WLS), considerando los m píxeles vecinos como variables explicativas. Los pesos son calculados como la probabilidad de cada vecino de pertenecer al conjunto de píxeles interpolados.
2. El mapa de probabilidad generado en la última iteración del algoritmo EM se secciona en bloques disjuntos de tamaño $b \times b$. Posteriormente se aplica la Transformada Discreta de Fourier (DFT) a cada bloque y el resultado es comparado con la DFT de un mapa sintético, que representa una clasificación ideal de los píxeles.
3. El bloque se considera como modificado si su medida de similitud contra el mapa sintético es menor a un umbral especificado.

En los experimentos llevados a cabo en [3], se consideran bloques de tamaño 128×128 o 256×256, lo cuál no permitiría una identificación adecuada para modificaciones pequeñas.

Una segunda contribución que puede estudiarse en [4] consiste de manera general de los siguientes pasos:

1. Se aplica el algoritmo de extracción de ruido basado en la Transformada Wavelet Discreta (DWT) [5] aplicado a la banda verde.
2. A partir de la imagen original $\mathbf{X}$ y la imagen sin ruido $\mathbf{X}'$ se obtiene una estimación del ruido del sensor para la banda verde: $\mathbf{R} = \mathbf{X} - \mathbf{X}'$.
3. Considerando el ruido estimado, se realiza un partición en bloques de tamaño $b \times b$, y en cada bloque se mide la diferencia entre las varianzas de los elementos adquiridos e interpolados mediante la siguiente característica:

$$\text{máx}\left(\frac{\text{Var}(\mathbf{R}(\mathcal{I}))}{\text{Var}(\mathbf{R}(\mathcal{A}))}, \frac{\text{Var}(\mathbf{R}(\mathcal{A}))}{\text{Var}(\mathbf{R}(\mathcal{I}))}\right). \quad (1)$$

Si el valor es cercano a 1, entonces el bloque puede considerarse como modificado.

La principal desventaja de estos dos trabajos es que se enfocan en modificaciones en las cuales los artefactos de interpolación desaparecen, como son los filtros de difuminado o el enpalme de imágenes provenientes de fuentes donde los artefactos son inexistentes, (e.g., imágenes generadas por computadora). Sin embargo algunas modificaciones en las que el filtro no es destruido pero si modificado, como es el caso del copiado y pegado o el enpalme de imágenes de la misma fuente no son detectadas.

Estos inconvenientes son resueltos de mejor manera en [2], donde se procede mediante los siguientes pasos:

1. Se calcula una estimación de la imagen a partir de un método de interpolación conocido. Entre los métodos considerados se mencionan las interpolaciones bilineal, bicúbica y basada en el gradiente.
2. Se calcula la matriz de errores, que corresponde a la diferencia entre la imagen original y la estimada. A partir de esta matriz de errores, se calcula una matriz de varianzas locales $\mathbf{\Sigma} = (\sigma_{ij})$, considerando los m vecinos más cercanos de cada (i, j).
3. Finalmente, por cada bloque de tamaño $b \times b$, se calcula la característica:

$$\log\left(\frac{GM_{\mathcal{A}}(\mathbf{\Sigma})}{GM_{\mathcal{I}}(\mathbf{\Sigma})}\right) \tag{2}$$

   donde $GM_{\mathcal{A}}(\mathbf{\Sigma})$ y $GM_{\mathcal{I}}(\mathbf{\Sigma})$ denota la media geométrica de las varianzas locales de los píxeles adquiridos e interpolados respectivamente.
4. Finalmente, se decide que los bloques son originales o modificados aplicando una segmentación basada en Modelos Gaussianos Mixtos (GMM) a la matriz de características. Los parámetros del GMM son estimados mediante el algoritmo EM.

En este método, los bloques con valores cercanos o por debajo de 0 pueden ser considerados como modificados. Sin embargo, la clasificación final depende del algoritmo de segmentación utilizado.

Se considera por último, el método propuesto en [6]. Este se trata de una versión computacionalmente más ligera y efectiva para el análisis de este tipo de modificaciones comparada con [3]. El análisis se lleva a cabo aplicando los siguientes pasos:

1. Se estima el kernel de interpolación utilizando Mínimos Cuadrados Ordinarios (OLS), considerando cada píxel interpolado como variable dependiente y sus vecinos adquiridos como variables explicativas. Con el kernel obtenido se genera una estimación de la imagen.
2. Se obtiene la matriz de errores y se genera el mapa de probabilidad aplicando a cada error la siguiente transformación

$$f(x) = 2\left(1 - \Phi\left(\frac{|x|\sqrt{2}}{\sigma}\right)\right) \tag{3}$$

   donde Φ es la distribución normal estándar y σ es la desviación estándar de los errores correspondientes a píxeles interpolados.
3. La imagen es dividida en bloques de tamaño $b \times b$ y se calcula la transformación DCT en cada bloque. El valor correspondiente al coeficiente de mayor frecuencia se considera como la característica del bloque. Valores cercanos a 0 o negativos se consideran como una modificación.

Finalmente, cabe mencionar que los métodos más precisos para la detección de manipulaciones están basados en la estimación del ruido PRNU del dispositivo del que proviene la imagen, como se procede en [7], [8]. Aunque estos métodos localizan modificaciones de forma precisa, estas técnicas requieren una buena estimación del ruido del sensor. Por tanto, requieren de una colección de imágenes del dispositivo con características específicas para realizar un entrenamiento previo, algo que no siempre es posible conseguir. En este trabajo se propone una técnica que no requiere información adicional a la contenida en la misma imagen.

Observamos que los métodos ya mencionados comparten una metodología basada en una serie de fases, donde cada una puede ser reemplazada por técnicas mejoradas, como estimaciones más precisas del ruido o la extracción de diversas características por bloque. Esto lo analizaremos con mayor detalle en la siguiente sección.

IV. Metodología propuesta

Derivado de la revisión de múltiples propuestas que tratan la identificación de imágenes modificadas mediante el análisis de artefactos de interpolación cromática, presentamos una metodología que consta de 4 fases:

1. **Estimación de la imagen**. Se obtiene una estimación de la imagen en la que se minimicen los artefactos de interpolación.
2. **Cálculo de errores**. Se obtiene una matriz de errores, resultado de la diferencia entre la imagen original y la estimada.
3. **Extracción de características**. Se segmenta la matriz de errores en bloques de un tamaño predefinido para analizarlos individualmente y clasificarlos mediante métodos estadísticos. El tamaño del bloque debe ser seleccionado adecuadamente, ya que el análisis de bloques muy grandes suelen ignorar manipulaciones pequeñas, mientras que bloques demasiado pequeños llevan a resultados estadísticamente inexactos, que imposibilitan una detección correcta.
4. **Segmentación**. Finalmente, con las características extraídas se decide si la imagen ha sido modificada. La identificación de las zonas donde la imagen ha sido modificada depende en gran medida de la eficacia de los 3 pasos previos.

A continuación se detalla cada una de las fases, describiendo las herramientas utilizadas en cada una. Se generalizan algunas de estas herramientas para poder aplicarlas a distintos filtros de color. El análisis puntual se centrará en imágenes obtenidas mediante el filtro de Bayer, debido a su gran popularidad.

IV-A. Estimación de la imagen

El primer paso consiste en estimar la imagen de tal forma que la diferencia entre la imagen original y la estimada exhiba los artefactos de interpolación. El camino seguido en [2] y [3] para llevar a cabo este paso consiste en estimar los coeficientes del modelo de interpolación cromática utilizado por el dispositivo, y a partir de este, generar una nueva imagen.

Aunque en [3] se argumenta que un procedimiento análogo puede ser utilizado para las demás bandas, esto puede ser poco preciso debido a la complejidad de los algoritmos usados en dispositivos reales. Algunos de estos se detallan en [9]. Esto ocasiona que la mayoría de los experimentos y resultados se enfoquen en la banda verde. Simplificando esta tarea, en [6] se considera una estimación por mínimos cuadrados utilizando únicamente los píxeles adquiridos, siguiendo la hipótesis de que la imagen ha sido obtenida mediante un filtro de Bayer. Sin embargo, como se menciona en [4], y como se comprueba experimentalmente, efectuando una interpolación bilineal es posible obtener buenos resultados.

Una opción adicional detallada en [4] consiste en aplicar un algoritmo de extracción de ruido a la imagen analizada, en este caso, mediante la transformada DWT. En general, métodos precisos de extracción de ruido ayudan a detectar estos artefactos ya que el ruido es suprimido por los algoritmos de interpolación aplicados al filtro de Bayer. En este trabajo se ha considerado analizar adicionalmente los resultados obtenidos con el método de extracción de ruido basado en el operador de variación total (TV) presentado en [10].

IV-B. Cálculo de errores

Una vez conseguida la imagen estimada, la diferencia entre ambas imágenes evidencia la diferencia entre las distribuciones de cada clase. En la Fig. 3 se muestra la distribución de los errores de píxeles interpolados y adquiridos aplicando 3 diferentes técnicas en la fase de estimación de la imagen: difuminado bilineal, extracción de ruido mediante el operador TV y extracción de ruido mediante la transformada DWT. En

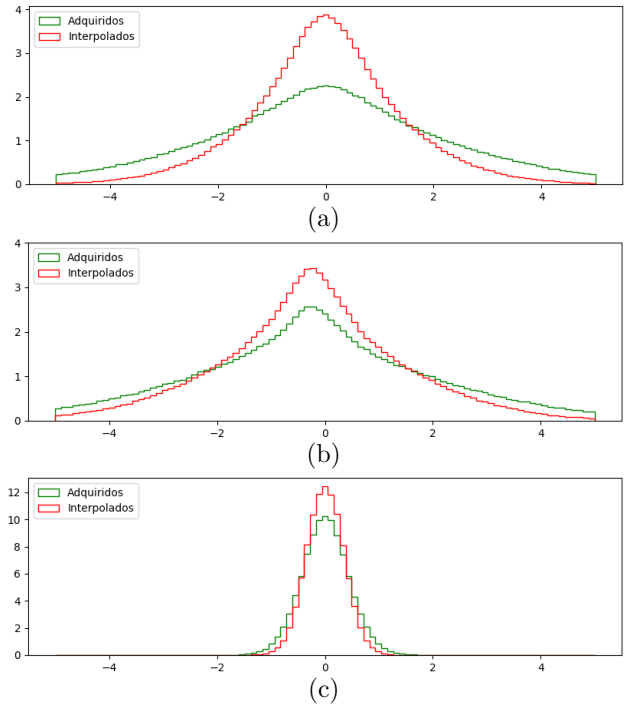


Figura 3: Residuos obtenidos mediante diferentes técnicas de estimación de la imagen: (a) difuminado bilineal, (b) extracción de ruido mediante operador TV y (c) extracción de ruido mediante transformación DWT.

todos los casos se puede apreciar la diferencia en distribuciones de píxeles adquiridos e interpolados, aunque en mayor medida en el caso del difuminado bilineal.

Debido a los buenos resultados obtenidos mediante el difuminado bilineal es que no se han considerado métodos de aproximación del kernel de interpolación, los cuales son computacionalmente costosos con un incremento prácticamente despreciable en la eficacia del análisis. En cuanto a los algoritmos de extracción de ruido, las pruebas experimentales han mostrado un mejor desempeño para el caso del operador TV.

IV-C. Mapas de probabilidad

Siguiendo la idea propuesta en [3], se procede a asignar a cada píxel un nivel de pertenencia al conjunto $\mathcal{I}$. En el caso del mapa definido en [3], se considera que los errores siguen un modelo probabilístico mixto, donde los píxeles interpolados poseen una distribución normal y los adquiridos una distribución uniforme en el rango de valores de intensidad (0 a 255 comúnmente). En [6] se observa que al utilizar la Ec. (3), basada en la función error complementaria, provee mejores resultados al discriminar los píxeles adquiridos más eficazmente.

Para tener la posibilidad de aplicar las ideas ya tratadas a los distintos filtros de color existentes, las propuestas pueden adecuarse a cada caso particular, lo que introduce la restricción de conocer a priori la disposición de los píxeles adquiridos e interpolados. Aunque en muchos casos esto no supone una restricción fuerte, se propone generalizar este proceso para su uso en diversos filtros. Esto puede ser potencialmente útil para modelos que emplean filtros menos conocidos como son RGBW [11] y X-trans [12], de tamaño 4×4 y 6×6, respectivamente. Este proceso se detalla a continuación.

Consideremos una pareja de enteros positivos (c_1, c_2), que representan el tamaño de la matriz de color (o múltiplos de ella). Para cada pareja (u, v) con $1 \leq u \leq c_1$, $1 \leq v \leq c_2$ se consideran los siguientes conjuntos

$$\mathcal{I}_{u,v} = \{(i,j) \mid i \mod c_1 = u, j \mod c_2 = v\}.$$

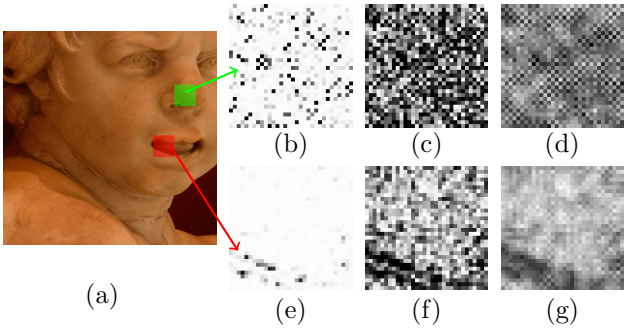
Estos conjuntos representan la selección de un único píxel en el filtro de tamaño $c_1 \times c_2$. Para ejemplificar esto, en el caso del filtro de Bayer tenemos $c_1 = c_2 = 2$. El conjunto de píxeles adquiridos de la banda roja está dado por $\mathcal{I}_{1,1}$, para la banda verde se considera $\mathcal{I}_{2,1} \bigcup \mathcal{I}_{1,2}$ y, finalmente, para la azul $\mathcal{I}_{2,2}$. La estimación de la varianza σ^2 requerida para la generación del mapa de probabilidad estará definida por la Ec. (4).

$$\sigma^2 = \min \{\mathrm{Var}(\mathbf{X}(\mathcal{I}_{ij})) | 1 \leq i \leq c_1, 1 \leq j \leq c_2]\} \qquad (4)$$

Obteniendo la mínima varianza entre las posibles configuraciones se espera obtener una buena estimación de la varianza para los píxeles interpolados.

En este trabajo se aplica un filtro adicional para incrementar la presencia de los artefactos de interpolación en el mapa de probabilidad $\mathbf{P}$. Este filtro consiste en calcular la media de los valores en los m vecinos más cercanos de la misma clase. Esto se define de manera formal en la Ec. (5).

$$\hat{\mathbf{P}}(i,j) = \frac{\sum_{(k,l)\in\mathcal{B}_{ij}} \mathbf{P}(k,l)}{|\mathcal{B}_{ij}|} \qquad (5)$$



(a) (b) (c) (d) (e) (f) (g)

Figura 4: Mapas de probabilidad de las zonas manipulada (en rojo) y original (en verde) de la imagen (a) calculados mediante: (b) y (e) modelo mixto, (c) y (f) función error complementaria $erfc$, (d) y (g) $erfc$ y media de los valores de la misma clase en los m vecinos más cercanos.

donde

$$\mathcal{B}_{ij} = \{(k,l) \mid k = i \mod c_1, l = j \mod c_2, \\ -m \leq k,l \leq m\}.$$

En la Fig. 4 se presentan los resultados conseguidos al aplicar 3 distintos mapas de probabilidad en dos zonas de la imagen, una original y otra modificada. Los mapas de probabilidada corresponden a los definidos en [3], [6] y el propuesto en este trabajo. Se aprecia una mayor discriminación de píxeles adquiridos calculando el mapa mediante la Ec. (5), propuesta en este trabajo, lo que permite identificar manipulaciones de forma más eficaz.

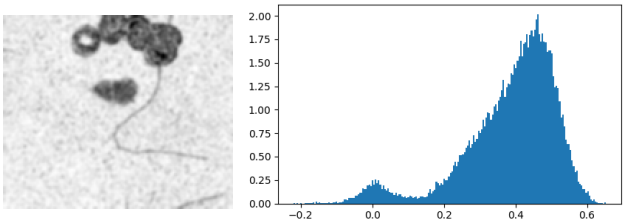
IV-D. Extracción de características

El siguiente paso consiste crear una partición de la imagen en bloques de tamaño $b \times b$ y extraer de cada uno de ellos un valor que será analizado posteriormente mediante métodos de segmentación para determinar las zonas manipuladas. De forma general, se pueden distinguir 2 técnicas empleadas en los trabajos examinados:

1. Comparación de la varianza entre píxeles adquiridos e interpolados. En [4] esto se verifica directamente como el cociente de las varianzas de píxeles adquiridos e interpolados. Una propuesta más refinada se detalla en [2], donde una matriz de varianzas locales es generada píxel a píxel, lo que permite analizar bloques pequeños.
2. Análisis de frecuencias. Estos consisten en verificar que los artefactos ocurren de forma periódica de acuerdo al patrón del filtro de Bayer, como se propone en [3] y [6], donde se utilizan la magnitud de las transformaciones DFT y DCT correspondientemente.

Continuando con el propósito de este trabajo, se propone una nueva característica que puede ser aplicada al mapa de probabilidad sin importar la banda o filtro de color. Adicionalmente, se observa que este enfoque permite analizar directamente la matriz, siempre que en ella se presenten artefactos de manera periódica

El primer paso para la extracción de esta característica consiste en calcular una aproximación del patrón periódico



(a) (b)

Figura 5: Distribución de los valores de la característica propuesta. Es posible ver que alrededor del 0, se acumulan los valores de los bloques con menor correlación contra el patrón calculado de acuerdo al método propuesto.

utilizando bloques de tamaño $b \times b$, como se muestra en la Ec. (6).

$$\mathbf{S}(i,j) = \frac{1}{\lfloor M/b \rfloor \lfloor N/b \rfloor} \sum_{k=0}^{\lfloor M/b \rfloor} \sum_{l=0}^{\lfloor N/b \rfloor} \mathbf{P}(bk+i, bl+j). \quad (6)$$

Entonces el elemento $\mathbf{S}(i,j)$ consta de la media de los elementos en la posición (i,j) en cada bloque de tamaño $b \times b$ extraído del mapa de probabilidad. Mediante este cálculo se espera detectar automáticamente aquellas posiciones en las que los valores mayores aparecen más frecuentemente, por lo que no es necesario conocer a priori la distribución precisa de los píxeles interpolados y adquiridos. Posteriormente, se calcula la correlación del patrón medio con cada uno de los bloques extraídos. Aquellos que tienen una mayor correlación son considerados auténticos, mientras que aquellos con una baja correlación serán etiquetados como alterados. Esta característica queda entonces definida por la Ec. (7).

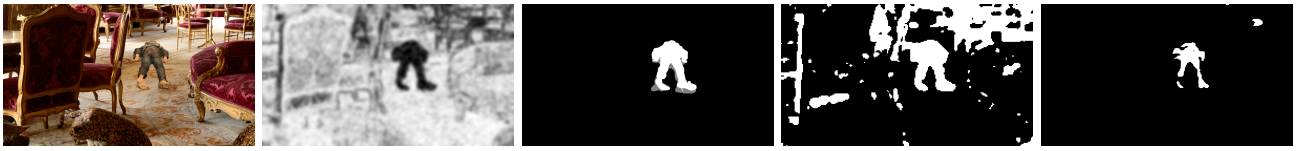
$$C = \text{corr}(\mathbf{P}, \mathbf{S}). \quad (7)$$

Se observa en la Fig. 5 el comportamiento de la característica aplicada a una imagen manipulada utilizando bloques de tamaño 8×8. La presencia de una distribución mixta permite, en la mayoría de los casos, detectar efectivamente la zona de la imagen que ha sido comprometida. Utilizando esta característica hemos conseguido reproducir los resultados obtenidos en [2] y [6], donde se analiza la banda verde. Además de esto, se ha aplicado este método de forma exitosa en las bandas de los espacios de colores RGB y YUV sin modificaciones adicionales, lo que provee a este método de una gran flexibilidad. Estos resultados se detallan en la Sección V.

IV-E. Localización de las modificaciones

Las características extraídas se analizan mediante métodos estadísticos para clasificar los bloques correspondientes como originales y modificados. Dos de las técnicas comúnmente utilizadas para este fin son:

IV-E1. Modelos de Mezcla Gaussiana (GMM): Son modelos de probabilidad en los que la distribución de los elementos de la población total pueden ser explicados mediante el comportamiento de ciertas sub-poblaciones. En el caso de los modelos de mezcla Gaussianos (GMM), se considera que la población consiste de dos subconjuntos que siguen una distribución normal con diversos parámetros. En el caso de



(a) (b) (c) (d) (e)

Figura 6: Segmentación del análisis realizado a una imagen manipulada visible en (a). La matriz de características puede verse en (b). En (c) se visualiza la zona afectada, en (d) la segmentación mediante el método de Otsu y en (e) la segmentación propuesta con $t = ,05$ y $\epsilon = ,05$.

las características extraídas mediante alguno de los métodos definidos en la sección IV-D, es posible distinguir entre dos poblaciones: las características de bloques originales y las de aquellos manipulados. En general, es posible ver que la media en el primer caso, μ_0 es cercana al cero, debido a la baja correlación entre bloques originales y modificados, mientras que el caso de la media para bloques originales es mucho mayor. Los parámetros de ambas distribuciones pueden ser estimados mediante el algoritmo EM [13]. Una vez estimados los parámetros $(\mu_0, \sigma_0$ para los elementos sospechosos, y (μ_1, σ_1) para los elementos originales, es posible proponer el umbral de segmentación como el valor x tal que:

$$\frac{\phi_{\mu_0,\sigma_0}(x)p_0}{\phi_{\mu_0,\sigma_0}(x)p_0 + \phi_{\mu_1,\sigma_1}(x)p_1} = 0{,}5 \tag{8}$$

donde p_0 y p_1 son las probabilidades a priori y se estiman de acuerdo al número de elementos en cada grupo.

IV-E2. Segmentación de Otsu: El método de segmentación Otsu [14] es un algoritmo ampliamente utilizado para la segmentación de imágenes monocromáticas. Su objetivo es encontrar el punto en el cual la segmentación produce dos grupos cuya varianza conjunta es mínima. De manera formal, Para un conjunto de número S, se busca s de tal forma que al definir

$$S_0 = \{x \in S \mid x \leq s\}, \quad S_1 = \{x \in S \mid x > s\}$$
$$p_i = \frac{|S_i|}{|S|}, \quad \sigma_i^2 = Var(S_i)$$

se minimiza el valor de σ^2 definido por la Ec. (9).

$$\sigma^2 = p_0\sigma_0^2 + p_1\sigma_1^2. \tag{9}$$

IV-E3. Método de segmentación propuesto: Finalmente se propone una segmentación que permite una mejor localización de las zonas afectadas cuando se ha utilizado la característica definida por la Ec. (7). Denotando por $\mathbf{C}$ a la matriz de características extraídas, el primer paso consiste en definir un umbral inicial t cercano a 0. Examinando la matriz $\mathbf{C}$ entrada por entrada, se marcan como sospechosos aquellos elementos cuyo valor este por debajo de este umbral. Posteriormente, se incrementa el umbral por un valor pequeño, y para cada elemento marcado como sospechoso, se buscan en sus m vecinos más cercanos elementos cuyo valor que se encuentren debajo del nuevo umbral y se marcan como sospechosos. Este proceso se itera hasta alcanzar estabilidad, esto es, hasta que no hay nuevos bloques agregados, o hasta alcanzar un valor del umbral predefinido.

La idea principal de esta técnica es expandir las zonas que son visiblemente modificadas sin agregar nuevos bloques, los cuales, visualmente no aportan información. Se puede observar una descripción de este procedimiento en el Algoritmo 1.

Algoritmo 1 Segmentación con k-vecinos más cercanos

Entrada: $\mathbf{C}$, t, $\epsilon > 0$ y m
Salida: Segmentación $\mathcal{T}$ de la matriz de características

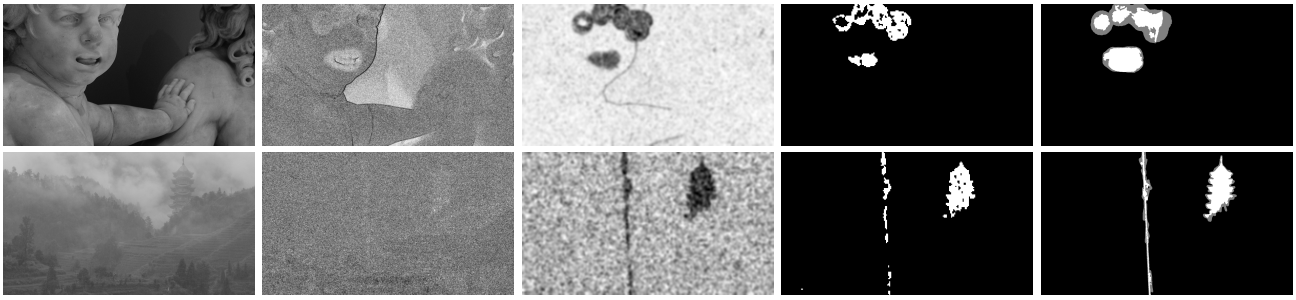
$\mathcal{T} \leftarrow \{(i,j) \mid \mathbf{C}(i,j) < t\}$
$\mathcal{T}' \leftarrow \emptyset$
Mientras $\mathcal{T} \neq \mathcal{T}'$ **hacer**
  $t \leftarrow t + \epsilon$
  $\mathcal{T}' \leftarrow \mathcal{T}$
  Para $(i,j) \in \mathcal{T}$ **hacer**
    Para $(i',j') \in [i-m, i+m] \times [j-m, j+m]$ **hacer**
      Si $(i',j') < t$ **entonces**
        $\mathcal{T} \bigcup (i',j')$
Devolver $\mathcal{T}$

En la Fig. 6 se presenta el proceso de segmentación de una imagen manipulada. En ella se comparan los resultados obtenidos utilizando el método de Otsu y el método propuesto. En la Fig. 6(e) se observa una mayor limpieza en la zona localizada por el algoritmo propuesto. Aunque es posible aplicar iterativamente los métodos de Otsu y de segmentación basada en GMM, la selección del valor adecuado puede ser compleja e introducir ruido al aplicar una segmentación estricta. Por tanto, se propone utilizar el método de segmentación aquí propuesto.

V. Resultados

Para evaluar el método propuesto se han utilizado dos datasets públicos de imágenes manipuladas, los cualesa pueden encontrarse en [7] y [15]. Se denota a estos datasets como D1 y D2 en lo que sigue. Se ha optado por utilizar imágenes disponibles públicamente para evitar la creación de casos generados ad-hoc, lo que puede sesgar los resultados obtenidos. D1 consiste de 4 grupos de 55 imágenes modificadas por cada uno de los siguientes dispositivos: Nikon D7000, Nikon D90, Sony α57 y Canon 60D. Las manipulaciones de este dataset son diversas y no se detallan en la fuente. D2 a su vez cuenta con 4 grupos de 12 de imágenes en las que únicamente se aplican modificaciones de copiado y pegado. Los grupos de D2 corresponden a una red social (Flickr) y 3 dispositivos: Panasonic, Nikon y Canon, aunque no se especifica el modelo.

Las implementaciones han sido hechas con ayuda de librerías dedicadas al análisis númerico y de imágenes disponibles para Python: Numpy, Matplotlib, Skimage y OpenCV. Se midió el desempeño del algoritmo contrastando el área obtenida contra la zona verdaderamente modificada, señaladas



(a) (b) (c) (d) (e)

Figura 7: Resultados del análisis en 2 imágenes de distintos dispositivos detallando los pasos del análisis: en (a) se muestra la imagen manipulada, en (b) el mapa de probabilidad calculado a partir de los errores. En (c) se observa ya la parte modificada exhibida por las características de los bloques. En (d) se tiene el mapa de manipulaciones, que puede contrastarse con la zona realmente manipulada mostrada en (e). En la primer fila vemos el análisis a la imagen mostrada anteriormente. En la segunda fila es posible notar que el análisis puede identificar manipulaciones de manera fina.

mediante máscaras provistas en los mismo datasets. Se definió que una detección es correcta si al menos el 5 % de la zona modificada es detectada adecuadamente y puede distinguirse de manera definida.

Para analizar las modificaciones en D1, se ha empleado lo siguiente:

- La matriz de errores se obtiene utilizando un difuminado bilineal para identificar artefactos de interpolación cromática.
- Se calcula un mapa de probabilidad usando la función de pertenencia definida por la Ec. (5) utilizando $m = 3$ vecinos más cercanos.
- La característica extraída está definida por la Ec. (7) utilizando bloques de tamaño 8×8.
- Finalmente la segmentación se lleva a cabo aplicando un filtro de Gauss en ventanas de tamaño 3×3 y utilizando posteriormente el Algoritmo 1 con $t = ,05$, $\epsilon = ,05$ y $m = 2$.

En los modelos de la marca Nikon hemos analizado las bandas RGB por separado y la banda Y de la transformación YUV, obteniendo resultados similares. El análisis para Sony α57 ha sido mucho más exitoso utilizando las bandas U y V de la transformación YUV. En todos los casos se ha obtenido una gran tasa de acierto para decidir si una imagen ha sido manipulada. En las imágenes pertenecientes a dispositivos Nikon se pudo decidir que 52 de las 55 imágenes han sido manipuladas, mientras que en las pertenecientes a la cámara Sony α57, esto pudo lograrse en 53 de las 55 imágenes. A pesar de los buenos resultados, nos encontramos con amplias diferencias en cuanto al procentaje detectado de la manipulación. Esto puede observarse a detalle en la Tabla I.

En el análisis llevado a cabo en D2 se obtuvieron mejores resultados con las imágenes de los dispositivos Panasonic y Nikon. En este caso las detecciones han sido posibles utilizando las bandas del espacio de color YUV con las siguientes opciones:

- Para la estimación de la imagen se utiliza el método de extracción de ruido basado en el operador TV.
- Se calcula el mapa de probabilidad usando la Ec. (5) con $m = 3$ vecinos.
- Se extrae la característica propuesta en la Ec. (7) al mapa de probabilidad en bloques de tamaño 8×8. Si el análisis falla en todas las bandas del espacio de color YUV entonces se ejecuta el mismo análisis directamente en la matriz de errores.
- La segmentación se lleva a cabo aplicando un filtro Gaussiano considerando una ventana de tamaño 3×3 y utilizando el Algoritmo 1 con $t = ,05, \epsilon = ,05$ y $m = 2$.

Como se observa en la Tabla I, el método propuesto ha sido exitoso en la mayoría de las imágenes provenientes de las cámaras Panasonic y Nikon. En algunos elementos de este conjunto de imágenes hemos observado en cada bloque de tamaño 8×8 la presencia de un píxel cuyo error es mayor al resto. Al realizar una modificación de tipo copiado y pegado, este elemento pierde la posición relativa que se observaba en la zona original, lo que hace detectable la manipulación.

Nuevamente se ha tenido problemas al analizar imágenes de la càmara Canon. En cuanto a las imágenes provenientes de la red social Flickr, la compresión efectuada podría imposibilitar la detección adecuada.

Dataset	**Dispositivo**	**Identificadas**	**% de la modificación detectada**		
			<25 %	**25-50 %**	**>50 %**
D1	Nikon D7000	52/55	12	10	30
D1	Nikon D90	52/55	13	11	28
D1	Sony α57	53/55	4	8	41
D1	Canon 60D	0/55	0	0	0
D2	Panasonic	10/12	0	0	10
D2	Nikon	10/12	0	0	10
D2	Canon	0/12	0	0	0
D2	Flickr	0/12	0	0	0

Tabla I: Detalle del análisis en cada dateset. Se presentan: la fuente de la imagen en la columna 2, la cantidad de imágenes reconocidas como manipuladas en la tercera columna y la cantidad en un rango de porcentaje de detección en las columnas 4 a 6.

VI. Conclusiones y Trabajo Futuro

Se ha propuesto una metodología que permite generalizar la detección de manipulaciones a distintos filtros de color sin perder precisión en los resultados observados en la literatura que tratan el caso RGB. Este método puede ser aplicado en diversos escenarios, siempre que se presenten artefactos de forma periódica en la imagen analizada. Para esto, es posible proponer nuevas herramientas de análisis en cada una de las fases consideradas. Se ha observado que algunos de los problemas para definir la región modificada provienen de la dificultad para definir parámetros adecuados en los mètodos de segmentación, por lo que un análisis más cuidadoso de está fase es necesario.

Como trabajo futuro se plantean los siguientes objetivos: verificar más tipos características que permitan localizar modificaciones para aumentar el alcance de la metodología propuesta, realizar pruebas de las herramientas propuestas en una mayor cantidad de datasets y comparar los resultados con los obtenidos mediante otras técnicas, como aquellas basadas en redes neuronales y métodos de deep learning [16], [17].

Agradecimientos

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326. Website: http://ramses2020.eu

Referencias

[1] A. C. Gallagher and T. Chen, "Image authentication by detecting traces of demosaicing," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage, AK, USA, July 2008.

[2] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, October 2012.

[3] A. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, October 2005.

[4] A. E. Dirik and N. Memon, "Image tamper detection based on demosaicing artifacts," in *Proceedings of the IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009, pp. 1497–1500.

[5] L. Sendur and I. W. Selesnick, "Bivariate shrinkage with local variance estimation," *IEEE Signal Processing Letters*, vol. 9, pp. 438–441, 2002.

[6] E. González Fernández, A. L. Sandoval Orozco, L. J. García Villalba, and J. Hernandez-Castro, "Digital image tamper detection technique based on spectrum analysis of CFA artifacts," *Sensors*, vol. 18, no. 9, p. 2804, 2018.

[7] P. Korus and J. Huang, "Multi-scale analysis strategies in PRNU-based tampering localization," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 809–824, April 2017.

[8] G. Chierchia, S. Parrilli, G. Poggi, L. Verdoliva, and C. Sansone, "Prnu-based detection of small-size image forgeries," in *2011 17th International Conference on Digital Signal Processing (DSP)*, July 2011, pp. 1–6.

[9] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Processing: Image Communication*, vol. 26, pp. 518–533, October 2011.

[10] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imaging Vis.*, vol. 20, no. 1-2, pp. 89–97, Jan. 2004.

[11] M. Tachi, "Image processing device, image processing method, and program pertaining to image correction. U.S. patent 8,314,863," November 2012.

[12] Fujifilm Global. [Online]. Available: https://www.fujifilmusa.com/products/digital_cameras/x/fujifilm_x20/features/page_02.html

[13] S. Tatiraju, "Image segmentation using k-means clustering, em and normalized cuts," 2008.

[14] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.

[15] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012.

[16] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security*, Dec 2016, pp. 1–6.

[17] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection." in *Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security*. IEEE, 2016, pp. 1–6.

Herramienta Automática de Adquisición de Información de Ransomware

Antonio López Vivar, Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco, Luis Javier García Villalba

Grupo de Análisis, Seguridad y Sistemas (GASS),
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
Emails:{alopezvivar, asandoval, javiergv}@fdi.ucm.es, esarmas@ucm.es

***Resumen*—Los ataques de ransomware reportados a las autoridades enfrentan la dificultad técnica de las dependencias de policía local para recopilar la información y ejecutar un análisis forense adecuado. En este trabajo se propone una herramienta de análisis forense que permite adquirir suficiente información para facilitar la posterior clasificación del ransomware. La herramienta propuesta combina la captura de ventana emergente que muestra el ransomware y a través de técnicas de reconocimiento óptico de caracteres, la obtención el mensaje de rescate junto con la dirección de pago y el valor. Además, extrae los ficheros generados por el ransomware y realiza un volcado de la memoria virtual del sistema para el análisis por parte del técnico forense. Para evaluar la precisión de la herramienta, se realizaron experimentos con distintas muestras de ransomware en un ordenador real, bajo un entorno controlado.**

***Index Terms*—Análisis Forense, Bitcoin, Criptomonedas, Internet, Ransomware, Reconocimiento Óptico de Caracteres, Reconocimiento de Patrones, Volcado de Memoria**

I. Introducción

En sus inicios, el uso de Internet era mínimo. Sobre todo era utilizado por sectores industriales, militares e investigación, pero poco a poco se fue popularizando su uso en la sociedad. El continuo desarrollo de la tecnología y su facilidad de adquisición atrae una gran cantidad de consumidores, facilitando la expansión de Internet. Los dispositivos tecnológicos se convirtieron en medios de almacenamiento y comunicación imprescindibles para el uso diario, llegando al punto de tener información privada en dispositivos con acceso a Internet. Esto provocó que un grupo minoritario de usuario de Internet se interesara en la sustracción ilícita de dicha información para obtener un beneficio. Como consecuencia, surgieron aplicaciones maliciosas para atacar a los dispositivos con acceso a Internet.

Entre los distintos tipos de software malicioso que se pueden hallar en Internet, uno de los más peligrosos y en los últimos años muy utilizado por cibercriminales, es el llamado ransomware. Campañas de ataque de este tipo de software malicioso se han visto en diferentes entidades, siendo estas públicas o privadas. En 2016, hubo una campaña dirigida a hospitales como el *Hollywood Presbyterian Medical Center* en los Ángeles, donde un malware de tipo ransomware bloqueó el acceso al sistema hasta que se efectuó el pago de $17000 dólares como "rescate". La portavoz del FBI, Laura Miller, comunicó que se hicieron cargo de las investigaciones. Sin embargo, fuentes policiales explicaron al medio de comunicación *The Time* que el hospital había pagado el rescate antes de solicitar asistencia de la policía [1].

Asimismo, algunos hospitales en Alemania fueron blancos de ataques afectando el funcionamiento de sus sistemas informáticos. Este fue el caso del sistema de rayos X que no pudo acceder a los datos que necesitaba debido a que se encontraban cifrados. Afortunadamente, los hospitales no llegaron a pagar el rescate debido a que recuperaron los datos desde sus copias de seguridad [2]. Otros objetivos destacables fueron grandes e importantes empresas de medios de comunicación como *The New York Times* [3]. También, Universidades de prestigio como la Universidad de Calgary en donde se pagó $16000 dólares para recuperar correos cifrados de una semana [4]. El ataque realizado a las máquinas expendedoras de billetes del tren en San Francisco, permitiendo a las personas viajar sin pagar dichos billetes [5].

La mayoría del software malicioso tiene como objetivo obtener información confidencial de las empresas y de los usuarios de Internet en general. Esto se debe a que el almacenamiento de datos personales en la red es cada vez más utilizado. De ahí la importancia que adquieren los servidores en el funcionamiento de las empresas tecnológicas. Esto las convierte en el blanco principal para los cibercriminales. Un ejemplo es un ataque masivo de ransomware realizado a MongoDB, donde se secuestraron 32.000 servidores, se exigió el pago de un rescate mediante bitcoins para recuperar la información. Varias empresas se vieron afectadas, como Telefónica y eBay, además de gobiernos que utilizan este servicio [6].

Nadie está exento de sufrir un ataque informático y con ataques cibernéticos cada vez más complejos que ocurren todos los días, las técnicas forenses basadas en el análisis de la memoria se están convirtiendo en instrumentos fundamentales en las investigaciones cibercriminales. Los analistas forenses pueden descifrar lo que sucedió en un sistema al adquirir e inspeccionar la información en la memoria virtual. Sin embargo, la base de este análisis puede invalidarse si la adquisición de la memoria ha sido alterada o mal ejecutada.

El resto del trabajo se estructura como sigue: En la Sección II se explica el concepto software malicioso haciendo énfasis en el ransomware. La Sección III presenta los trabajos relacionados con el análisis de malware y su clasificación. La herramienta de clasificación automática de ransomware propuesta se presenta en la Sección IV. la Sección V muestra los resultados obtenidos en los experimentos realizados con la herramienta. Finalmente, la Sección VI recoge las principales conclusiones extraídas del trabajo y el trabajo futuro.

II. Ramsonware y la Evolución del Software Malicioso

El software malicioso es cada vez más sofisticado, sus métodos de propagación también han mejorado, adaptado y son cada vez más diversos e ingeniosos. Aumentando de esta forma el número de usuarios víctimas de este tipo software. Por definición, un malware es un software malicioso que busca infectar un dispositivo o datos sin el conocimiento del usuario, ejecutando acciones no deseadas o dañinas. Este tipo de software está diseñado para sustraer información o inutilizar los dispositivos que infectan [7]. Cualquier aplicación (software) que dañe a un usuario, ordenador o red se puede considerar como malware. En la Figura 1 se exponen los predecesores más importantes del malware moderno desde sus inicios.

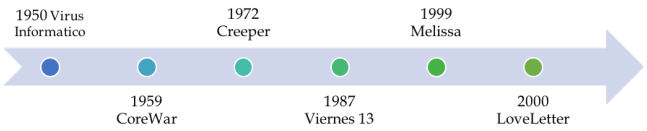


Figura 1. Evolución del malware

John Von Neumann fue el primero que teóricamente postuló el concepto de un virus de ordenador en 1949. Fue el primer concepto de software malicioso y tomó varios años hasta que apareciera la primera instanciación del concepto de Von Neumann [8]. En 1959 se desarrolló un juego en los laboratorios de Bell Computer llamado CoreWar que consistía en dos programas compitiendo por ocupar la memoria del oponente [9]. El programa llamado Creeper, desarrollado en 1971, fue considerado la primera implementación del concepto de Von Neumann y se expandió a través de ARPARNET. Sin embargo, Creeper no era un malware porque no fue diseñado para hacer daño. Creeper puede considerarse el padre de todos los gusanos y virus. Una década después Fred Cohen en 1984 desarrollo el primer malware para obtener acceso a un ordenador [10]. El virus Viernes 13 (1988), también conocido como "Jerusalén". Obtuvo su nombre debido a que se activaba cuando coincidiera en el calendario el viernes y el 13 [11].

Melissa fue unos de los primeros malware en utilizar ingeniería social para propagarse mediante el envío de correos electrónicos con un fichero adjunto con extensión *.doc que contenía el código malicioso. En el 2000, el virus LoveLetter con características parecidas a Melissa, pero con una mayor complejidad, fue catalogado como troyano y gusano.

Como se ha observado, con el paso del tiempo la tecnología evoluciona y el malware se ha ido adaptando a esos cambios. Estas adaptaciones generan variaciones de los mismo y aumentando su alcance. Existen diferentes tipos de malware, pero todos ellos se pueden clasificar en tres grandes grupos: Virus, gusanos y troyanos. Los Virus se replican en los recursos del ordenador, son destructivos, infectan y toman el control de sistemas vulnerables. Por otro lado, los gusanos intentan obtener las direcciones de equipos en la red, la ralentiza y bloquea sus comunicaciones. Finalmente, los troyanos efectúan acciones sin que el usuario se dé cuenta, obtienen o modifican datos y los envían a los criminales.

II-A. Ransomware

El término ransomware viene de la unión de dos palabras "ransom" que significa rescate y "ware" que desciende del término software. Por tanto, hablar de ransomware se refiere a un software malicioso que pide un rescate a cambio de devolver el control ya sea del dispositivo o de los datos que contiene el equipo infectado [12]. Existen muchas versiones, pero la mayoría de ellos ejecutan un cifrado de los datos del dispositivo, negando así el acceso al usuario a ellos y pidiendo un rescate económico para volver a recuperar la información, lo cual pocas veces sucede.

Existen diferentes versiones acerca de las fechas de inicio del ransomware. La forma en que empezó a propagarse, la cantidad de dinero que se exigió para obtener los datos de vuelta y el lugar donde surgió. Glassberg hizo referencia a que ha existido durante muchos años, pero no aportó una fecha exacta [13]. Kharraz, Robertson, Balzaroti, Bilge Kirda expusieron que pudo aparecer a partir de 2004, pero no fue más significativo hasta diez años más tarde [14]. Una de las versiones mas conocidas se remonta a finales de la década de los ochenta, en la que se propagó un malware que reemplazaba el archivo autoexec.bat por un archivo diferente mediante el cual y al cabo de 90 reinicios del ordenador, se informaba al usuario de que había sido infectado por un malware que bloqueaba el acceso a los datos contenidos en el equipo hasta que se pagara un rescate en efectivo de $189 dólares americanos [15].

Durante los años noventa el ransomware paso casi desapercibido hasta que reapareció en 2005 haciendo uso de nuevos y mas potentes esquemas de cifrado. Sin embargo, a partir de 2009 cuando se produce el punto de inflexión para el ransomware con el nacimiento de la criptomoneda Bitcoin. Los cibercriminales encontraron en ella la respuesta al problema del anonimato al momento de cobrar el rescate.

II-B. Características de un Ataque de Ransomware

La Figura 2 muestra las fases del ataque de un ransomware, que a continuación se describen.

- **Distribución**: La manera de propagación más común es a través de un correo electrónico, el cibercriminal se vale de varias técnicas de ingeniería social para conseguir que el usuario confie en el mensaje y así lograr su objetivo: la ejecución del software malicioso.
- **Infección**: Una vez que el *payload* malicioso ha sido entregado al sistema de la víctima, comienza la infección. El código malicioso se instala automáticamente en el sistema, agregando entradas nuevas en los registros del sistema que aseguran su ejecución permanente y automática cada vez que el equipo se reinicie.
- **Command & Control**: En esta etapa, el software malicioso intenta comunicarse con el servidor que lo controla para obtener, en la mayoría de los casos, las claves de cifrado y también las instrucciones a seguir a partir de este momento. La forma en la que se comunica el ransomware con su servidor que lo controla varia de familia en familia y no siempre son las mismas. En algunos casos la comunicación puede llevarse a cabo a través de un canal simple http sin cifrado o pueden

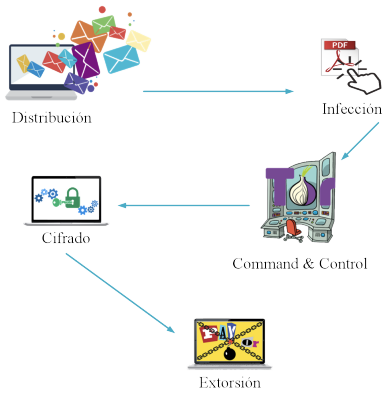


Figura 2. Anatomía de un ataque de ransomware.

utilizar canales mas complejos como el uso de la red TOR para acceder al servidor controlador.

- **Cifrado**: En esta etapa, el ransomware empieza a cifrar los ficheros anteriormente identificados utilizando las instrucciones y las claves enviadas por su servidor controlador.
- **Extorsión**: Una vez que los ficheros han sido cifrados en su totalidad, la siguiente fase es hacerle saber al usuario acerca de esto. Para comunicárselo a la víctima se muestra una ventana en la cual se indican instrucciones a seguir para consignar el pago y poder liberar los datos cifrados. De familia en familia también varía la forma en la que se realiza la extorsión.

En este trabajo se presenta una herramienta que actúa durante la etapa final del ciclo de infección de ransomware con el fin de otorgar una opción rápida y sencilla de adquirir información valiosa para el analista forense.

III. Análisis Forense de Malware

El objetivo del análisis de malware suele ser proporcionar la información necesaria para responder a una intrusión en un equipo o en una red. Para ello, existen dos enfoques fundamentales para el análisis forense de malware: estático y dinámico. En esta sección se explica cada uno de estos enfoques.

III-A. Análisis Estático

El análisis estático clasifica el malware a través de su representación persistente como son los archivos binarios u otros formatos de archivo que pueden contener código malicioso. Estos archivos tienen una estructura de datos que almacenan información acerca del código del malware, como: tipo de aplicación, librerías usadas, funciones importadas, funciones exportadas, fecha de compilación, secciones, consola o *GUI* y recursos utilizados [10]. Los antivirus también pueden utilizar esta técnica ya que hacen un escaneo de memoria sobre todo del disco sólido [16]. El análisis estático se basa en dos tipos de búsquedas: búsqueda de cadena de texto y búsquedas basadas en lo semántico.

- **Búsqueda de cadenas de texto**: Opera sobre binarios sin ninguna abstracción o interpretación. Se basa en la creación de reglas que permite detectar cadenas de texto, secuencias de instrucciones, y otros patrones existentes dentro del archivo malicioso.
- **Búsqueda basada en lo semántico**: Cubre lo abstracto de un binario en concreto, se centra en el análisis del comportamiento de los programas como el control de flujo y gráficas de llamadas al sistema [10].

Las librerías utilizadas por el malware arrojan mucha información de su comportamiento, según sean enlazadas estáticamente, dinámicamente o en tiempo de ejecución. Los enlaces en tiempo de ejecución son comunes en el malware, especialmente cuando están ofuscados o empaquetados. Los analistas se interesan por el enlazado dinámico ya que cuando el programa llama a la función se podrá saber para que se utilizara [16].

III-B. Análisis Dinámico

El análisis dinámico es el segundo enfoque después de haber realizado un análisis estático del software malicioso. Para llevar a cabo este tipo de análisis, es necesario que el malware sea ejecutado y observar su comportamiento para posteriormente clasificarlo. La ejecución de cualquier software malicioso debe ser cuidadosa y dentro de un entorno controlado [16]. Para este tipo de análisis se utilizan entornos aislados como mecanismo de seguridad para ejecutar programas no confiables de forma segura sin temor de propagar o hacer daño al sistema real. Estas herramientas generan informes de actividad en la red, archivos creados, abiertos o eliminados por cada proceso y actividad en los registros del sistema [10]. Los entornos aislados en algunos casos resultan poco efectivos debido a que muchos malware pueden detectar que se están ejecutando en una máquina virtual cambiando su comportamiento y alterando los resultados del análisis [17].

IV. Herramienta Propuesta

En este trabajo se propone una solución que permite obtener además de la memoria del sistema afectado, información detallada del ransomware responsable. Detalles como el mensaje de rescate mostrado al usuario, la dirección de *wallet* de bitcoin a donde se debe dirigir el pago y el valor que se solicita como rescate. Estos datos permiten al analista forense clasificar y posteriormente correlacionar diferentes casos dentro de una misma campaña de ransomware los cuales pueden ser utilizado como evidencia ante una corte para imputar estos ataques a una organización criminal determinada.

Para alcanzar estos objetivos, la herramienta, la cual como herramienta forense se ejecuta una vez el equipo ya ha sido infectado, debe ser capaz de realizar un volcado de memoria completo del sistema afectado y al mismo tiempo realizar una captura de pantalla que le permite obtener la información adicional anteriormente mencionada para su uso en la clasificación y correlación del malware.

IV-A. Volcado de Memoria

En la adquisición de memoria, la memoria del sistema se recopila como una imagen. Esta imagen luego es examinada por el forense. La precisión del análisis de memoria se basa en la correcta adquisición de la memoria del sistema. Para ello se llevo a cabo el análisis de distintas herramientas que permiten una adecuada obtención de la memoria para verificar si sus

Tabla I
EJECUCIÓN DE LAS HERRAMIENTAS DE VOLCADO DE MEMORIA SOBRE MUESTRAS DE RANSOMWARE.

Herramienta \ Muestra	Cerber	Locky	TeslaCrypt	VarianteTesla	Wannacry	Sage	Tiempo (s)
Dumpit	✓	✓	✓	✓	✓	✓	50
RAMCapturer	✓	✓	✓	✓	✓	✓	58
FTK Imager	✓	✓	✓	✓	✓	✓	52
Winpmem	✓	✓	✓	✓	✓	✓	51

características se adecúan a las necesidades de la herramienta propuesta en este trabajo.

Las herramientas evaluadas en este trabajo fueron:

- **Dumpit**:Herramienta de volcado de memoria desarrollada por MoonSols, que se ejecuta de manera sencilla y que extrae los datos que se encuentran en memoria. Es una herramienta que se mantiene en continuo desarrollo, siendo su la última versión Dumpit v3.0. Su ejecución genera dos archivos, uno es el volcado de la memoria y el otro con extensión *.json que contiene información acerca de la arquitectura de la máquina anfitriona. Esta información es necesaria para facilitar que herramientas especializadas en el análisis de memoria puedan conocer la arquitectura del ordenador. Dumpit funciona para la mayoría de las versiones de Windows y se ejecuta por consola de comando teniendo varias opciones para automatiza el proceso sin que el usuario interfiera.
- **RAMCapturer**: Es una pequeña herramienta forense gratuita que permite extraer el contenido de la memoria volátil del ordenador, incluso si está protegido por un sistema activo anti-depuración o anti-volcado. Desarrollado por la empresa Belkasoft. Están disponibles versiones separadas de 32 y 64 bits. Es compatible con todas las versiones y ediciones de Windows, incluidos XP, Vista, Windows 7, 8 y 10, 2003 y 2008 Server.
- **Winpmem**: Es un framework de código abierto que sirve para la extracción de la memoria volátil. Se puede encontrar en GitHub y está escrita en Python, las primeras versiones se ejecutan de manera idónea. Pero la versión más actual genera un archivo de formato AFF4 el cual hace que se comprima los archivos ya que la memoria volátil puede llegar a ser muy grande dependiendo la cantidad memoria que posea el ordenador. Herramientas de análisis de memoria como Volatility no son capaces de examinar volcados con extensión AFF4, al momento de realizar este trabajo.
- **FTK imager**: Es una herramienta de obtención de imágenes de memoria virtual y vista previa de datos utilizada para adquirir información (volcados de memoria) de manera forense mediante la creación de copias sin realizar cambios en el estado de la evidencia original.Es una herramienta ampliamente utilizada tanto para la extracción como para el análisis de memoria, gracias a que presenta un entorno gráfico que facilita su uso para el usuario

A fin de seleccionar la mejor herramienta de volcado de memoria, se ejecutaron varios volcados de memoria sobre un equipo infectado con distintas muestras de ransomware. Estas pruebas fueron llevadas a cabo sobre un ordenador portátil de arquitectura de 32bits, con una memoria RAM de 2GB y ejecutando Windows 7 como sistema operativo base.

Las muestras de ransomware ejecutados en el ordenador fueron: Cerver, Locky, TeslaCrypt, VarianteTesla, Wannacry y Sage, 6 de los mas representativos de estos últimos años.

En la Tabla I se describe en resumen el comportamiento y el tiempo total que le llevó a cada herramienta extraer la memoria virtual y almacenarla en un dispositivo USB conectado al ordenador.

IV-B. Análisis del volcado de memoria

Aunque fuera del alcance de este trabajo, una posible línea de actuación sería la llevada en [18] donde se desarrolló una herramienta basada en el framework Volatility [19] a fin de automatizar el análisis del volcado de memoria. La herramienta carga el fichero de volcado de memoria y busca conexiones de red abiertas en el momento en que se extrajo el volcado de memoria. Esto es muy útil pues muchos malwares necesitan conectarse con su centro de comando y control. De la lista de conexiones, se saca una lista de procesos y con esa lista se extraen todos los ejecutables presentes en el fichero de volcado para analizarlos mediante la herramienta web VirusTotal a fin de comprobar si se trata de un malware. El resultado del análisis se almacena en un archivo de registro.

IV-C. Adquisición y Análisis de Captura de Pantalla

La manera más habitual de comunicar al usuario que ha sido infectado por ransomware, es a través de ventanas emergentes en donde se le alerta que sus datos han sido ”secuestrados”. Por lo tanto, analizar los datos que estas ventanas contienen permiten al analista obtener mas información que ayuden a encaminar la investigación criminal.

Para ello la herramienta propuesta en este trabajo deberá realizar una serie de pasos para obtener una captura de pantalla que contenga únicamente la ventana del ransomware y que además a partir de ella se pueda extraer el mensaje que ahí se muestra. Cada una de las etapas que se ejecutan para la obtención de la captura de pantalla y su posterior extracción de información se observa en la Figura 3.

- **Captura de Pantalla**: En esta primera etapa la herramienta realiza una captura de pantalla completa del ordenador. Incluyendo todas las ventanas abiertas.
- **Recorte**: Posterior a la captura de pantalla, la herramienta busca sobre la captura obtenida, patrones que le permitan determinar el área de la ventana relacionada con un ransomware. De esta forma la imagen obtenida finalmente no contendrá ninguna información que no

Tabla II
RESULTADOS DE EJECUTAR LA HERRAMIENTA.

Funcionalidad / Muestra	Captura	Recorte	OCR	Archivos Adicionales	Volcado	Coincidencias	Tiempo (seg.)
CryptoLocker	Sí	Sí	Sí	Sí	Sí	21	57,09
TeslaCrypt	Sí	Sí	Sí	No	No	15	70,40
Cerber	Sí	Sí	Sí	Sí	Sí	13	46,37
JigSaw	Sí	Sí	Sí	Sí	Sí	10	58,87
Hermez	Sí	Sí	Sí	Sí	Sí	13	82,21
BTCware	Sí	Sí	Sí	Sí	Sí	12	52,16
Saturn	Sí	Sí	Sí	Sí	Sí	6	45,70
Jaff	Sí	Sí	Sí	Sí	Sí	10	44,66
GrandCrab	Sí	Sí	Sí	Sí	Sí	2	58,52
WannaCry	No	No	No	No	No	0	0

fuese la ventana del ransomware, asegurando la privacidad y anonimato del usuario.

- **Extracción de Mensaje**: Una vez que se obtiene la imagen de la ventana del ransomware, se procede a un reconocimiento óptico de caracteres sobre la imagen con el objetivo de extraer todo el texto del mensaje y principalmente los datos de dirección de *wallet* de bitcoin así como el valor a pagar por el rescate. Al término de esta etapa se almacena esta imagen final en el dispositivo USB externo.

Para la fase de captura de pantalla y de recorte se utilizó un algoritmo de visión artificial a fin de obtener los puntos de interés invariantes. Se eligió entre dos candidatos: SIFT y SURF Aún siendo ligeramente más rápido SURF a la hora de detectar dichos puntos de interés, se optó por SIFT, ya que el segundo supera al primero en el número de características detectadas (extrayendo el doble de punto de interés) y por el menor número de falsos positivos de SIFT frente a SURF.

En cuanto a la fase OCR, se optó por usar la librería de Python Pytesseract por su versatilidad frente a otras opciones similares como Texttract o Pyocr.

Pese a la gran variedad de ransomware existente, la gran mayoría de ellos hacen uso de un fichero como medio de comunicación adicional con la víctima. Este fichero contiene un mensaje de alerta, así como las instrucciones a seguir para recuperar la información. Como es objetivo de esta herramienta la obtención de la mayor cantidad de datos e información que se pueda de los ordenadores infectados, se estableció la funcionalidad de búsqueda de dichos ficheros que contienen esa información, para su futuro análisis. Esto se llevará a cabo mediante el recorrido recursivo de los directorios, en búsqueda de ficheros con extensiones específicas como *.txt y *.html, este último es común en las versiones más recientes de las distintas familias de ransomware.

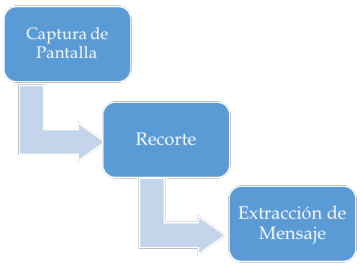


Figura 3. Extracción de información de la ventana del ransomware.

La búsqueda de archivos relevantes, así como el volcado de memoria son utilidades que se realizarán siempre.

El resultado final es una herramienta capaz de realizar un volcado de memoria, hacer una captura de pantalla y analizarla para extraer la ventana del ransomware y de ella obtener la información del mensaje y otros detalles relevantes. Además, se añadió la funcionalidad de buscar en el ordenador analizado, ficheros que contengan mas detalles o instrucciones que muchas de las familias de ransomware suelen dejar una vez que han ingresado al sistema.

V. EXPERIMENTOS Y RESULTADOS

En estos experimentos se utilizó la herramienta con todas las funcionalidades automatizadas para la extracción de información relevante de ransomware, así como el volcado de memoria y la búsqueda de archivos relacionados con el ataque dentro del sistema.

La máquina utilizada tiene las siguientes características: Procesador Intel Pentium de Arquitectura 64bits, Memoria RAM de 4GB y Sistema Operativo Windows 7. También, con la intención de mantener un entorno real se instalaron diversas aplicaciones de uso de cotidiano para un usuario tales como: Microsoft Office, Adobe Reader, además de tener una gran variedad de archivos de texto, imágenes y vídeos con distintos formatos almacenados en las distintas carpetas del ordenador.

En cuanto a la unidad USB que contendrá la herramienta y almacenará el volcado de memoria, tiene capacidad de 32GB con formato NTFS. Los experimentos fueron realizados utilizando diez muestras de ransomware. En la Tabla II se puede apreciar las funcionalidades que la herramienta consiguió llevar a cabo dentro de la maquina infectada.

La herramienta funcionó bien en términos generales, exceptuando el caso de TeslaCrypt donde sólo se consiguió llevar a cabo 3 funcionalidades de las 6 implementadas y WannaCry en donde no se pudo ejecutar la herramienta por completo.

En el caso de TeslaCrypt, el reconocimiento de patrones dentro de la imagen no fue capaz de identificar la ventana como perteneciente a ransomware, esto se debe al nuevo formato de este tipo de ventana, y la falta de patrones para

poder identificarlo. Al no encontrar suficientes coincidencias en la imagen, el reconocimiento óptico de caracteres se descarta. El volcado de memoria en este experimento resulto fallido, debido a que la muestra bloqueaba la ejecución de la herramienta encargada de realizar dicha función.

En el caso de la muestra WannaCry, la herramienta no pudo ejecutarse. Por su configuración, WannaCry cifró totalmente el contenido de la unidad externa que contenía la herramienta y todo lo que genera y extrae. Este incidente abre una linea de trabajo futuro que en el apartado correspondiente se abordará.

VI. Conclusiones y Trabajo Futuro

Hoy en día el análisis de ransomware ha tomado gran relevancia debido al crecimiento y diversidad de este tipo de malware, además de la necesidad de comprender y analizar su comportamiento.

La herramienta propuesta en este trabajo permite obtener información relevante de un ransomware, dentro de un entorno real sin la necesidad de realizar configuraciones complejas, automatizando funcionalidades estudiadas en diversas fuentes como puede ser el volcado de memoria.

La mayoría de los trabajos investigados tratan el malware en un entorno controlado, lo que puede alterar su comportamiento real. En muchos de los estudios la creación de los entornos se lleva a cabo mediante el uso de herramientas que necesitan una configuración específica en cada caso. Limitando estás tareas a un grupo de usuarios con conocimientos suficientes para llevarlo a cabo. En este trabajo se realizó la búsqueda de patrones en imágenes con el objetivo de obtener información suficiente para determinar si se trata de un ransomware. En caso de que la información sea suficiente se realiza el reconocimiento óptico de caracteres y la extracción en un texto plano, con la finalidad de facilitar su manipulación. En cualquier caso, se hará el volcado de memoria y búsqueda de archivos relevantes. Todas estas funcionalidades se ejecutan desde una memoria USB donde también se almacena la información obtenida. (En ningún caso la herramienta se queda residente en la memoria). Para observar el comportamiento de la herramienta se realizaron experimentos con diez muestras distintas de ransomware en un entorno real. Donde se llegó a la conclusión que la herramienta tiene un comportamiento óptimo, en los distintos entornos que fue ejecutada obteniendo los resultados esperados. También se observó que el tiempo de ejecución de la herramienta aumenta dependiendo la velocidad del USB y del tamaño de la memoria RAM. Por último, como líneas de trabajo futuro podrían citarse las siguientes:

- Ampliación de las capacidades OCR de la herramienta.
- Mejora en la detección de la ventana del mensaje del ransomware así como en su procesamiento y recorte.
- Añadir una base de datos con más muestras para incrementar el reconocimiento de patrones.
- Implementar el almacenamiento online de los resultados a fin de evitar la dependencia del USB y conseguir proteger la herramienta frente a ransomware como WannaCry.
- Explorar la posibilidad de una versión de la herramienta que funcione en otros sistemas operativos, como por ejemplo, Linux.

Agradecimientos

Este trabajo no pudo haberse realizado sin la colaboración y desarrollo de Lenin Benavides Quintana y Carlos Roa Medina, estudiantes de grado de la Facultad de Informática de la Universidad Complutense de Madrid.

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 700326. Website: http://ramses2020.eu

Referencias

[1] R. Winton, "Hollywood hospital pays $17,000 in bitcoin to hackers; FBI investigating," 2016. [Online]. Available: http://www.latimes.com/business/technology/la-me-ln-hollywood-hospital-bitcoin-20160217-story.html

[2] R. Millman, "Ransomware holds data hostage in two German hospitals." [Online]. Available: https://www.scmagazineuk.com/ransomware-holds-data-hostage-in-two-german-hospitals/article/530494/

[3] Ediciones El País, "Lista de empresas afectadas por el ciberataque," 2016. [Online]. Available: https://elpais.com/internacional/2016/10/21/actualidad/1477081741_222586.html

[4] A. Ivanov, E. David, S. Fedor, and S. Pontiroli, "Kaspersky Security Bulletin 2016 Story of the Year: The Ransomware Revolution," Kaspersky Labs, Tech. Rep., 2016.

[5] S. Gibbs, "Ransomware attack on San Francisco public transit gives everyone a free ride — Technology — The Guardian," 2016. [Online]. Available: https://www.theguardian.com/technology/2016/nov/28/passengers-free-ride-san-francisco-muni-ransomeware

[6] A. Martinez, "Un enorme ataque de «ransomware» secuestra 32.000 servidores de MongoDB," 2017. [Online]. Available: https://tinyurl.com/hs6flb5

[7] Avast, "¿Qué es el malware y cómo eliminarlo? — Antimalware." [Online]. Available: https://www.avast.com/es-es/c-malware

[8] Panda Security, "Virus y Antivirus — Informacion — Historia — Evolución-Información sobre Seguridad-Panda Security." [Online]. Available: http://www.pandasecurity.com/spain/homeusers/security-info/classic-malware

[9] M. J. Erquiaga, "Botnets: Mecanismos de control y de propagacion," in *Actas del XVII Congreso Argentino de Ciencias de la Computación*, La Plata, Argentina, October 2011.

[10] T. Wüchner, "Behavior-based Malware Detection with Quantitative Data Flow Analysis," Technical University of Munich, Bachelor Thesis, 2016.

[11] Panda Security, "Los virus más famosos de la historia: Viernes 13 - Panda Security Mediacenter." [Online]. Available: http://www.pandasecurity.com/spain/mediacenter/malware/virus-viernes-13

[12] F. d. B. Nafría Oñate, "Plataformas de Ejercicios de Ciberseguridad," Universidad Politécnica de Madrid, Bachelor Thesis, 2016.

[13] J. Glassberg, "Defending Against the Ransomware Threat." [Online]. Available: http://www.elp.com/articles/powergrid_international/print/volume-21/issue-8/features/defending-against-the-ransom-war-threat.html

[14] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the gordian knot: A look under the hood of ransomware attacks," in *Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Milano, Italy, July 2015, pp. 3–24.

[15] H. Salvi and R. Kerkar, "Ransomware: A cyber extortion," *Asian Journal for Convergence in Technology*, vol. 2, 2016.

[16] M. Sikorski and A. Honig, *PRACTICAL MALWARE ANALYSIS The Hands-On Guide to Dissecting Malicious Software*, 1st ed. No Starch Press, Inc, March 2012.

[17] C. Valeriu Liţă, D. Cosovan, and D. Gavriluţ, "Anti-emulation trends in modern packers: a survey on the evolution of anti-emulation techniques in upa packers," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 2, pp. 107–126, 2018.

[18] P. H. Rughani, "Formality: Automated forensic malware analysis using volatility," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, 2017.

[19] M. S. Webb, "Evaluating tool based automated malware analysis through persistence mechanism detection," Ph.D. dissertation, Kansas State University, 2018.

Guidelines Towards Secure SSL Pinning in Mobile Applications

F.J. Ramírez-López, A. J. Varela-Vaca, J. Ropero, A. Carrasco
Universidad de Sevilla, Spain
{framirez4, ajvarela, jropero, acarrasco}@us.es

***Resumen*—Security is a major concern in web applications for so long, but it is only recently that the use of mobile applications has reached the level of web services. This way, we are taking OWASP Top 10 Mobile as our starting point to secure mobile applications. Insecure communication is one of the most important topics to be considered. In fact, many mobile applications do not even implement SSL/TLS validations or may have SSL/TLS vulnerabilities. This paper explains how an application can be fortified using secure SSL pinning, and offers a three-step process as an improvement of OWASP Mobile recommendations to avoid SSL pinning bypassing. Therefore, following the process described in this paper, mobile application developers may establish a secure SSL/TLS communication.**

***Index Terms*—SSL pinning, security, mobile applications, certificate, OWASP**

Tipo de contribución: *Investigación original*

I. Introducción

Nowadays, the use of mobile devices is constantly increasing to do the same operations that used to be done using web services less than a decade ago [1], [2]. However, it is necessary to provide the same security solutions in both environments since both operations are equally critical.

Every day, we read cases of users who have been scammed through the use of mobile applications [3]. For example, users may download some modified version of an application that is not controlled by the owner. In some cases, access to sensitive information from other users of the application has also been detected. This is due to the fact that many of the controls that have been applied in the web environment have not been considered in the mobile environment. Moreover, several mobile applications do not even implement SSL/TLS validations [4].

With this aim, the OWASP Mobile Application Security Verification Standard (MASVS) is an attempt to standardize these requirements using verification levels that fit different threat scenarios [5]. One of the most important challenges in mobile application security is to protect data flows over the insecure communication channel [6]. Insecure communications includes poor handshaking, incorrect SSL versions, weak negotiation or cleartext communication of Personally Identifiable Information (PII) [7]. Even when using SSL/TLS, applications may have vulnerabilities, especially to Man-in the-middle (MiTM) attacks [8]. Security measures such as SSL pinning are desirable [9]. Nevertheless, it is also possible to circumvent SSL/TLS validations [4]. In this paper, we explain how an application can be fortified taking into account certain security controls, where we should shield certain points to avoid attacks. This paper offers an improvement of OWASP mobile recommendations offering a three-step set of controls to avoid SSL pinning problems, and also wants to be a good practice guide for all the mobile application developers and users.

The paper is organized as follows. Section II deals with OWASP mobile recommendations. Section III introduces SSL validations and their possible vulnerabilities. Section IV offers solutions to SSL pinning bypass and shows a case of use. Finally, Section IV presents the conclusions of the paper.

II. OWASP Mobile recommendations

OWASP is the worldwide organization responsible for generating a standard for security in web applications [10]. This way, we can find several sources of information and methodologies in the OWASP documentation. The best-known methodology is the so-called Top 10, where the most frequent vulnerabilities are shown. OWASP group develop Top 10 security risks for web, mobile, and IoT software [11]. Based on our experience, we choose OWASP Top 10 Mobile as our starting point. Table I shows OWASP Mobile Top 10 in December 2016, which is the last update [7].

Tabla I
OWASP Top 10.

Category	Name
M1	Improper Platform Usage
M2	Insecure Data Storage
M3	Insecure Communication
M4	Insecure Authentication
M5	Insufficient Cryptography
M6	Insecure Authorization
M7	Client Code Quality
M8	Code Tampering
M9	Reverse Engineering
M10	Extraneous Functionality

As shown, improper platform usage is considered the most relevant security risk. This category covers the security control that is part of the mobile operating system. However, insecure communication ranks #3 in OWASP Top 10, so it is also quite an important topic to be considered. SSL pinning is included in this category.

Although there are some other methodologies or lists of controls where applications may be reviewed, we are focusing on MASVS, which is defined in the OWASP Testing Guide. The OWASP Mobile Testing Guide has published recently its first version [5]. In this guide, security controls are defined and can be reviewed according to different categories. Every control describes the control itself, shows how can it be tested, and it sometimes offers a solution to the problem. However, the solution must be adapted to the system or the client that we are auditing.

Within the OWASP controls, there are several control layers that must be considered. The utilization of these layers depends on the application. There are three existing layers, called verification levels: L1, Standard Security; L2, Defense-in-Depth; and R, Resiliency Against Reverse Engineering and Tampering.

L1 layer controls - Standard Security. This control layer groups the most basic controls. These controls constitute a set of minimum characteristics that any application should accomplish. With these controls, a certain number of attacks on the application are avoided. This fact may be sufficient for some types of applications.

L2 layer controls - Defense-in-Depth. These controls are more advanced controls than the ones in the L1 layer. They help to avoid complex attacks on our application. This level is more demanding in terms of security since the controls ask for a more mature level of security in the application.

R layer controls - Resiliency Against Reverse Engineering and Tampering. This layer focuses on the reverse engineering attacks that can be done on an application. Therefore, it deals with everything that refers to both the code of an application and what can be modified within the source code. This constitutes an important level of verification of the source, hardware and other components in the application.

Depending on the type of application, several controls are used, while the others are discarded. For example, if an application only shows some information and no registration is needed, there is no point in applying L2 plus reverse engineering controls since there is not any sensitive information. The verification levels that applications may accomplish are the following:

- **MASVS-L1**. It constitutes the most basic security level, as there is no impact on the application development cost. All mobile apps must follow these requirements.
- **MASVS-L2**. E-Health and E-commerce applications, as they store sensitive PII.
- **MASVS-L1+R**. Applications with IP protections. Gaming industry.
- **MASVS-L2+R**. Applications managing critical data, like e-banking applications.

All the controls are grouped into categories, which are shown in Table II. In practice, category V1 is usually excluded, because those controls can be applied only in white-box tests or if we participate in the development of the application. As we can see, category V8 corresponds to level R.

Tabla II
CONTROL CATEGORIES.

Category	Name
V1	Architecture, Design and Threat Modelling Requirements
V2	Data Storage and Privacy Requirements
V3	Cryptography Requirements
V4	Authentication and Session Management Requirements
V5	Network Communication Requirements
V6	Platform Interaction Requirements
V7	Code Quality and Build Setting Requirements
V8	Resiliency Against Reverse Engineering Requirements

Within each category, we can distinguish which controls are applied at each level. We are focusing on category V5, Network Communication Requirements. To secure network communication, we should follow the recommendations shown in Table III.

Tabla III
NETWORK COMMUNICATION SECURITY VERIFICATION REQUIREMENTS.

Control	Description
5.1	Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app
5.2	The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards
5.3	The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted
5.4	The app uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish the connection with endpoints that offer a different certificate or key, even if signed by a trusted CA
5.5	The app does not rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery

In this paper, we show that it is only necessary to achieve the requirements corresponding to control 5.4. In practice, we can identify this control with SSL pinning.

III. WHY ARE SSL/TLS COMMUNICATIONS INSECURE?

Secure Socket Layer (SSL) [12] protocol and Transport Layer Security (TLS) [13] protocol, (hereinafter SSL/TLS) are widely used to provide confidentiality, authentication, and integrity in data communications. SSL/TLS provides three main security services: confidentiality, by encrypting data; message integrity, by using a message authentication code (MAC); and authentication, through digital signatures.

SSL/TLS allows the authentication of both parties, server authentication with an unauthenticated client, and total anonymity. The authentication of client and server may be carried out through digital signatures. Nowadays, digital signatures are mostly based on certificates (i.e., X.509 standard) or shared keys. In the case of using certificates, they always have to be verified to ensure proper signing by a trusted Certificate Authority (CA). On the other hand, these protocols also provide anonymous authentication by using Diffie-Hellman for key exchange from SSLv3.0, TLSv1.0 and later versions.

SSL/TSL protocol is based on a handshake sequence whose main features [14] are used by client and server, as follows: (1) Negotiate the Cipher Suite to be used during data transfer, and exchange random numbers (master key); (2) Establish and share a Session ID between client and server; (3) Authenticate the server to the client; (4) Authenticate the client to the server.

There are several providers widely used as JSSE (Java Security Socket Extension) [15], OpenSSL [16], LibreSSL [17], or GnuTLS [18]. Even there exist specific hardware with built-in SSL/TLS solutions such as iOS devices.

III-A. SSL/TLS vulnerabilities: Bypassing SSL/TLS

As mentioned before, one of the top-3 risks identified by OWASP is an insecure communication due to a poor configuration of an SSL/TLS channel. However, SSL/TLS is a non-free vulnerability protocol since it can be broken by MiTM attacks [11]. MiTM attacks take place due to lack of validation or incorrect validation in the protocol.

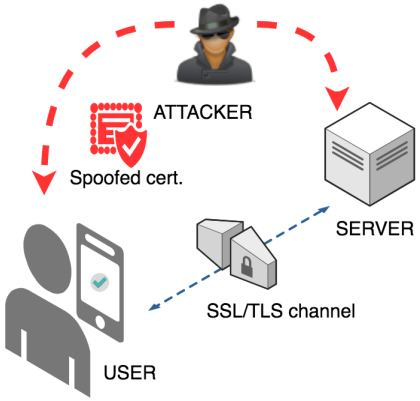


Figura 1. Bypassing SSL/TLS by using spoofed certificates.

In SSL/TLS, certificates are verified to check whether they are signed by proper CA. In this case, we can mislead the application giving a certificate (cf. spoofed certificated, see Fig. 1). The certificate is trusted, though its origin is unknown. Once the certificates are accepted and the handshake is finished, the SSL/TLS communication is established as secure. Meanwhile, a third party is bypassing the channel intercepting and decrypting all the packets in the communication.

III-B. Solution to Bypassing SSL/TLS: SSL pinning

The pinning technique or HTTP Public Key Pinning (HPKP) [4] has emerged in the last years as a security control to fortified HTTPS-based applications against MiTM attacks.

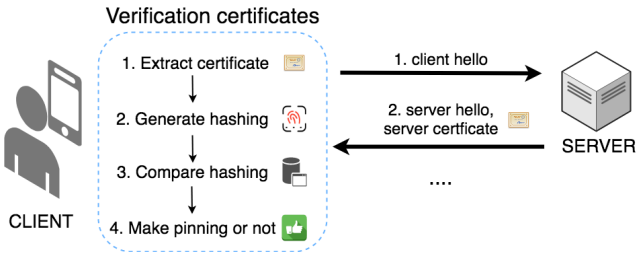


Figura 2. Certificate verification process and pinning.

Fig. 2 shows the SSL Pinning implementation process, which is divided into two stages. In the first stage, the mobile device must initiate communication with the server. The server responds whether it is active or not (cf. server hello in Fig. 2). Then, the client asks for the server's certification when server answers with the content of the information of its certificate and public key (cf. verification certificates in Fig. 2). The second stage is called pinning. The mobile device follows a verification process where the certificate is received from the server. Besides, the public key has to match the one that is stored. If so, the client opens a negotiation or sends packages signed with that public key. When the client does not coincide, it cuts off the communication. Thus, it does not send anything to the server.

III-C. Vulnerabilities of SSL Pinning: Bypassing SSL pinning

SSL pinning is also vulnerable when it is not well implemented. There are several ways to bypassing it, as described by D'Orazio and Choo [4] or by Andzakovic [19]. Several tools can be used to bypassing SSL pinning, such as follows:

- *SSL Kill Switch 2* takes advantage of the fact that the code that implements SSL Pinning is a known code. Application developers use a well-known or common template. In this case, an attacker may guess this and use SSL Kill Switch 2 to bypass SSL Pinning in the application.
- *Dynamic analysis of code* can be applied. For example, Frida or Cycript enable the modification of some functions of the application in runtime.
- If the application does not implement anti-tampering or exceptions for the modification of the application, SSL Pinning functions can be replaced to bypass the pinning process.

IV. Good practices to implement Secure SSL Pinning

Here, we present some good practices or guidelines as a set of several steps to implement an adequate secure solution to the SSL Pinning. This way, bypassing SSL Pinning is avoided. Although, OWASP propose to use a set of controls in order to ensure channels of communications, our guideline demonstrates and ensures that with only three steps the mobile applications can be fortified against bypassing SSL pin-ning and no more control need to be checked.

The proposed process is shown Fig. 3 and indicates the points that have to be tackled to solve the problems mentioned in the previous section. All the measures that should be taken are described below.

Figura 3. Process to ensure SSL Pinning.

1. **Deploy Anti-tampering** solution is an important option since any attacker may decompile the application and recompile it. Modifying some parts of our application, as previously explained in the SSL pinning bypass process, an attacker could skip the SSL pinning, and thus make our application invalid.
   Listing 1 gives an example of the code that can be included into an Android application, particularly in the *onCreate* function within *MainActivity*. so that nothing else but starting check the signature of the application. In iOS, the mechanism is similar, as indicated in the Apple security transforms programming guide [20].

```
for (Signature signature : packageInfo.signatures) {
  byte[] signatureBytes = signature.toByteArray();
  MessageDigest md =
       MessageDigest.getInstance("SHA");
  md.update(signature.toByteArray());
  final String currentSignature =
       Base64.encodeToSpring(md.digest(),
    Base64.DEFAULT);
  Log.d("REMOVE\ME","Include this string as a value
       for SIGNATURE:" +
    currentSignature);
  //compare signatures
  if (SIGNATURE.equals (currentSignature)){
```

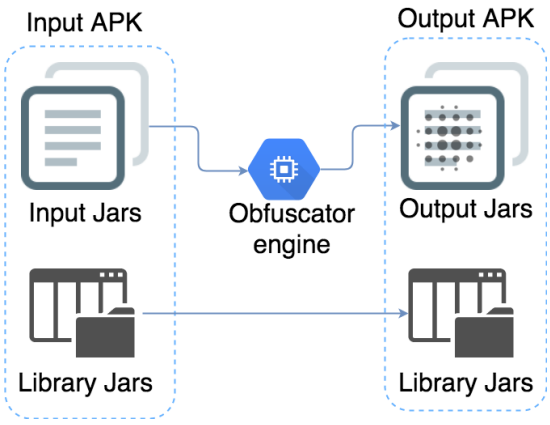


Figura 4. Obfuscation process.

```
    return VALID;
    };
}
return INVALID;
}
...
```

Listing 1. Example of code to check signatures.

2. **Implement Debug Detection**, to prevent our code from being controlled. If we detect that our device is in debug mode, the execution of the application is stopped. This measure stops an attacker from seeing our code step-by-step behavior. Together with next measure, obfuscation of code, allows hiding the internal functioning of our application. We may use functions as the one we are providing in listing 2 for our Android application.

```
protected void OnCreate (Bundle bundle) {
  if (sg.vantagepoint.a.c.a() ||
      sg.vantagepoint.a.c.b()
    || sg.vantagepoint.a.c.c()){
    this.a ("Root detected!"); //This is the message
        we are looking for
  }

  if (sg.antagepoint.a.b.a((Context)this.
          getApplicationContext(())) {
    this.a("App is debuggable!");
  }
  super.onCreate(bundle);
  this.setContentView(2130903040);
}
```

Listing 2. Example of code to avoid debugging mode.

3. **Obfuscate code**. All the measures above do not make sense without prevent-ing any attacker from knowing our code and making the analysis of SSL Pinning functions or any of the previous ones more difficult. For this reason, code obfuscation is necessary. There are code obfuscators which convert our existing code into a more illegible code (cf. Fig. 4). Therefore, it is more difficult to detect which are the critical functions of the code for an attacker.
As may be seen, replacing variables and function names for letters and numbers, makes it more difficult to guess what a function does. An example of obfuscation is shown in listing 3, where names of variables and functions are hidden.

```
public void a(B c){
switch (C.a()){
case R.a.b:
B d = (B) c.a();
A f = (A) d.c(R.a.j);
A g = (A) d.c(R.a.k);
D h = (D) d.c(R.a.m);
String i = d.b.c ();
String j = L.a(f.b.c());
if(!i.equals(String.a)){
  E.c(0);
} else if (secret.equals(String.f)) {
  d.setTextColor(c.getResources().
    getColor(R.color.color_nebula));
  f.setTextColor(c.getResources().
    getColor(R.color.color_nebula));
  ((Vibrator) parent.getContext().
    getSystemService("vibrator")).
    vibrate(400);
D.makeText(c.getContext(),String.m,1).a();
D.makeText(c.getContext(),String.n,1).a();
} else {
}
```

Listing 3. Example of obfuscation code.

There are numerous tools on Android and iOS that allow the obfuscation of code, as *Proguard* [21] for Android or *iXGuard* [22] for iOS.

IV-A. Case of use: securing a mobile application

The analysis carried out in [4], where 40 mobile applications from different environments were analyzed, showed that only 10 of the applications used SSL pinning. Moreover, all these applications are vulnerable when tampering with application at runtime, or when modifying the application executable. Next, we offer a solution that fixes all these vulnerabilities, but focusing on an Android mobile application case study.

Samsung Galaxy J5 device and Android 8.0 Oreo OS have been used to carry out the tests detailed in this section. Regarding the application, we have customized an Android-based template [16], but all the results are also applicable to iOS systems. Many functions are given in the GitHub AeroGear library [23]. The template is given by default with a set of tests to check some security controls, such as root detection, device lock, etc. These controls are related to the ones proposed by OWASP mobile recommendations. However, other security controls such as anti-tampering are not included. In order to illustrate the application and the effectiveness of our guideline, the three-step process is detailed below:

Step 1. Detection of the debug mode in the application. We must verify some controls of our device, as the detection of debug mode, hooking tools and emulation mode. Debug and emulation mode can be detected by means of the code shown in listing 4.

```
public void debuggerDetected() {
  totalTests++;
  SecurityCheckResult result = securitySer-vuce.check(
      SecurityCheckType.IS_DEBUGGER);
  if (result.passed()) {
    setDetected(debuggerAccess,
      R.string.debugger_detected_positive);
  }
}

public void detectEmulator() {
totalTests++;
  SecurityCheckResult result = securitySer-vuce.check(
      SecurityCheckType.IS_EMULATOR);
  if (result.passed()) {
    setDetected(emulatorAccess,
      R.string.emulator_detected_positive);
  }
```

```
}
```

Listing 4. Example of code to avoid debugging mode.

The function *detectEmulator* call another function inside it, named *securityService.check*. This function depends on the AeroGear library, where we can find the function shown in listing 5. This function checks if there is any debugger connected to the device.

```
protected boolean execute(@NonNull Context context){
  return !Debug.isDebuggerConnected();
}
```

Listing 5. Function checking if the debugger is connected.

We can also provide a function for detecting Hooking tools. This function is important to prevent tools like Xposed (i.e., JustTrustMe and SSLUnpinning 2.0 modules) from using a process to bypass SSL pinning. It is not exactly a debugging process, but is it is quite similar, as we are debugging the process in memory several times. Listing **??** shows Hooking tool detection function.

```
public void detectHookingFramework() {
  totalTests++;
  String xposedPackageName =
       "de.robv.android.xposed.installer";
  String substratePackageName = "com.saurik.substrate";
  if (checkAppInstalled(xposedPackageName) ||
    checkAppIn-stalled(substratePackageName))
  {
  setDetected(hookingDetected,
       R.string.hooking_detected_positive);
  }}
}
```

Listing 6. Function for detecting Hooking tools.

This way, the application cannot be debugged, as it would detect the debug mode, as shown in Fig. 5 Thus, it is impossible to circumvent SSL pinning tampering with application at runtime. This is due to the fact that all the frameworks used with this aim cannot be directly used.

Step 2. Check of the anti-tampering solution. The used template uncovers anti-tampering control. Thus, it must be added inside the method *checkAppInstalled*, which checks if the application was downloaded from a correct source. The proposed code is highlighted in red in listing 7.

```
@RequiresApi(api = Build.VERSION_CODES.M)
public void detectAntiTampering() throws
     PackageManager.NameNotFoundException,
     No-SuchAlgorithmException {
boolean result = true;
String packageName =
     "com.feedhenry.securenativeandroidtemplate";
PackageManager packageManager =
     this.getContext().getPackageManager();
packageManager.getPackageInfo (packageName, 0);
for (android.content.pm.Signature signature :
packageManager.getPackageInfo(packageName, 0).signatures)
{
byte[] signatureBytes = signature.toByteArray ();
MessageDigest md = MessageDigest.getInstance ( "SHA" );
md.update ( signature.toByteArray () );
final String currentSignature = Base64.encodeToString (
     md.digest (),
Base64.DEFAULT );
//compare signatures
if ("478yYkKAQF+KST8y4ATKvHkYibo".equals (
     currentSignature )) {
result = true;
}
}

if (!result) {
WarningDialog warning = WarningDialog.createWarningDialog (
"Application is not original");
warning.show(getFragmentManager(), "device_warning");
}
}
}
```

Listing 7. New code to check anti-tampering.

With this code, APK (Android Application Package) can be modified, and compiled again. The application then warns about the existence of an error in the signature verification, as shown in Fig. 6. This step prevents attackers from SSL pinning bypassing, as the attackers should also bypass the modifications.

Step 3. Code obfuscation. Code obfuscation is done just to hide SSL pinning methods, and the verifications mentioned above. This step makes it difficult for the attacker to check the source code. This way, we are preventing any bypassing method. The obfuscation may be configured in the setup of the project using a third-party library, as mentioned previously. The templated is already prepared to use obfuscation, and it is preconfigured to use Proguard with this aim. Listing 8 the obfuscation code. The obfuscation code may be configured in the gradle of the application. The functioning is similar for other packaging systems, like iOS systems.

```
release {
  versionNameSuffux System.getenv("CIRCLE_BUILD_NUM")
  signingConfig signingConfigs.release
  minifyEnabled true
  proguardFiles.getDefaultProguardFile(
    'proguard-android.txt'),
    'proguard-rules.pro'
}
```

Listing 8. Obfuscation code configuration.

We can check Proguard configuration opening the files mentioned as in listing 8. Moreover, a piece of code of the resulting configuration might be as shown in listing 9. In this code, the two first lines are referenced to the used obfuscation tools. The third line is used to erase the logs.

```
  -dontwarn com.google.errorprone.annotations.*
  -dontwarn okio.**
  -dontwarn org.slf4j.**
```

Listing 9. Proguard configuration.

After obfuscation is done, the code might be decompiled from the APK. However, the code is complete unpredictable, and it is impossible to determine which functions are responsible for the verification of the debug mode detection. Listing 10 shows an example of obfuscated code from the functions of listing 7.

```
public abstract class b implements a {
  private Fragment a;
  private c b;

  public b(Fragment fragment) {
    this.a = fragment;
    this.b = new c();
  }

  private Context d() {
    return this.a.getActivity();
  }

  public void a() {
    Context d = d();
    if (d != null) {
      this.b.a(d);
```

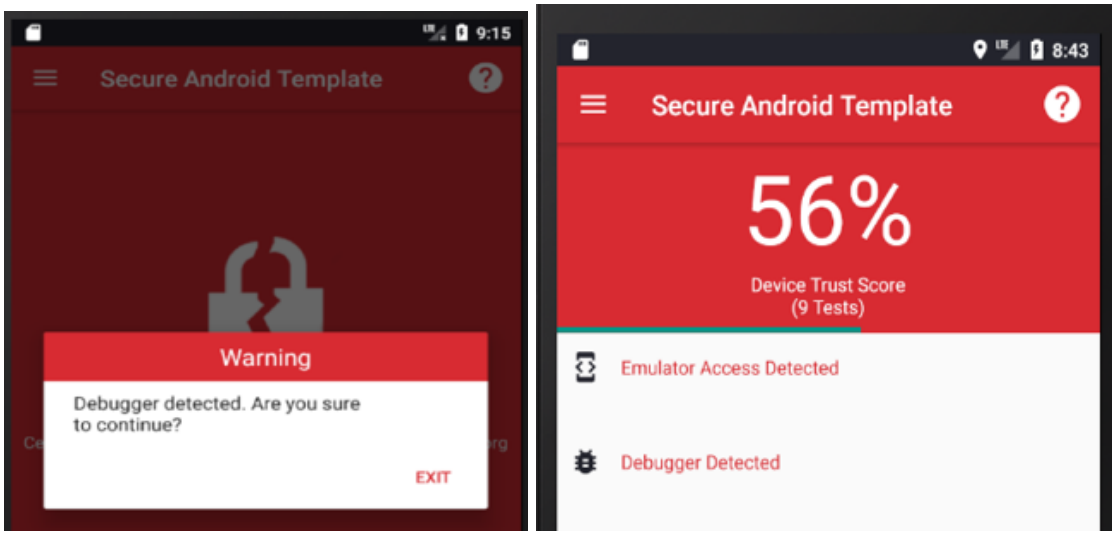


Figura 5. Results debug detection in the template and device trust score.

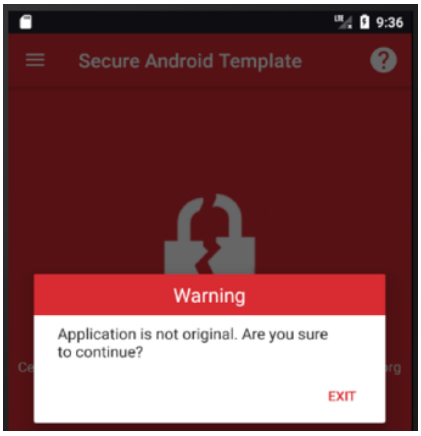


Figura 6. Results of the Application after checking application originality.

```
    }
  }

  ...
}
```

Listing 10. Obfuscated code extracted.

With these three steps, all the methods to bypass SSL pinning are fully invalid, so that a more secure communication channel is created, and other controls do not need to be checked.

V. Conclusions

This paper presents some guidelines for implementing secure communications. We offer solutions to SSL pinning problems, introducing some good practices for all the mobile application developers and users.

SSL/TLS provides confidentiality, message integrity, and authentication in data communication. However, SSL/TLS is a non-free vulnerability protocol since it can be broken by MiTM attacks. The pinning technique has emerged in the last years as a security control to fortified applications against MiTM attacks. SSL pinning is also vulnerable when it is not well implemented. There are several ways to circumventing it, like using SSL Kill Switch 2, a dynamic analysis of the application code, or not implementing anti-tampering.

We propose a securing mechanism that implements three security measures. First, an anti-tampering solution must be deployed, modifying some parts of the application. Second, it is necessary to implement debug detection, to prevent the application code from being controlled. Finally, obfuscating code converts our existing code into a more illegible code. This way, SSL/TLS is totally secure.

Concluding, we used an Android-based templated mobile application to implement the proposed measures, and we demonstrated that it is converted into a secure applica-tion using the SSL Pinning mechanism.

As future lines, we would like to test the proposed mechanism with more applications, to prove the universality of the method.

Acknowledgement

This work has been partially funded by the Ministry of Science and Technology of Spain through ECLIPSE (RTI2018-094283-B-C33), the Junta de Andalucia via the PIRAMIDE and METAMORFOSIS projects, the European Regional Development Fund (ERDF/FEDER). The authors would like to thank the Cátedra de Telefónica “Inteligencia en la red“ of the Universidad de Sevilla for its support.

Referencias

[1] Li, D., Guo, B., Shen, Y., Li, J., Huang, Y.: The evolution of open-source mobile appli-cations: An empirical study. Journal of Software: Evolution and Process 29 (7), Article number e1855 (2017).

[2] Unal, P., Temizel, T.T., Eren, P.E.: What installed mobile applications tell about their owners and how they affect users’ download behavior. Telematics and Informatics 34 (7), 1153-1165 (2017).

[3] Khan, J., Abbas, H., Al-Muhtadi, J. Survey on mobile user’s data privacy threats and defense mechanisms. In: 12th Iberian Conference on Information Systems Technolo-gies, CISTI, article number 7975981, Lisbon, Portugal (2017).

[4] D’Orazio, C.J., Choo, K-K.R. A technique to circumvent SSL/TLS validations on iOS devices. Future Generation Computer Systems 74, 366-374 (2017).

[5] Mueller, B., Schleier, S. OWASP Mobile Application Security Verification Standard v 1.0. Last Consulted: March 2018.

[6] Dhawale, C.A., Misra, S., Jambhekar, N.D., Thakur, S.U. Mobile computing security threats and solution. International Journal of Pharmacy and Technology 8 (4), 23075-23086 (2016).
[7] OWASP Mobile Top 10 2016. https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10. Last modified: February 2017.
[8] Razaghpanah, A., Sundaresan, S., Niaki, A.A, Amann, J., Vallina-Rodriguez, N., Gill, P. Studying TLS usage in Android apps. In: Proceedings of the 13th International con-ference on emerging technologies, CoNEXT 2017, pp. 350-362, Ingeon, South Korea (2017).
[9] Fahl, S., Harbach, M., Perl, H., Koetter, M., Smith, M. Rethinking SSL development in an appified world. In: Proceedings of the ACM SIGSAG Conference on Computer & Communications Security, CCS 2013, pp. 49-60, Berlin, Germany, 2013.
[10] Kim, S., Han, H., Shin, D., Jeun, I., Jeong, H. A study of International Trend Analysis on Web Service Vulnerabilities in OWASP and WASC. In: 3rd International Confer-ence on Information Security and Assurance, ISA 2009, LNCS, vol. 5576, pp. 788-796. Springer, Heidelberg (2009).
[11] Szczepanik, M., Jozwiak, I. Security of mobile banking applications. Advances in In-telligent Systems and Computing 635, 412-419 (2018).
[12] Hickman, K. The SSL Protocol. Netscape Communications Corp (1995).
[13] Dierks, T., Rescorla, E. The TLS Protocol Version 1.2. RFC 5246 (2008).
[14] Varela-Vaca, A.J., Gasca, R.M. Towards the automatic and optimal selection of risk treatments for business processes using a constraint programming approach. Information & software technology, vol. 55(11), pp. 1948-1973 (2013).
[15] Oracle – Java Secure Socket Extension (JSSE) Reference Guide. https://docs.oracle.com/javase/8/docs/technotes/guides/security/jsse/JSSERefGuide.html (2018). Last consulted: April 2018.
[16] OpenSSL. https://www.openssl.org/. Last consulted: April 2018.
[17] LibreSSL. http://www.libressl.org/. Last consulted: April 2018.
[18] GNUTLS. https://www.gnutls.org/. Last consulted: April 2018.
[19] Andzakovic, D. Bypassing SSL Pinning on Android via Reverse Engineering. https://security-assessment.com/files/documents/whitepapers/Bypassing%20SSL%20Pinning%20on%20Android%20via%20Reverse%20Engineering.pdf. Last consulted: March 2018.
[20] Apple Inc. Security Transforms Programming Guide. https://developer.apple.com/library/content/documentation/Security/Conceptual/SecTransformPG/SigningandVerifying/SigningandVerifying.html. Last consulted: March 2018.
[21] ProGuard. https://www.guardsquare.com/en/proguard. Last consulted: April 2018.
[22] iXGuard. https://www.guardsquare.com/en/ixguard. Last consulted: April 2018.
[23] AeroGear Services Android SDK. https://github.com/aerogear/aerogear-android-sdk. Last consulted: April 2018.
[24] FeedHenry Templates – RedHat, http://feedhenry.org/. Last consulted: April, 2018.

A Review of Key Enumeration Algorithms for Cold Boot Attacks

Ricardo Villanueva Polanco
Universidad del Norte, Barranquilla, Colombia
rpolanco@uninorte.edu.co

***Abstract*—In this paper, we study the cold boot attack setting. In this setting, the attacker with physical access to a machine may recover cryptographic key information of a cryptographic scheme via this data remanence attack. We first describe the attack setting and then pose the problem of key recovery in a general way and establish a connection between the key recovery problem and the key enumeration problem. The latter problem arises in the side-channel attack literature, where, for example, the attacker might procure scoring information for each byte of an AES key from a side-channel attack and then want to efficiently enumerate and test a large number of complete 16-byte candidates until the correct key is found. Therefore, we study several algorithms to solve the key enumeration problem, such as the optimal key enumeration algorithm (OKEA) and several other non-optimal key enumeration algorithms. Additionally, we proposed variants of some of them and make a comparison of all of them, highlighting their strengths and weaknesses.**

***Index Terms*—Cold Boot Attacks, Key Recovery Problem, Key Enumeration Problem, Key Enumeration Algorithms**

Type of Contribution: *Original Research Paper*

I. Introduction

A cold boot attack is a type of data remanence attack by which sensitive data are read from a computer's main memory after supposedly having been deleted. This attack relies on the data remanence property of DRAM to retrieve memory contents that remain readable in the seconds to minutes after power has been removed. This attack was first described in the literature by Halderman et al. nearly a decade ago [5] and since then it has received significant attention. In this setting, an attacker with physical access to a computer can retrieve content from a running operating system after using a cold reboot to restart the machine. A running computer is cold-booted when the operating system is not shut down in an orderly manner, skipping file system synchronisation and other activities that would occur on an orderly shutdown. Therefore, after cold-rebooting the machine, such an attacker may use a removable disk to boot a lightweight operating system, which is then used to dump the contents of pre-boot physical memory to a file. Further analysis can then be performed against the data that was dumped from memory to find various sensitive information, such as cryptographic keys contained in it [5]. Unfortunately for such an adversary, the bits in memory will experience a process of degradation once the computer's power is interrupted. This implies that if the adversary can retrieve any data from the computer's main memory after the power is cut off, the extracted data will probably have random bit fluctuations, i.e., the data will be noisy, or rather, be dissimilar from the original data.

Because only a noisy version of the original key may be retrievable from main memory once the attacker discovers the location of the data in it, the adversary's main task then becomes the mathematical problem of recovering the original key from a noisy version of that key. Additionally, the adversary may have access to reference cryptographic data created using that key (e.g. cipher-texts for a symmetric key encryption scheme) or have a public key available (in the asymmetric setting). So the focus of cold boot attacks after the initial work pointing out their feasibility [5] has been to develop algorithms for efficiently recovering keys from noisy versions of those keys for a range of different cryptographic schemes, whilst exploring the limits of how much noise can be tolerated. Heninger and Shacham [6] focussed on the case of RSA keys, giving an efficient algorithm based on Hensel lifting to exploit redundancy in the typical RSA private key format. This work was followed up by Henecka, May and Meurer [7] and Paterson, Polychroniadou and Sibborn [15], with both papers also focusing on the mathematically highly structured RSA setting. The latter paper in particular pointed out the asymmetric nature of the error channel intrinsic to the cold boot setting and recast the problem of key recovery for cold boot attacks in an information theoretic manner. On the other hand, Lee et al. [9] were the first that discussed these attacks in the discrete logarithm setting, however their proposed algorithm would likely be unable to recover keys that were affected by particularly high noise levels in the true cold boot scenario, i.e., only assuming a bit-flipping model. This work was improved upon by Poettering and Sibborn[18], who exploited redundancies found in the in-memory private key representations from two ECC implementations found in TLS libraries and developed cold boot key-recovery algorithms that were applicable to the true cold boot scenario. Other papers have considered cold boot attacks in the symmetric key setting, including Albrecht and Cid [2] who focused on the recovery of symmetric encryption keys in the cold boot setting by employing polynomial system solvers, and Kamal and Youssef [8] who applied SAT solvers to the same problem. Finally, recent research papers have explored cold boot attacks on post-quantum cryptographic schemes. [1] focused on schemes based on the ring -and module - variants of the Learning with Errors (LWE) problem, while the paper by Paterson et. al [16] focused on cold boot attacks on NTRU.

II. Preliminaries

A. Cold Boot Attack Model

Our cold boot attack model assumes that the adversary can obtain a noisy version of a secret key (using whatever format is used to store it in memory). We assume that the corresponding public parameters are known exactly (without noise). We do not consider here the important problem of how to locate the appropriate area of memory in which the

secret key bits are stored, though this would be an important consideration in practical attacks. Our aim is then recover the secret key. Note that it is sufficient to recover a list of key candidates in which the true secret key is located, since we can always test a candidate by executing known algorithms linked to the scheme we are attacking.

We assume throughout that a 0 bit of the original secret key will flip to a 1 with probability $\alpha = P(0 \to 1)$ and that a 1 bit of the original private key will flip with probability $\beta = P(1 \to 0)$. We do not assume that $\alpha = \beta$; indeed, in practice, one of these values may be very small (e.g. 0.001) and relatively stable over time, while the other increases over time. Furthermore, we assume that the attacker knows the values of α and β and that they are fixed across the region of memory in which the private key is located. These assumptions are reasonable in practice: one can estimate the error probabilities by looking at a region where the memory stores known values (e.g. where the public key is located), and the regions are typically large.

B. Problem Statement

Let us suppose that a noisy version of the encoding of the secret key $\mathtt{r} = b_0 b_1 b_2 \dots b_W$ can be represented as a concatenation of $\mathcal{N} = W/w$ chunks, each on w bits. Let us name the chunks $\mathtt{r}^0, \mathtt{r}^1, \dots, \mathtt{r}^{\mathcal{N}-1}$ so that $\mathtt{r}^i = b_{i \cdot w} b_{i \cdot w+1} \dots b_{i \cdot w+(w-1)}$. Additionally, we suppose there is a key recovery algorithm that constructs key candidates $\mathtt{c}$ for the encoding of the secret key and that these key candidates $\mathtt{c}$ can also be represented by concatenations of chunks $\mathtt{c}^0, \mathtt{c}^1, \dots, \mathtt{c}^{\mathcal{N}-1}$ in the same way.

The method of maximum likelihood estimation then suggests picking as $\mathtt{c}$ the value that maximises $\mathbf{P}(\mathtt{c}|\mathtt{r})$. Using Bayes' theorem, this can be rewritten as $\mathbf{P}(\mathtt{c}|\mathtt{r}) = \frac{\mathbf{P}(\mathtt{r}|\mathtt{c})\mathbf{P}(\mathtt{c})}{\mathbf{P}(\mathtt{r})}$. Note that $\mathbf{P}(\mathtt{r})$ is a constant and $\mathbf{P}(\mathtt{c})$ is also a constant, independent of $\mathtt{c}$. Therefore, the ML estimation suggests picking as $\mathtt{c}$ the value that maximises $\mathbf{P}(\mathtt{r}|\mathtt{c}) = (1-\alpha)^{n_{00}} \alpha^{n_{01}} \beta^{n_{10}} (1-\beta)^{n_{11}}$, where n_{00} denotes the number of positions where both $\mathtt{c}$ and $\mathtt{r}$ contain a 0 bit, n_{01} denotes the number of positions where $\mathtt{c}$ contains a 0 bit and $\mathtt{r}$ contains a 1 bit, etc. Equivalently, we may maximise the log of these probabilities, viz. $\log(\mathbf{P}(\mathtt{r}|\mathtt{c})) = n_{00}\log(1-\alpha) + n_{01}\log\alpha + n_{10}\log\beta + n_{11}\log(1-\beta)$. Therefore, given a candidate $\mathtt{c}$, we can assign it a score, namely $S_{\mathtt{r}}(\mathtt{c}) := \log(\mathbf{P}(\mathtt{r}|\mathtt{c}))$.

Assuming that each of the at most 2^w candidate values for chunk $\mathtt{c}^i (0 \leq i < W/w)$ can be enumerated, then its own score also can be calculated as $S'_{\mathtt{r}^i}(\mathtt{c}^i) = n^i_{00}\log(1-\alpha) + n^i_{01}\log\alpha + n^i_{10}\log\beta + n^i_{11}\log(1-\beta)$, where the n^i_{ab} values count occurrences of bits across the i-th chunks, $\mathtt{c}^i, \mathtt{r}^i$. So we have $S_{\mathtt{r}}(\mathtt{c}) = \sum_{i=0}^{\mathcal{N}-1} S'_{\mathtt{r}^i}(\mathtt{c}^i)$. Hence we may assume we have access to $\mathcal{N}$ lists of chunk candidates, where each list contains up to 2^w entries. A chunk candidate is defined as a 2-tuple of the form $(score, value)$, where the first component $score$ is a positive real number (candidate score) while the second component $value$ is an array of w-bit strings (candidate value). The question then becomes: can we design efficient algorithms that traverse the lists of chunk candidates to combine chunk candidates $\mathtt{c}^i$, obtaining complete key candidates $\mathtt{c}$ having high total scores obtained by summation? This question has been previously addressed in the side-channel analysis literature[3], [4], [10], [11], [12], [17], [19], with a variety of different algorithms being possible to solve the problem.

Let $L^i = [c^i_0, c^i_2, \dots, c^i_{m_i-1}]$ be the list of chunk candidates for chunk i, $0 < m_i \leq 2^w$. Let $\mathtt{c}^{i_0}_{j_0}, \dots, \mathtt{c}^{i_n}_{j_n}$ be chunk candidates, $0 \leq i_0 < \dots < i_n < \mathcal{N}, 0 \leq j_i < m_i$. The function $\mathtt{combine}(\mathtt{c}^{i_0}_{j_0}, \dots, \mathtt{c}^{i_n}_{j_n})$ returns a new chunk candidate $\mathtt{c}$ such that $\mathtt{c} = (\mathtt{c}^{i_0}_{j_0}.score + \dots + \mathtt{c}^{i_n}_{j_n}.score, \mathtt{c}^{i_0}_{j_0}.value || \dots || \mathtt{c}^{i_n}_{j_n}.value)$. Note that when $i_0 = 0, i_1 = 1, \dots, i_{\mathcal{N}-1} = \mathcal{N}-1$, $\mathtt{c}$ will be a full key candidate.

Definition 1: The key enumeration problem entails traversing the $\mathcal{N}$ lists L^i, $0 \leq i < \mathcal{N}$, while picking a chunk candidate $c^i_{j_i}$ from each L^i to generate full key candidates $\mathtt{c} = \mathtt{combine}(\mathtt{c}^{i_0}_{j_0}, \dots, \mathtt{c}^{i_n}_{j_n})$. Moreover, we call an algorithm generating full key candidates $\mathtt{c}$ a key enumeration algorithm (KEA).

Note that the key enumeration problem has been stated in a general way, however there are many other variants of this problem. These variants relate to the manner in which the key candidates are generated by a key enumeration algorithm.

A variant consists in enumerating key candidates c such that their total accumulated scores follow a specific order. For example, in many side-channel scenarios it is desirable to enumerate key candidates $\mathtt{c}$ starting at the one having the highest score, followed by the one having the second highest score and so on. In these scenarios, we need a key enumeration algorithm to enumerate high scoring key candidates in decreasing order based on their total accumulated scores. For example, such an algorithm would allow us to find the top M highest scoring candidates in decreasing order, where $1 \leq M \ll 2^W$. Furthermore, such an algorithm is known as an optimal key enumeration algorithm.

Another variant consists in enumerating all the key candidates $\mathtt{c}$ such that their total accumulated scores satisfy a defined condition rather than a specific order. For example, we may need to enumerate all key candidates whose total accumulated scores lie in an interval $[B_1, B_2]$. In this scenario, we need a key enumeration algorithm to enumerate key candidates whose total accumulated scores lie in that interval. Such an algorithm may not enumerate all the key candidates in a decreasing order, still it does need to ensure that all of them will be generated once it has completed. This is, the algorithm only concerns itself with generating all the key candidates whose total accumulated scores satisfy the condition in any order. Such an algorithm would allow us to find the top M highest scoring candidates in any order if the interval is well-defined, for example. Moreover, such an algorithm is commonly known as a non-optimal key enumeration algorithm.

III. Key Enumeration Algorithms

In this section, we will analyse and detail various key enumeration algorithms.

A. An Optimal Key Enumeration Algorithm

We study the optimal key enumeration algorithm (OKEA) that was introduced in [19]. We will give the basic idea behind the algorithm by assuming the encoding of the secret key is represented as two chunks, hence we have access to two lists of chunk candidates.

Let $L^0 = [c^0_0, c^0_1, \ldots, c^0_{m_0-1}]$ and $L^1 = [c^1_0, c^1_1, \ldots, c^1_{m_1-1}]$ be the two lists respectively. Each list is in decreasing order based on the score component of its chunk candidates. Let us define an extended candidate as a 4-tuple of the form $\mathtt{C} := (\mathtt{c}^0_{j_0}, \mathtt{c}^1_{j_1}, j_0, j_1)$ and its score as $\mathtt{c}^0_{j_0}.score + \mathtt{c}^1_{j_1}.score$. Additionally, let Q be a priority queue that will store extended candidates in decreasing order based on their score. Furthermore, let $\mathtt{X}, \mathtt{Y}$ be two vectors of bits that grow as needed. These are employed to track an extended candidate $\mathtt{C}$ in $\mathtt{Q}$. $\mathtt{C}$ is in $\mathtt{Q}$ if only if both $\mathtt{X}_{j_0}$ and $\mathtt{Y}_{j_1}$ are set to 1. By default, all bits in a vector initially have the value 0.

At the initial stage, the queue $\mathtt{Q}$ will be created. Next the extended candidate $(c^0_0, c^1_0, 0, 0)$ will be inserted into the priority queue and both $\mathtt{X}_0$ and $\mathtt{Y}_0$ will be set to 1. In order to generate a new key candidate, the routine nextCandidate, defined in Algorithm 1, should be executed. Note that since both input lists are in decreasing order, the manner in which this algorithm travels through the $m_0 \times m_1$ matrix of key candidates guarantees to output key candidates in a decreasing order based on their total accumulated score, i.e., this algorithm is an optimal key enumeration algorithm.

Algorithm 1 Next highest-scoring key candidate from L^0,L^1.

```
function NextCandidate(Q)
    (c^0_{j0}, c^1_{j1}, j0, j1) ← Q.poll()
    X_{j0} ← 0; Y_{j1} ← 0;
    if (j0 + 1) < L^0.size() and X_{j0+1} = 0 then
        c^0_{j0+1} ← L^0.get(j0 + 1)
        Q.add(c^0_{j0+1}, c^1_{j1}, j0 + 1, j1);
        X_{j0+1} ← 1; Y_{j1} ← 1;
    end if
    if (j1 + 1) < L^1.size() and Y_{j1+1} = 0 then
        c^1_{j1+1} ← L^1.get(j1 + 1)
        Q.add(c^0_{j0}, c^1_{j1+1}, j0, j1 + 1);
        X_{j0} ← 1; Y_{j1+1} ← 1;
    end if
    return combine(c^0_{j0}, c^1_{j1})
end function
```

This algorithm may be generalised to a number of lists greater than 2 by employing a divide and conquer approach, which works by recursively breaking down the problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.

The general algorithm then receives $\mathcal{N}$ lists of chunk candidates, where each list is in decreasing order based on the score component of its chunk candidates. This algorithm constructs a binary tree, where the leaves of this tree are lists, and works its way up to construct and output full candidates in a decreasing order based on their total accumulated score. However, this algorithm presents several disadvantages, such as, its excessive use of memory (affecting its performance) and its inherent serial behaviour (i.e. it is non-parallelizable).

B. A Simple Stack-Based, Depth-First Key Enumeration Algorithm

We next present a memory-efficient, non-optimal key enumeration algorithm that generates key candidates whose total scores are within a given interval $[B_1, B_2]$ that is based on the algorithm introduced by Martin et al. in [11]. We note that the original algorithm is fairly efficient while generating a new key candidate, however its overall performance may be negatively affected by its use of memory, since it was originally designed to store each new generated key candidate, each of which is tested only once the algorithm has completed the enumeration. Our variant, however, makes use of a stack (LIFO queue) during the enumeration process. This helps in maintaining the state of the algorithm. Each newly generated key candidate may be tested immediately and there is no need for candidates to be stored for future processing.

Our variant basically performs a depth-first search in an undirected graph G originated from the $\mathcal{N}$ lists of chunk candidates $L^i = [c^i_0, c^i_n, \ldots, c^i_{m_i-1}]$. This graph G has $\sum_{i=0}^{\mathcal{N}-1} m_i$ vertices, each of which represents a chunk candidate. Each vertex v^i_j is connected to the vertices $v^{i+1}_k, 0 \leq i < \mathcal{N}-1, 0 \leq j < m_i, 0 \leq k < m_i + 1$. At any vertex v_i,the algorithm will check if $c^i_j.score$ plus an accumulated score is within the given interval $[B1,B2]$. If so, it will select the chunk candidate c^i_j for the chunk i and travel forward to the vertex v^{i+1}_0. Or else, it will continue exploring and attempt to travel to the vertex v^i_{j+1}. Otherwise, it will travel backwards to a vertex from the previous chunk $v^{i-1}_k, 0 \leq k < m_{i-1}$, when there is no suitable chunk candidate for the current chunk i.

Note that the previous algorithm uses a simple backtracking strategy, and we improve it by making use of two precomputed tables minArray(maxArray). The entry minArray[i](maxArray[i]) holds the global minimum (maximum) value that can be reached from chunk i to chunk $\mathcal{N}-1$: $\mathtt{minArray}[i]=\min\{\sum_{j=i}^{\mathcal{N}-1} \mathtt{c}^j_{k_j}.score : \mathtt{c}^j_{k_j} \in L^j\}, 0 \leq i < \mathcal{N}$, $\mathtt{maxArray}[i]=\max\{\sum_{j=i}^{\mathcal{N}-1} \mathtt{c}^j_{k_j}.score : \mathtt{c}^j_{k_j} \in L^j\}, 0 \leq i < \mathcal{N}$, and $\mathtt{minArray}[\mathcal{N}]=\mathtt{maxArray}[\mathcal{N}]=0$.

The basic algorithm is sped up by computing maxS (minS), which is the maximum(minimum) score that can be obtained from the current chunk candidate, and then checking if the intersection of the intervals $[\mathtt{minS}, \mathtt{maxS}]$ and $[B1, B2]$ is not empty.

1) Setup: We now introduce a couple of tools that we will use to describe the algorithm, using the following notations. S will denote a stack. This stack S will store 4-tuples of the form $(score, i, j, ind)$, where $score$ is the accumulated score at any stage of the algorithm, i, j are the indices for the chunk candidate c^i_j, and ind is an array of positive integers holding the indices of the selected chunk candidates, i.e., the chunk candidate $c^k_{ind[k]}$ is assigned to chunk k, for each k, $0 \leq k \leq i$.

2) Complete Algorithm: Firstly, at the initialisation stage, the 4-tuple $(0, 0, 0, [])$ will be inserted into the stack S. The main loop of this algorithm will call the function $\mathtt{nextCandidate}(\mathtt{S}, B_1, B_2)$, defined in Algorithm 2, as long as the stack S is not empty. Specifically the main loop will call this function to obtain a key candidate whose score is in the range $[B1, B2]$. Algorithm 2 will then attempt to find such a candidate and once it has found such a candidate, it will return the candidate to the main loop (at this point S may not be empty). The main loop will get the key candidate, process or test it and continue calling the function $\mathtt{nextCandidate}(\mathtt{S}, B_1, B_2)$ as long as S is not empty. Because of the use of the stack S, the state of Algorithm 2 will not be lost, therefore each time the main loop calls it, it will return a new key candidate whose score lie in the interval $[B_1, B_2]$. The main loop will terminate once all possible key candidates whose scores are within the interval $[B_1, B_2]$ have already been generated, which will happen once the stack is

empty.

Algorithm 2 outputs a key candidate in the interval $[B_1, B_2]$.

```
function NextCandidate(S, B_1, B_2)
  while S is not empty do
    (aScore, i, j, indices) ← S.pop
    if j < L^i.size() − 1 then
      S.push((aScore, i, j + 1, ind));
    end if
    uScore ← aScore + c^i_j.score;
    maxS ← uScore + maxArray[i + 1]
    minS ← uScore + minArray[i + 1]
    if maxS ≥ B_1 and minS ≤ B_2 then
      if uScore ≤ B_2 then
        if i = N − 1 then
          if B_1 ≤ uScore then
            Indices ← Indices ||_ [j];
            c ← combine(c^{i_0}_{j_0}, ..., c^{i_{N−1}}_{j_{N−1}})
          end if
        else
          S.push((aScore, i + 1, 0, ind || [j]));
        end if
      end if
    end if
  end while
  return c
end function
```

3) Memory Consumption: We claim that at any stage of the algorithm, there are at most $\mathcal{N}$ 4-tuples stored in the stack S. Indeed, after the stack is initialised, it only contains the 4-tuple $(0, 0, 0, [])$. Note that during the execution of a **while** iteration, a 4-tuple is removed out of the stack and two new 4-tuples might be inserted. Hence, after s **while** iterations have been completed, there will be $N^s_{\mathrm{S}}=1+(-1+l_1)+(-1+l_2)+(-1+l_3)+(-1+l_4)+\cdots+(-1+l_s)$ 4-tuples, where $0 \leq l_r \leq 2$, for $1 \leq r \leq s$.

Suppose now that the algorithm is about to execute the k-th **while** iteration during which the first valid key candidate will be returned. Therefore, $N^{k-1}_{\mathrm{S}}=1+(-1+l_1)+(-1+l_2)+(-1+l_3)+(-1+l_4)+\cdots+(-1+l_{k-1})\leq\mathcal{N}$. During the execution of the k-th **while** iteration, a 4-tuple will be removed and only a new 4-tuple will be considered for insertion in the stack. Therefore, we have that $N^k_{\mathrm{S}}=N^{k-1}_{\mathrm{S}}-1+l_k\leq\mathcal{N}-1+l_k\leq\mathcal{N}$, since $0 \leq l_k \leq 1$. Applying a similar reasoning, we have $N^k_{\mathrm{S}}\leq\mathcal{N}$ for $n > k$.

4) Parallelisation: One of the most interesting features of the previous algorithm is that it is parallelizable. The original authors suggested as a parallelisation method to run instances of the algorithm over different disjoint intervals[11]. Although this method is effective and has a potential advantage as the different instances will produce non-overlapping lists of key candidates with the instance searching over the first interval producing the most-likely key candidates, it is not efficient, since each instance will inevitably repeat a lot of the work done by the other instances. We here propose another parallelisation method that partitions the search space to avoid the repetition of work.

Suppose that we want to have t parallel, independent tasks $T_1,T_2,T_3,...,T_t$ to search over a given interval in parallel. Let $L^i=[c^i_0,c^i_1,...,c^i_{m_i-1}]$ be the list of chunk candidates for chunk i, $0 \leq i \leq \mathcal{N}-1$. We first assume that $t \leq m_0$, where m_0 is the size of L^0. In order to construct these tasks, we partition L^0 into t disjoint, roughly equal-sized sublists $L^0_j, 1 \leq j \leq t$. We set each task T_j to perform its enumeration over the given interval but only considering the lists of chunk candidates $L^0_j,L^1,...,L^{\mathcal{N}-1}$. It is clear that this method can be easily generalised for $m_0<t\ll\prod_{k=0}^{\mathcal{N}-1} m_k$. Additionally, both parallelisation methods can be combined by partitioning the given interval $[B1, B2]$ into n_s disjoint sub-intervals and searching each such sub-interval with t_k tasks, hence amounting to $\sum_{k=1}^{n_s} t_k$ enumerating tasks.

This algorithm shares some similarities with the algorithm Threshold introduced in [10], since Threshold also makes use of an array (`partialSum`) similar to the array `minArray` to speed up the pruning process. However, Threshold works with non-negative integer values (weights) rather than scores. Threshold restricts the scores to weights such that the smallest weight is the likeliest score, by making use of a function that converts scores into weights.

C. A Weight-Based Key Enumeration Algorithm

In this subsection, we will describe a non-optimal enumeration algorithm based on the algorithm introduced in [3]. This algorithm differs from the original algorithm in the manner in which this algorithm builds a precomputed table (`iRange`) and uses it during execution to constructing key candidates whose total accumulated score is equal to a certain accumulated score. This algorithm shares similarities with the stack-based, depth-first key enumeration algorithm described in Section III-B, because both algorithms essentially perform a depth-first search in the undirected graph G. However, this algorithm controls the pruning by the accumulated total score that a key candidate must reach to be accepted. To achieve this, the scores are restricted to positive integer values (weights), which may be derived from a correlation value in a side-channel analysis attack.

This algorithm starts off by generating all key candidates with the largest possible accumulated total weight W_1, and then proceeds to generate all key candidates whose weights are equal to the second largest possible accumulated total weight W_2, and so forth, until generating all key candidates with the minimum possible accumulated total weight W_N . To find a key candidate whose weight is equal to a certain accumulated weight, this algorithm makes use of a simple backtracking strategy, which is efficient because impossible paths can be pruned early. The pruning is controlled by the accumulated weight that must be reached for the solution to be accepted. To achieve a fast decision process during the backtracking, this algorithm precomputes tables for minimal and maximal accumulated total weights that can be reached by completing a path to the right, like the tables `minArray` and `maxArray` introduced in III-B. Additionally, this algorithm precomputes an additional table, `iRange`.

Given $0 \leq i \leq \mathcal{N}$ and $\texttt{minArray}[i]\leq w\leq\texttt{maxArray}[i]$, the entry $\texttt{iRange}[i][w]$ points to a list of integers $L^{(i,w)}=[k_0^{(i,w)},k_1^{(i,w)},...,k_n^{(i,w)}]$, where each entry represents a distinct index of the list L^i, i.e., $0 \leq k_j^{(i,w)} \neq k_l^{(i,w)} < m_i$ for $j \neq l$. The algorithm uses these indices to construct a chunk candidate with an accumulated score s from chunk i to chunk $\mathcal{N}-1$. Algorithm 3 describes precisely how this table is precomputed.

Algorithm 4 describes the backtracking strategy more precisely, making use of the precomputed tables for pruning impossible paths. The integer array `TWeights` contains accumulated weights in a selected order, where an entry $w \in$ `TWeights` must satisfy that the list $\texttt{iRange}[0][w]$ is non-empty, i.e., $\texttt{iRange}[0][w].\texttt{size}()>0$. This helps in constructing a key candidate with an accumulated score s from chunk 0 to chunk $\mathcal{N}-1$. In particular, `TWeights` may

Algorithm 3 precomputes the table `iRange`.

```
function PrecomputeIRange( )
    iRange[𝒩][0] ← [0];
    for i = 𝒩 − 1 to 0 do
        for w = minArray[i] to maxArray[i] do
            L^(i,w) ← []
            for k = 0 to m_i − 1 do
                cw ← w − c^i_k.score
                if iRange[i + 1][cw].size() > 0 then
                    L^(i,w).add(k)
                end if
            end for
            if L^(i,w).size() > 0 then
                iRange[i][w] ← L^(i,w);
            end if
        end for
    end for
    return iRange
end function
```

Algorithm 4 enumerates key candidates for given weights.

```
function KeyEnumeration(TWeights, iRange)
    for w ∈ TWeights do
        i ← 0;
        k[0] ← (0, iRange[0][w].get(0)); 2-tuple (e_1, e_2)
        cw ← w;
        while i ≥ 0 do
            while i < 𝒩 − 1 do
                cw ← cw − c^i_{k[i].e_2}.score;
                i ← i + 1;
                k[i] ← (0, iRange[i][cw].get(0));
            end while
            c ← combine(c^0_{k[0].e_2}, c^1_{k[1].e_2}, ..., c^{𝒩−1}_{k[𝒩−1].e_2});
            Test(c);
            lim ← iRange[i][cw].size() − 1;
            while i ≥ 0 and k[i].e_1 ≥ lim do
                i ← i − 1
                if i ≥ 0 then
                    cw ← cw + c^i_{k[i].e_2}.score;
                    lim ← iRange[i][cw].size() − 1;
                end if
            end while
            if i ≥ 0 then
                next ← k[i].e_1 + 1;
                k[i] ← (next, iRange[i][cw].get(next));
            end if
        end while
    end for
    return iRange
end function
```

be set to $[W_1,W_2,...,W_N]$, i.e., the array containing all possible accumulated scores that can be reached from chunk 0 to chunk $\mathcal{N}-1$. Furthermore, the order in which the elements in the array `TWeights` are arranged is important. For this array $[W_1,W_2,...,W_N]$, for example, the algorithm will first enumerate all key candidates with accumulated weight W_1, then all those with accumulated weight W_2 and so on. This guarantees a certain quality, since good key candidates will be enumerated earlier than worse ones. However key candidates with the same accumulated weight will be generated in no particular order, so a lack of precision in converting scores to weights will lead to some decrease of quality.

1) Memory Consumption: Besides the precomputed tables, it is easy to see that Algorithm 4 makes use of negligible memory while enumerating key candidates. Indeed, testing key candidates is done on the fly to avoid storing them during enumeration. However, the table iRange may have many entries. Indeed, it can be shown that the number of bits `iRange` occupies in memory after Algorithm 3 has completed its execution is $T_b=B_{int}+B_p+\sum_{i=0}^{\mathcal{N}-1}\sum_{w\in W_i}(n^{i,w}\cdot B_{int}+B_p)$,where B_{int} is the number of bits to store an integer, B_p is the number of bits to store a pointer, $n^{i,w}$ $(1 \leq n^{i,w} \leq m_i)$ is the number of entries of the list $L^{i,w}$, W_i is the set of integers in the range from `minArray[i]` to `maxArray[i]` such that $L^{i,w}$.`size`$() > 0$. Note that $\mid W_i \mid$ may increase if the size of the range [`minArray[i]`,`maxArray[i]`] is large, which relies on the scaling technique used to get a positive integer from a real number.

2) Paralellisation: Suppose we would like to have t tasks $T_1,T_2,T_3,\cdots,T_t$ executed in parallel to enumerate key candidates whose accumulated total weights are equal to those in the array `TWeights`. We can split the array `TWeights` into t disjoint sub-arrays `TWeights`$_i$, and then set each task T_i to run Algorithm 4 through the sub-array `TWeights`$_i$. As an example of a partition algorithm to distribute the workload among the tasks, we set the sub-array `TWeights`$_i$ to contain elements with indices congruent to i mod t from `TWeights`. Additionally, note that if we have access to the number of candidates to be enumerated for each score in the array `TWeights` beforehand, we may design a partition algorithm for distributing the workload among the tasks almost evenly.

D. A Key Enumeration Algorithm using Histograms

In this subsection, we will describe a non-optimal key enumeration algorithm introduced in [17].

1) Setup: We now introduce a couple of tools that we will use to describe the sub-algorithms used in the algorithm of [17], using the following notations: H will denote a histogram, N_b will denote a number of bins, b will denote a bin and x a bin index.

The function H_i=`createHist`(L^i,N_b) creates a standard histogram from the list of chunk candidates L_i with N_b linearly-spaced bins. Given a list of chunk candidates L_i, the function `createHist` will first calculate both the minimum score min and maximum score max among all the chunk candidates in L^i. It will then partition the interval $I=[min,max]$ into subintervals $I_0=[min,min+\delta),I_1=[min+\delta,min+2\delta),...,I_{N_b-1}=[min+(N_b-1)\delta,max]$, where $\delta=\frac{max-min}{N_b}$. It will then proceed to build the list L_{H_i} of size N_b. The entry $0 \leq x \leq N_b$ of L_{H_i} will point to a list that contains all chunk candidates from L_i such that their scores lie in I_x. The returned standard histogram H_i is therefore stored as the list L_{H_i} whose entries will point to lists of chunk candidates. For a given bin index x, L_{H_i}.`get`(x) outputs the list of chunk candidates contained in the bin of index x of H_i. Therefore, $H_i[x]=L_{H_i}$.`get`(x).`size()` is the number of chunk candidates in the bin of index x of H_i.

The function $H_{1,2}$=`conv`(H_1,H_2) computes the convolution $H_{1,2}$ from two histograms H_1 and H_2 of sizes n_1 and n_2 respectively, where $H_{1,2}[k]=\sum_{i=0}^{k} H_1[i]\cdot H_2[k-i]$. The computation of $H_{1,2}$ is done efficiently by using Fast Fourier Transformation (FFT) for polynomial multiplication, where each histogram is seen as a polynomial in coefficient form. The convoluted histogram $H_{1,2}$ is therefore stored as a list of integers.

The method `size`() returns the number of bins of a histogram. This method simply returns L.`size`(), where L is the underlying list used to represent the histogram. Also, given a standard histogram H_i and an index $0 \leq x \leq H_i$.`size`(), the method H_i.`get`(x) outputs the list of all chunk candidates contained in the bin of index x of H_i, i.e., this method simply returns the list L_{H_i}.`get`(x).

This key enumeration algorithm uses histograms to represent scores, and the first step of the key enumeration is a convolution of histograms modelling the distribution of the $\mathcal{N}$ lists of scores. This step is described by Algorithm 5.

This key enumeration algorithm enumerates key candidates that are ranked between two bounds R_1 and R_2 $(R_1 \leq R_2)$,

Algorithm 5 computes standard and convoluted histograms.

```
function CREATEHISTOGRAMS(L^0, L^1, ..., L^{N-1}, N_b)
    H_0 ← createHist(L^0, N_b);
    H_1 ← createHist(L^1, N_b);
    H_{0,1} ← conv(H_0, H_1);
    for i = 2 to N − 1 do
        H_i ← createHist(L^i, N_b);
        H_{0,i} ← conv(H_i, H_{0,i−1});
    end for
    return H = [H_0, H_1, ..., H_{N−1}, H_{0,1}, ..., H_{0,N−1}];
end function
```

so it computes the corresponding indices x_{start} and x_{stop} of bins of $H_{0,\mathcal{N}-1}$, where x_{start} is the greatest index such that $\sum_{j=x_{start}}^{H_{0,\mathcal{N}-1}.\mathtt{size}()-1} H_{0,\mathcal{N}-1}[j] \geq R_1$, while x_{stop} is the greatest index such that $\sum_{j=x_{stop}}^{H_{0,\mathcal{N}-1}.\mathtt{size}()-1} H_{0,\mathcal{N}-1}[j] \geq R_2$ (hence $x_{stop} \leq x_{start}$).

Given the list of histograms of scores H and the indices of bins of $H_{0,\mathcal{N}-1}$ between which we want to enumerate, the enumeration simply consists of performing a backtracking over all the bins between x_{start} and x_{stop}. More precisely, during this phase we recover the bins of the initial histograms (i.e. before convolution) that were used to build a the convoluted histogram $H_{0,\mathcal{N}-1}$. For a given bin b with index x of $H_{0,\mathcal{N}-1}$, we have to run through all the non-empty bins $b_0, \ldots, b_{\mathcal{N}-1}$ of indices of $x_0, \ldots, x_{\mathcal{N}-1}$ of $H_0, \ldots, H_{\mathcal{N}-1}$ such that $x_0 + \ldots + x_{\mathcal{N}-1} = x$. Each bin will then contain at least one and at most m_i chunk candidates of the list L^i that we must enumerate. This leads to storing a table `kf` of $\mathcal{N}$ entries, each of which points to a list of chunk candidates. The list pointed to by the entry `kf`$[i]$ holds at least one and at most mi chunk candidates contained in the bin bi of the histogram H_i. Any combination of these $\mathcal{N}$ lists, i.e., picking an entry from each list, results in a key candidate. Algorithm 6 describes more precisely this bin decomposition process. This algorithm calls the function `processKF` which takes as input the table `kf`. This function basically generates all the possible combinations from the $\mathcal{N}$ lists `kf`$[i]$ and may be seen as a particular case of Algorithm 4.

Algorithm 6 performs bin decomposition.

```
function DECOMPOSEBIN(H, i, x_bin, kf)
    if i = 1 then
        x ← H_0.size() − 1;
        while (x ≥ 0) and (x + H_1.size()) ≥ x_bin do
            if H_0[x] > 0 and H_1[x_bin − x] > 0 then
                kf[0] ← H_0.get(x);
                kf[1] ← H_1.get(x_bin − x);
                processKF(kf);
            end if
            x ← x − 1;
        end while
    else
        x ← H_i.size() − 1;
        while (x ≥ 0) and (x + H_{0,i−1}.size()) ≥ x_bin do
            if H_i[x] > 0 and H_{0,i−1}[x_bin − x] > 0 then
                kf[i] ← H_i.get(x);
                DecomposeBin(H, i − 1, x_bin − x, kf);
            end if
            x ← x − 1;
        end while
    end if
end function
```

2) Paralelisation: Suppose we would like to have t tasks $T_1, T_2, T_3, \cdots, T_t$ executing in parallel to enumerate key candidates that are ranked between two bounds R_1 and R_2 in parallel. We can then calculate the indices x_{start}, x_{stop}, and then create the array $X = [x_{start}, x_{start} - 1, \ldots, x_{stop}]$. We then partition the array X into t disjoint sub-arrays X_i, and finally set each task T_i to call the function `DecomposeBin` for all the bins of $H_{0,\mathcal{N}-1}$ with indices in X_i. An example of a partition algorithm that could almost evenly distribute the workload among the tasks is removing an index $x \in X$ such that $H_{0,\mathcal{N}-1}[x]$ is the maximum and add it to the corresponding X_i at each iteration, until X is empty.

3) Memory Consumption: It can be shown that the total number of bits to store all lists $L_{H_i}, 0 \leq i < \mathcal{N}$, is $\sum_{i=0}^{\mathcal{N}-1}(B_p \cdot N_b + B_c \cdot m_i) = \mathcal{N} \cdot B_p \cdot N_b + B_c \cdot \sum_{i=0}^{\mathcal{N}-1} m_i$, where B_p is the number of bits to store a pointer and B_c is the number of bits to store a chunk candidate $(score, [e])$. On the other hand, assuming that an integer is stored in B_{int} bits, then the number of bits for storing all the convoluted histograms is $B_{int} \cdot (N_b - 1)\frac{(\mathcal{N}-1)\mathcal{N}}{2} + B_{int} \cdot N_b(\mathcal{N}-1)$.

4) Equivalence with the path counting approach: The stack-based key enumeration algorithm and the weight-based key enumeration algorithm can be also used for rank computation (instead of enumerating each path, the rank version counts each path). Similarly, the histogram algorithm can also be used for rank computation by simply summing the size of the corresponding bins in $H_{0,\mathcal{N}-1}$ These two approaches were believed to be distinct from each other. However, Martin et al. in [13] show that both approaches are mathematically equivalent, i.e., they both compute the exact same rank when choosing their discretisation parameter correspondingly.

E. Comparison of Key Enumeration Algorithms

In this section, we will make a comparison of the previously described algorithms. We will show some results regarding their overall performance by computing some measures of interest.

1) Implementation: All the algorithms discussed in this chapter were implemented in Java. This is because the Java platform provides the Java Collections Framework to handle data structures, which reduces programming effort, increases speed of software development and quality, and is reasonably performant. Furthermore, the Java platform also easily supports concurrent programming, providing high-level concurrency APIs.

2) Scenario: In order to make a comparison, we will consider a common scenario in which we will run the key enumeration algorithms to measure their performance. Particularly, we generate a random secret key encoded as a bit string of 128 bits, which is represented as a concatenation of 16 chunks, each on 8 bits. We use a bit-flipping model, as described in Section II-B. We particularly set α and β to particular values, namely 0.01 and 0.01 respectively. We then create an original key `k` (AES key) by picking a random value for each chunk i, where $0 \leq i < 16$. Once this key `k` has been generated, its bits will be flipped according to the values α and β to obtain a noisy version of it,`r`. We then use the procedure described in Section II-B to assign a score to each of the 256 possible candidate values for each chunk i. Therefore, once this algorithm has ended its execution, there will be 16 lists, each having 256 chunk candidates.

These 16 lists are then given to an auxiliary algorithm that does the following. For $0 \leq i < 16$, this algorithm outputs 2^e, with $1 \leq e \leq 8$, chunk candidates for the chunk i, ensuring that the original chunk candidate for this chunk is one of the 2^e chunk candidates. This is, the secret key `k` is one out of all the $2^{16 \cdot e}$ key candidates. Therefore, we finally have access to 16 lists, each having 2^e chunk candidates, on which we run each of the key enumeration algorithms. Additionally, on

execution, the candidate keys generated by a particular key enumeration algorithm are not "tested", but rather "verified" by comparing them to the known key. Note that this is done only for the sake of testing these algorithms, however, in practice, it may be not possible to have such an auxiliary algorithm and the candidate keys have to be tested, rather than verified.

3) Results per Algorithms: By running OKEA from Section III-A, we find the following issues: it is only able to enumerate at most 2^{30} key candidates; and its overall performance decreases as the number of key candidates to enumerate increases. In particular, the number of key candidates considered per millisecond per core ranges from 2336 in a 2^{20} enumeration, through 1224 in a 2^{25} enumeration, to 582 in a 2^{30} key enumeration. The main reason for this is that its memory usage grows rapidly as the number of key candidates to generate increases.

Regarding the key enumeration algorithm using histograms from Section III-D, we first analyse the algorithm computing the histograms, i.e. Algorithm 5, and the algorithm computing x_{start}, x_{stop}. These two algorithms were run for $N_b = 10, 20, ..., 100$, $R_1 = 1$ and $R_2 = 2^{30}$ for 100 times. We notice that the running time increases as N_b increases, especially for Algorithm 5 as Fig. 1 shows. On the other hand, the other algorithm shows some negligible variations in its running time. Besides, as expected, we note that the parameter N_b makes the number of bins of $H_{0,\mathcal{N}-1}$ increases, therefore setting this parameter to a proper value helps in guaranteeing the number of key candidates to enumerate while running through the enumeration bounds x_{start}, x_{stop} will be closer to $R_2 - R_1 + 1 = 2^{30} = 1073741824$. Table II shows the number of bins of $H_{0,\mathcal{N}-1}$ and the total number of key candidates to be enumerated between bounds x_{start}, x_{stop} on average. Concerning the memory consumed by the arrays used to store histograms, Table I shows the number of bits for storing both standard histograms and convoluted histograms for values $N_b = 10, 30, 50, 70, 100$.

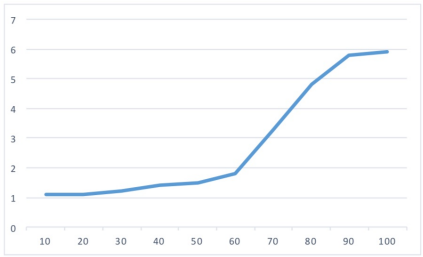


Fig. 1. Running Times of Algorithm 5 of KEA with histograms. The y-axis represents the running time (milliseconds), while the x-axis represents N_b.

We now describe results related to the enumeration algorithm of KEA with histograms, i.e., Algorithm 6. To run this algorithm, we first set the parameter R_1 to 1, R_2 to 2^z, where $z = 30, 33, 36$, and N_b to 60. Once the pre-computation algorithms have ended their execution, we then run Algorithm 6 for each index bin in the range from $x_{start}, x_{start} - 1, x_{start} - 2, \ldots, x_{stop}$. As a result, we find that this algorithm can enumerate $2^{30}, 2^{33}, 2^{36}$ key candidates and that its enumeration rate is between 3500 and 3800 key candidates per millisecond per core. Additionally, as seen, its memory consumption is low.

TABLE I
NUMBER OF BITS FOR STORING HISTOGRAMS IN KEA WITH HISTOGRAMS.

Parameter N_b	Standard Histogram	Convoluted Histogram	Total Number of Bits
10	13312	39360	52672
30	23552	125760	149312
50	33792	212160	245952
70	44032	298560	342592
100	59392	428160	487552

Concerning the stack-based key enumeration algorithm from Section III-B we first calculate appropriate values for B_1 and B_2 by making use of the convoluted histogram $H_{0,\mathcal{N}-1}$ output by Algorithm 5. We then run Algorithm 2 with parameters B_1 and B_2, but limiting the enumeration over this interval to not exceed the number of key candidates to enumerate; this number is obtained from the previous enumeration. As a result, we find that this algorithm can enumerate $2^{30}, 2^{33}, 2^{36}$ key candidates and that its enumeration rate is between 3300 and 3500 key candidates per millisecond per core.

TABLE II
VARIATION OF THE NUMBER OF KEY CANDIDATES IN KEA WITH HISTOGRAMS.

Parameter N_b	Number of bins of $H_{0,\mathcal{N}-1}$	Total number of key candidates
10	145	1412497166
20	224	1310161019
30	305	1260927932
40	384	1228979005
50	464	1207956426
60	545	1191780722
70	625	1178891769
80	705	1169493889
90	784	1162092971
100	864	1156185368

Concerning its memory consumption, the stack-based key enumeration algorithm only uses two precomputed arrays `minArray` and `maxArray`, both of which have $\mathcal{N}+1 = 17$ double entries. Additionally, as pointed out in Section III-B3, at any stage of the algorithm, there are at most 16 4-tuples stored in the stack `S`. Note that a 4-tuple consists of a double entry, two int entries and an entry holding an int array indices. This array indices may have at most 16 entries, each holding an integer value. Therefore, its memory consumption is low.

Lastly, concerning the score-based key enumeration algorithm from Section III-C, we first run its pre-computation algorithms, i.e., the algorithms for computing the tables `minArray`, `maxArray` and `iRange`. As was pointed out in Section III-C1, the size of table `iRange`, hence the running time for calculating it, depends heavily on the scaling technique used to get a positive integer (weight) from a real number (score). We particularly use $score \cdot 10^s$ with $s = 4$ to get an integer score (weight) from a real-valued score. We find that the table `iRange` has around 15066 entries on average. Each of these entries point to a list of integers whose number of entries is about 4 on average. Therefore, we have that the number of bits to store this table is $64 + (32 \cdot 5)(15066) = 2410624$ on average. Furthermore, we run Algorithm 4, but limiting it to not exceed the number of key candidates to enumerate. As a result, we find that this algorithm can enumerate between 2600 and 3000 key candidates per millisecond per core.

4) Discussion: From our previous results, it can be seen that all key enumeration algorithms except for OKEA have a much better overall performance and are able to enumerate a higher number of key candidates. In particular, we find that all of them are able to enumerate $2^{30}, 2^{33}, 2^{36}$ key candidates, while OKEA only can enumerate up to 2^{30}. Its poor performance is caused by their excessive consumption of memory. In particular, OKEA is the most memory-consuming algorithm, hence degrading its overall performance and scalability. In general, scalability is low in optimal key enumeration algorithms [19] considering that not too many candidates can be enumerated, as a result of the exponential growth in their memory consumption. However, by relaxing the restriction on the order in which the key candidates will be enumerated, we are able to design non-optimal key enumeration algorithms, having a better overall performance and scalability. In particular, relaxing this restriction on the order allows for the construction of parallelizable and memory-efficient key enumeration algorithms, as was evinced in this paper and the results previously described. Moreover, all the algorithms save OKEA [3], [11], [17] as described in this paper are non-optimal ones and their respective descriptions and empirical results show that they are expected to have a better overall performance and consume much less computational resources. Table III briefly summarises some quality and functional attributes of the described algorithms. Additionally, note that when an array is used to store a private key and each entry of this array contains much more data than required, in the sense that the number of bits used to store a reduced set of numbers is greater than required, this redundancy as well as the small number of candidates per chunk allow us to generate more "reliable" scores for the candidates per chunk (which would make the key enumeration algorithms find the correct key after enumerating much fewer candidates). From an implementer's view, this may be mitigated by reducing the redundancy used to store a particular private key.

TABLE III
QUALITATIVE AND FUNCTIONAL ATTRIBUTES OF KEY ENUMERATION ALGORITHMS.

Algorithm Name	Parallelizable	Memory Consumption	Scalabilty
Optimal KEA	No	High	Low
Stack-based KEA	Yes	Low	High
Weight-based KEA	Yes	Low	High
KEA with histograms	Yes	Low	High

IV. CONCLUSIONS

In this paper, we investigated the key enumeration problem, since there is a connection between the key enumeration problem and the key recovery problem. In summary, we first stated the key enumeration problem in a general way and then studied and analysed several algorithms to solve this problem. For each studied algorithm, we described its inner functioning, showing its functional and qualitative features, such as memory consumption, amenability to parallelisation and scalability. Furthermore, we proposed variants of some of them, implemented all of them on Java and made an experimental comparison of all of them, drawing special attention to their strengths and weaknesses.

ACKNOWLEDGEMENTS

This work was supported by Colciencias, my sponsor during my Ph.D. studies.

REFERENCES

[1] M. R. Albrecht, A. Deo, K. G. Paterson. Cold Boot Attacks on Ring and Module LWE Keys Under the NTT. TCHES, vol. 2018, no. 3, pp. 173-213, Aug. 2018.

[2] M. Albrecht and C. Cid. Cold boot key recovery by solving polynomial systems with noise. In J. Lopez and G. Tsudik, editors, ACNS 11, vol. 6715 of LNCS, pp. 57 - 72. Springer, Heidelberg, June 2011.

[3] A. Bogdanov, I. Kizhvatov, K. Manzoor, E. Tischhauser and M. Witteman. Fast and memory-efficient key recovery in side-channel attacks. In O. Dunkelman and L. Keliher, editors, SAC 2015, vol. 9566 of LNCS, pp. 310 - 327. Springer, Heidelberg, Aug. 2016.

[4] L. David and A. Wool. A bounded-space near-optimal key enumeration algorithm for multi-subkey side-channel attacks. In H. Handschuh, editor, Topics in Cryptology - CT-RSA 2017 - The Cryptographers' Track at the RSA Conference 2017, San Fran- cisco, CA, USA, February 14-17, 2017, Proceedings, vol. 10159 of LNCS, pp. 311 - 327. Springer, 2017.

[5] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten. Lest we remember: Cold boot attacks on encryption keys. In P. C. van Oorschot, editor, Proceedings of the 17th USENIX Security Symposium, July 28 - August 1, 2008, San Jose, CA, USA, pp. 45 - 60. USENIX Association, 2008.

[6] W. Henecka, A. May, and A. Meurer. Correcting errors in RSA private keys. In T. Rabin, editor, CRYPTO 2010, vol. 6223 of LNCS, pp. 351 - 369. Springer, Heidelberg, Aug. 2010.

[7] N. Heninger and H. Shacham. Reconstructing RSA private keys from random key bits. In S. Halevi, editor, CRYPTO 2009, vol. 5677 of LNCS, pp. 1 - 17. Springer, Heidelberg, Aug. 2009.

[8] A. Kamal and A. M. Youssef. Applications of SAT solvers to AES key recovery from decayed key schedule images. In R. Savola, M. Takesue, R. Falk, and M. Popescu, editors, Fourth International Conference on Emerging Security Information Systems and Technologies,SECURWARE 2010, Venice, Italy, July 18-25, 2010, pp. 216 - 220. IEEE Computer Society, 2010.

[9] H. T. Lee, H. Kim, Y. J. Baek, and J. H. Cheon. Correcting errors in private keys obtained from cold boot attacks. In H. Kim, editor, ICISC 11, vol. 7259 of LNCS, pp. 74 - 87. Springer, Heidelberg, Nov. / Dec. 2012.

[10] J. Longo, D.P. Martin, L. Mather, E. Oswald, B. Sach and M. Stam. How low can you go? Using side-channel data to enhance brute-force key recovery. Cryptology ePrint Archive: Report 2016/609, 2016. https://eprint.iacr.org/2016/609.

[11] D. P. Martin, L. Mather, E. Oswald, and M. Stam. Characterisation and estimation of the key rank distribution in the context of side channel evaluations. In J. H. Cheon and T. Takagi, editors, ASIACRYPT 2016, Part I, vol. 10031 of LNCS, pp. 548 - 572. Springer, Heidelberg, Dec. 2016.

[12] D. P. Martin, J. F. O'Connell, E. Oswald, and M. Stam. Counting keys in parallel after a side channel attack. In T. Iwata and J. H. Cheon, editors, ASIACRYPT 2015, Part II, vol. 9453 of LNCS, pp. 313 - 337. Springer, Heidelberg, Nov. / Dec. 2015.

[13] D. P. Martin, L. Mather and E. Oswald. Two Sides of the Same Coin: Counting and Enumerating Keys Post Side-Channel Attacks Revisited. In N. Smart, editors, Topics in Cryptology – CT-RSA 2018, vol. 10808 of LNCS, springer, 2018.

[14] K. G. Paterson, A. Polychroniadou, and D. L. Sibborn. A coding-theoretic approach to recovering noisy RSA keys. In X. Wang and K. Sako, editors, ASIACRYPT 2012, vol. 7658 of LNCS, pp. 386 - 403. Springer, Heidelberg, Dec. 2012.

[15] K. G. Paterson, R. Villanueva-Polanco R. Cold Boot Attacks on NTRU. In A. Pa- tra, N. Smart, editors, INDOCRYPT 2017, vol. 10698 of LNCS, pp. 107–125, Springer, Nov. 2017.

[16] R. Poussier, F. Standaert and V. Grosso. Simple Key Enumeration (and Rank Esti- mation) Using Histograms: An Integrated Approach In B. Gierlichs,A. Poschmann, editors, CHES 2016, vol. 9813 of LNCS, pp. 61 - 81. Springer, Heidelberg, Jun. 2016

[17] B. Poettering and D. L. Sibborn. Cold boot attacks in the discrete logarithm setting. In K. Nyberg, editor, CT-RSA 2015, vol. 9048 of LNCS, pp. 449 - 465. Springer, Heidelberg, Apr. 2015.

[18] N. Veyrat-Charvillon, B. Gerard, M. Renauld, and F.-X. Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In L. R. Knudsen and H. Wu, editors, SAC 2012, vol. 7707 of LNCS, pp. 390 - 406. Springer, Heidelberg, Aug. 2013.

Protocolos de clave pública en anillos de grupo torcidos

María Dolores Gómez Olvera
Universidad de Almería
Crta. Sacramento S/N, Cañada de San Urbano (Almería)
gomezolvera@ual.es

Juan Antonio López Ramos
Universidad de Almería
Crta. Sacramento S/N, Cañada de San Urbano (Almería)
jlopez@ual.es

Blas Torrecillas Jover
Universidad de Almería
Crta. Sacramento S/N, Cañada de San Urbano (Almería)
btorreci@ual.es

***Resumen*—La Criptografía es la ciencia que estudia la seguridad en las comunicaciones. Actualmente, los protocolos que protegen nuestra privacidad se encuentran en un proceso de renovación, y se están proponiendo nuevos algoritmos que puedan preservar nuestra seguridad, ante la aparición de nuevas amenazas. En el ámbito del álgebra no conmutativa se está investigando en este sentido, y en esta línea proponemos un intercambio de clave en un anillo de grupo torcido mediante un cociclo, con la intención de probar que es una buena estructura en la cual basar nuestros protocolos, y posteriormente implementarlos en ella.**

***Index Terms*—Álgebra no conmutativa, Intercambio de clave, Anillo de grupo, Criptografía de clave pública**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La investigación en Matemáticas es un pilar fundamental para desarrollar sociedades más seguras desde el punto de vista de la ciberseguridad. Entre otras aportaciones (como la creación de protocolos para identificación de vulnerabilidades, detección de intrusos, el análisis de ciberriesgos), son la base de los algoritmos que permiten que nuestras comunicaciones se mantengan íntegras y confidenciales.

La Criptografía es la disciplina que se encarga del estudio de los algoritmos que se utilizan para dotar de seguridad a las comunicaciones, y a las entidades que se comunican. En particular, la investigación en criptografía de clave pública, vital para las comunicaciones actuales, a través de la red, es muy necesaria estos días.

Iniciada en 1976, de la mano de un matemático y un ingeniero, Whitfield Diffie y Martin Hellman, la criptografía que utilizamos actualmente está basada en problemas de Teoría de Números (como el problema del logaritmo discreto o la factorización de enteros) , y está comenzando a presentar ciertas dificultades. La más lejana pero potencialmente más problemática es la amenaza que representa la computación cuántica, ya que los problemas mencionados en los que se basa la seguridad de gran parte de nuestras comunicaciones se podrían resolver mediante el algoritmo de Shor o sus variaciones, en un ordenador cuántico lo suficientemente potente. Además, se buscan algoritmos cada vez más eficientes, para una creciente variedad de dispositivos conectados con información confidencial, muchos de ellos pequeños y con capacidad computacional limitada.

Esto motiva que se estén proponiendo actualmente ambientes alternativos en los que basar la seguridad de nuestras comunicaciones. Una de las principales vías actualmente es el álgebra no conmutativa. En este sentido, se ha venido proponiendo que los anillos de grupo pueden ser una buena base para realizar un intercambio de clave [2], [3], [4], [5], si bien es cierto que algunos de ellos empiezan a presentar ciertos problemas. La propuesta de este trabajo es continuar en esta dirección, proponiendo algunas variaciones, y evitando así las dificultades con las que se encuentran estos algoritmos.

II. Anillos de grupo torcidos

El intercambio que proponemos utiliza una variación del problema de la descomposición (DP), y un anillo de grupo, pero en este caso, torcido mediante un cociclo. Describimos en primer lugar la estructura.

Sea G un grupo multiplicativo, y K un anillo conmutativo unitario. El **anillo de grupo** $K[G]$ está definido como

$$K[G] = \Big\{ \sum_{g_i \in G} r_i g_i \mid r_i \in K \Big\}$$

con $r_i = 0$ para todo i salvo un número finito.

La suma de los elementos en $K[G]$ se define como:

$$\Big(\sum_{g_i \in G} a_i g_i \Big) + \Big(\sum_{g_i \in G} b_i g_i \Big) = \sum_{g_i \in G} (a_i + b_i) g_i$$

Y la multiplicación en $K[G]$ se define de la siguiente forma:

$$\Big(\sum_{g_i \in G} a_i g_i \Big) \Big(\sum_{g_i \in G} b_i g_i \Big) = \sum_{g_i \in G} \Big(\sum_{g_j g_k = g_i} a_j b_k \Big) g_i$$

En nuestro caso, definimos además el concepto de cociclo. Sea G un grupo finito, y K un cuerpo. Una función

$$\alpha : G \times G \longrightarrow K$$

es un **2-cociclo** si

1. $\alpha(g, 1) = \alpha(1, g) = 1$, para todo $g \in G$

2. $\alpha(g,h)\alpha(gh,k) = \alpha(g,hk)\alpha(h,k)$, para todo $g,h \in G$.

Así, dado K un anillo, G un grupo cualquiera, y un 2-cociclo $\alpha : G\times G \longrightarrow U(K)$, donde $U(K)$ denota el grupo de las unidades de K. Un **anillo de grupo torcido** será denotado por $R = K^{\alpha}G$, donde la suma $+$ será la usual en un anillo de grupo, y la multiplicación $*$ para cualesquiera $x, y \in G$ viene dada por

$$x * y = \alpha(x,y)xy$$

III. Intercambio de clave

El intercambio que proponemos es el siguiente: Sea h un elemento cualquiera público de un anillo de grupo R, que escinda como $R = R_1 \oplus R_2$, con R_1 y R_2 tales que los elementos de ambos 'conmuten' a través de una aplicación *. El intercambio de clave entre Alicia y Bruno es el siguiente:

1. Alicia elige un par de elementos $s_A = (g_1, k_1)$, con $g_1 \in A_1 \leq R_1, k_1 \in A_2 \leq R_2$.
2. Bruno elige un par de elementos $s_B = (g_2, k_2)$, con $g_2 \in A_1 \leq R_1, k_2 \in A_2 \leq R_2$.
3. Alicia envía a Bruno $p_A = g_1 h k_1$, y Bruno envía a Alicia $p_B = g_2 h k_2$.
4. Alicia calcula $K_A = g_1 p_B k_1^*$, y Bruno calcula $K_B = g_2 p_A k_2^*$, siendo esta su clave compartida.

$$K = K_A = K_B$$

donde k_1^*, k_2^* son elementos que dependen de k_1, k_2 y el cociclo α.

IV. Ejemplo

Como ejemplo más específico de estructura, proponemos el anillo de grupo $R = GF(2^n)^{\alpha}D_{2m}$, siendo $GF(2^n)$ el cuerpo finito de 2^n elementos, D_{2m} el grupo diédrico de $2m$ elementos, y α el cociclo

$$\alpha : D_{2m} \times D_{2m} \longrightarrow GF(2^n)^*$$

con $\alpha(x^i, x^j y^k) = 1$ y $\alpha(x^i y, x^j y^k) = t^j$, para todo k; con t una raíz primitiva de la unidad que genera $GF(2^n)$.

En este caso, tenemos más concretamente que

- $g_i \in A_1 = R_1 = GF(2^n)[\mathbb{Z}_m]$
- $k_i \in A_2 \leq R_2 = GF(2^n)^{\alpha}[\mathbb{Z}_m]y$, con $A_2 = \left\{ \sum_{i=0}^{m-1} r_i x^i y \in R_2 : r_i = r_{m-i} \right\}$
- k_i^* definido de la siguiente forma: dado un elemento $k_i \in GF(2^n)^{\alpha}D_{2m}$,

$$k_i = \sum_{\substack{0\leq i\leq m-1 \\ k=0,1}} r_i x^i y^k$$

con $r_i \in GF(2^n)$, y $x, y \in D_{2m}$. Definimos el elemento $k_i^* \in GF(2^n)^{\alpha}D_{2m}$ como

$$k_i^* = \sum_{\substack{0\leq i\leq m-1 \\ k=0,1}} r_i t^{-i} x^i y^k$$

con $r_i, t \in GF(2^n)$, y $x, y \in D_{2m}$.

V. Conclusiones

En este documento se propone un intercambio de clave con ciertas características nuevas que suponen una ventaja con respecto a otros protocolos existentes. Además, puede generalizarse para varios usuarios, y también puede definirse un criptosistema relacionado con seguridad equivalente.

Además, hemos implementado parcialmente los protocolos mencionados, y comprobado que ciertos ataques conocidos no suponen una amenaza para los mismos. Por ejemplo, algunos ataques a la estructura del anillo de grupo como [1], [6] o ataques al problema de la descomposición como [7] no son aplicables en nuestro caso.

Referencias

[1] A. Childs, G. Ivanyos: "Quantum computation of discrete logarithms in semigroups", en *J. Math. Cryptology*, vol. 8, n. 4, pp. 405-416, 2014.

[2] M. Eftekhari: "A Diffie-Hellman key exchange protocol using matrices over group rings", en *Groups Complex. Cryptol.*, vol. 4, n. 1, pp. 167-176, 2012.

[3] I. Gupta, A. Pandey, M. Kant DUbey: "A Key Exchange Protocol using Matrices over Group Rings", en *Asian-European Journal of Mathematics*, https://doi.org/10.1142/S179355711950075X, 2018.

[4] M. Habeeb, D. Kahrobaei, D. Koupparis, C. Shpilrain: "Public key exchange using semidirect product of (semi)groups", en *Lecture Notes Comp. Sc.*, vol. 7954, p. 475-486. Springer, 2013.

[5] D. Kahrobaei, C. Koupparis, V. Shpilrain: "Public key exchange using matrices over group rings", *Groups Complex. Cryptol.*, vol. 5, n. 1, pp. 97-115, 2013.

[6] A. Myasnikov, A. Ushakov: "Quantum algorithm for discrete logarithm problem for matrices over finite groups rings", , en *Groups Complexity Cryptology*, vol. 6, n. 1, pp. 31-36, 2013.

[7] V. Roman'kov: "A general encryption scheme using two-sided multiplications with its cryptanalysis", en *https://arxiv.org/pdf/1709.06282.pdf*, 2017.

Comunicaciones VoIP cifradas usando Intel SGX

Raúl Ocaña Isaac Agudo
Network, Information and Computer Security (NICS) Lab
Universidad de Málaga, 29071
{roa, isaac}@lcc.uma.es

***Resumen*—Cada día es más frecuente encontrar servicios en internet gestionados desde plataformas online y con la expansión de la tecnología IoT, los *smartphones*, las *smartTV* y otros tantos dispositivos: la autenticación, la distribución y al fin y al cabo, la comunicación entre extremos puede verse seriamente comprometida si dicha plataforma es atacada. La inclusión de nuevas medidas de seguridad en este tipo de ecosistemas requiere de un cambios sustancial de la arquitectura subyacente en muchos casos, por lo que su avance es lento. En este trabajo se trata de forma concreta el desarrollo de una alternativa *OpenSource* a uno de estos servicios, la telefonía IP (VoIP), que esta expandiéndose cada día más, empezando por redes locales y privadas y llegando a grandes centralitas de conmutación de tele operadoras, consiguiendo así una transmisión de voz segura extremo a extremo transparente para los servidores VoIP, que no requiera modificar la infraestructura subyacente.**

***Index Terms*—Intel SGX, VoIP, Seguridad Extremo a Extremo, Cifrado, Comunicaciones**

Tipo de contribución: *Investigación en desarrollo.*

I. Introducción

Uno de los protocolos predominantes para el desarrollo de las comunicaciones de Voz sobre IP (VoIP) es el protocolo RTP (Real-time Transport Protocol). Mediante el uso de este protocolo se puede producir una comunicación fluida y síncrona, si bien no siempre se hace uso de medidas de seguridad que lo acompañen. El uso de las alternativas seguras, SRTP [2] o ZRTP [3], no está tan extendido como sería deseable; esto puede poner en riesgo la seguridad tanto de la información tratada durante estas llamadas, la de sus interlocutores y su privacidad.

Lo sistemas que dan soporte las comunicaciones VoIP no dejan de ser servicios en línea, y por tanto es amplia la lista de posibles amenazas heredadas: desde ataques lanzados a redes subyacentes, a los protocolos de transporte, a los dispositivos de transmisión VoIP y sus aplicaciones, servidores, puertas de enlace, o incluso a su protocolo de configuración DHCP, llegando incluso al sistema operativo [1]. Esto unido a que en algunos casos la confianza en los proveedores de estos servicios no es plena obliga a trabajar asumiendo una configuración de sistemas *Honestos-pero-Curiosos* [10], es decir que proporcionan el servicio deseado pero pueden estar interesados en los datos de los usarios, en este caso, sus llamadas.

La necesidad por tanto de proveer a estos sistemas de una seguridad adecuada es una tarea compleja, que implica muchos factores independientes. Es por ello que se presenta la idea de encapsular la seguridad de estos protocolos, de manera paralela a la ejecución de las propias aplicaciones y del sistema operativo, en pequeños espacios de memoria, aislados y sellados contra el acceso tanto externo como interno a la propia máquina. Estos espacios de memoria son los Enclaves [4], una tecnología relativamente reciente de Intel llamada SGX (*Secure Guard Extensions*) que permite el almacenamiento y ejecución de código de forma aislada al sistema operativo utilizando al procesador como agente. De esta manera se podría generar confianza entre ambos extremos ignorando cualquier elemento intermedio de la cadena de comunicación, el cual será transparente a nuestro protocolo de seguridad subyacente.

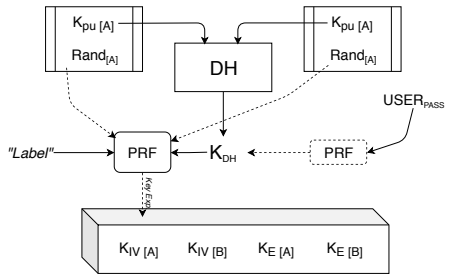


Figura 1. Derivación de claves entre procesos.

Este trabajo parte del desarrollo de una aplicación de Chat cifrado basado en un intercambio DH, donde todas las operaciones criptográficas se implementan dentro una enclave Intel SGX [8] e intenta ampliar su ámbito de aplicación a las llamadas VoIP. Todo el código del proyecto se puede encontrar en el repositorio oficial [9].

II. Implementación

Para gestionar el desarrollo de esta investigación se ha usado como motor de llamadas VoIP la librería PJSIP[1], de dominio público y que ofrece diferentes aplicaciones básicas para pruebas y realización de llamadas. En términos de desarrollo se plantean las siguientes etapas:

II-A. Emparejamiento y generación de claves

La generación de claves es un proceso crítico en todo el diseño; para esta tarea se ha planteado un proceso en cascada que está inspirado en el diseño de TLS (Transport Layer Security) [7] para la derivación y expansión de claves mediante el uso de la función pseudoaleatoria PRF (Pseudo Random Function).

Tal y como podemos ver en la Figura 1, para la derivación de las claves se hace uso del protocolo Diffie-Helman (DH), que se ejecutará bien en un canal paralelo o en el contexto del protocolo SIP (Session Initiation Protocol) usando mensajes instantáneos cuando el servidor los soporte. En ambos casos el servidor será el mismo, variando sólo el canal a través del cual se negocie dicha clave inicial. También se define un mecanismo simplificado en el caso en el que no sea posible ejecutar el protocolo DH, en cuyo caso se genera

[1]Repositorio oficial de PJSIP: https://github.com/pjsip/pjproject.

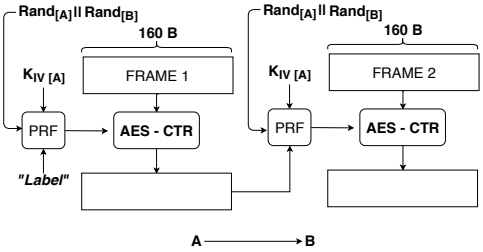


Figura 2. Cifrado de paquetes.

la clave maestra del sistema (MK) a través de la función PRF y usando como entrada una contraseña compartida entre los interlocutores. A partir de esa clave maestra y a través de la función PRF se genera un nuevo bloque de claves (*key expansion*) que alberga las claves de cifrado de ambos clientes, así como la clave de derivación de los IV usados en el cifrado de las tramas. Para dicha expansión de claves se utiliza una *etiqueta* definida y dos números aleatorios aportados por los clientes, esto hace que las claves sean diferentes en cada ejecución del protocolo aunque se utilizara la misma contraseña.

II-B. Proceso de cifrado

El cifrado se ha implementado dentro del mecanismo de codificación de voz, integrándolo en el códec G.722 [5]. A la hora de realizar la codificación se aplica un proceso extra de cifrado, usando el modelo de cifrado secuencial encadenado descrito a continuación (ver Figura 2). Al inicio, se parte de un IV definido y concreto, que ambos clientes son capaces de generar con facilidad tanto en transmisión, como en recepción; este se crea a partir de la concatenación de los números aleatorios proveídos por el cliente que inicia la comunicación, a partir de ahora A, y el que la recibe, en adelante B. En posteriores tramas el cifrado utilizará como *etiqueta* para la generación del IV (también mediante PRF) la última trama enviada. Las tramas son de 160 Bytes y para cifrarlas utilizamos el algoritmo AES en modo CTR [6] usando en cada trama el IV correspondiente. Al no tener que enviar el IV, evitamos tener que alterar el tamaño de las tramas de voz. Este modelo de encadenamiento de las tramas se asimila al modo CBC. Si bien en la figura se muestra el cifrado de las tramas de A a B el cifrado en el otro sentido se realiza de forma análoga pero con las claves correspondientes.

II-C. La aplicación y sus contextos

Como se ha mencionado al principio de este documento este desarrollo usa Intel SGX como sello de garantía. En ambos clientes, los enclaves son los encargados de generar, procesar y almacenar la clave compartida y sus derivadas, así como inicializar los métodos de cifrado y descifrado, que posteriormente serán usados por el códec G.722.

Tal y como se puede ver en la Figura 3, dentro de la aplicación encontramos dos grandes contextos: el primero, que es el que usa SIP o un canal auxiliar previamente acordado para el emparejamiento y la derivación de claves compartidas; y un segundo contexto en el que se lleva a cabo la comunicación RTP y que va a ser el que prácticamente predomine durante el uso de dicha aplicación. En ambos contextos se utiliza un servidor: tanto para el acuerdo de

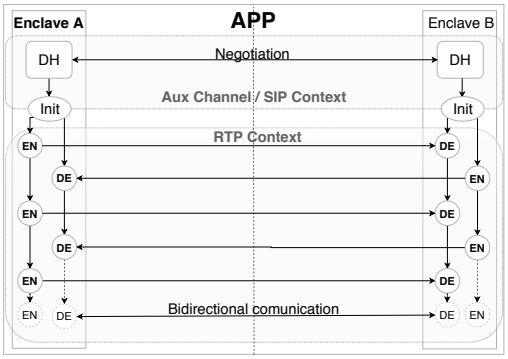


Figura 3. Modelo contextual de la aplicación

claves, como para el desarrollo de la comunicación; que hace de intermediario entre las partes. Gracias al cifrado extremo a extremo implementado dentro de los enclaves SGX, ninguno de los servidores tendrá acceso a la información en claro.

III. Conclusiones y Trabajo Futuro

Este trabajo muestra como se puede integrar un cifrado extremo a extremo de forma casi transparente al servidor VoIP, entrelazando el proceso de cifrado con el de codificación de la voz. Esto permite una comunicación segura extremo a extremo, donde el servidor no tiene acceso en ningún momento a la llamada. Esta solución a nivel de códec evita que el proceso de cifrado tenga lugar en el contexto de la aplicación y que se realicen altos cambios en la librería VoIP, aún así se valora la futura integración de protocolos como SRTP o SRTCP dentro del contexto del enclave, aunque esto implicaría mayores modificaciones en las librerías actuales.

Como trabajo futuro, se plantea la implementación de un cliente plenamente funcional donde el acuerdo de claves DH se realice de forma transparente a través del protocolo SIP, sin la necesidad de utilizar un canal fuera de banda. El encapsulamiento del intercambio DH dentro de SIP mejoraría la escalabilidad y la usabilidad del sistema.

Otras líneas futuras son el estudio de la integración con otros códecs de audio así como la integración con otros clientes o el soporte para autenticación extremo a extremo.

Referencias

[1] SICKER, Douglas C.; LOOKABAUGH, Tom. "VoIP security: Not an afterthought". *Queue*, vol. 2, no 6, p. 56. 2004.

[2] McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP)", RFC 5764, DOI 10.17487/RFC5764, May 2010.

[3] P. Zimmermann, ZRTP: Media Path Key Agreement for Unicast Secure RTP, RFC 6189 (proposed standard), April 2011.

[4] COSTAN, Victor; DEVADAS, Srinivas. Ïntel SGX Explained". *IACR Cryptology ePrint Archive*, vol. 2016, no 086, p. 1-118. 2016.

[5] MERMELSTEIN, Paul. "G. 722: a new CCITT coding standard for digital transmission of wideband audio signals". *IEEE Communications Magazine*, vol. 26, no 1, p. 8-15. 1988.

[6] Dworkin, Morris. Recommendation for block cipher modes of operation. methods and techniques. No. NIST-SP-800-38A. National Inst of Standards and Technology Gaithersburg MD Computer security Div, 2001.

[7] DIERKS, T. y RESCORLA, E. "The Transport Layer Security (TLS) Protocol Version 1.2". *IETF. Request for Comments, 5246*. 2008.

[8] Raúl Ocaña. .Aplicación de chat segura basada en Intel SGX". *TFG, E.T.S.I. Telecomunicaciones, Universidad de Málaga.* 2018.

[9] Repositorio oficial del proyecto: https://github.com/nicslabdev/VoIP_seguro_IntelSGX

[10] Paverd AJ, Martin A, Brown I. "Modelling and automatically analysing privacy properties for honest-but-curious adversaries". *Tech. Rep.*. 2014.

Aplicación de técnicas de transfer learning a problemas de ciberseguridad

David Escudero García
RIASC, Universidad de León
Campus de Vegazana s/n, 24071 León, España
descg@unileon.es

Angel Luis Muñoz Castañeda
Departamento de Matemáticas, Universidad de León
Campus de Vegazana s/n, 24071 León, España
amunc@unileon.es

***Resumen*—Cada vez es más común encontrarse con diferentes aplicaciones de machine learning a diferentes problemas, incluido el ámbito de la ciberseguridad, debido a que permite automatizar procesos importantes como determinar la maliciosidad de aplicaciones software, la detección de ataques de red, etc. No obstante, para construir una solución eficaz es necesario disponer de una cantidad abundante de datos ya clasificados (etiquetados), lo cual puede ser costoso en cuanto a tiempo y recursos.**

El uso de técnicas de transfer learning, que permite reutilizar el conocimiento derivado de un conjunto de datos para mejorar un modelo de machine learning para un un problema con un conjunto de datos distinto, puede ser una buena solución para paliar el problema de la escasez de datos.

En este trabajo se pretende realizar una evaluación del rendimiento predictivo y computacional de técnicas de transfer learning en problemas de ciberseguridad para determinar su eficacia.

***Index Terms*—transfer learning, machine learning, ciberseguridad**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Existen numerosos artículos de investigación en el ámbito de la ciberseguridad que usan técnicas de machine learning para intentar resolver determinados problemas: tanto en detección de intrusiones de red [1], como la detección de malware en móviles [2], etc. Una de las principales dificultades en este ámbito se produce a causa de los datos [3]: puede que la cantidad de datos disponibles para el problema concreto sea insuficiente, o que los datos no tengan etiquetas. Etiquetar los datos manualmente requiere de conocimiento experto y es costoso tanto en tiempo como en recursos. Obtener un conjunto de datos etiquetado por medios propios también resulta poco práctico, así que el uso de técnicas de transfer learning puede paliar este problema.

El marco de aplicación de técnicas de transfer learning es el siguiente: se tiene un dominio objetivo $\mathcal{D}_\mathcal{T} = \{\mathcal{X}_\mathcal{T}, P(X_T)\}$, formado por un espacio de features $\mathcal{X}_\mathcal{T}$ y una distribución de probabilidad sobre este espacio $P(X_T)$, sobre el que se quiere construir un modelo predictivo; y un dominio fuente $\mathcal{D}_\mathcal{S}$ definido de forma análoga con datos *relacionados*.

El objetivo es aprovechar los datos del dominio fuente para mejorar la capacidad predictiva del modelo sobre el dominio objetivo. Esto es deseable en el caso de que los datos del dominio objetivo sean insuficientes por escasez de datos, desbalanceo de clases, etc. Incluso si se cuenta con datos apropiados, añadir datos adicionales puede mejorar el modelo.

La solución no es tan simple como mezclar ambos dominios. Se asume que existen diferencias entre ambos dominios, ya sea en su espacio de features, distribución o ambas. La principal suposición que se realiza en el ámbito del machine learning [4] es que los datos de entrenamiento y los usados en predicción comparten features y función de distribución de probabilidad; si estas condiciones no se cumplen el modelo construido verá su rendimiento perjudicado.

II. Estado del arte

Uno de los enfoques más populares [5] consiste en transformar los datos fuente y objetivo $\mathcal{D}_\mathcal{S}$ y $\mathcal{D}_\mathcal{T}$, proyectándolos sobre un espacio común en el que se minimizan las diferencias en la distribución entre los dominios, mientras que se conserva la estructura original de los datos, usando ideas de teoría de grafos para mantener las relaciones de vecindad entre las instancias.

Ideas similares en cuanto a la proyección de los datos se exponen en [6], pero en este caso no se minimizan explícitamente las diferencias de distribución sino que se sigue un criterio heredado de *manifold learning*, que implica la reconstrucción de los datos transformados a partir de sus vecinos más cercanos.

Por otro lado, existen procedimientos más simples [7] basados simplemente en ajustar las covarianzas de los dominios mediante un *blanqueado* y posterior coloración. Igualar las distribuciones de los datos es una idea que sigue presente, pero se aborda el problema desde la perspectiva de otras propiedades estadísticas.

Existen procedimientos basados en el uso de técnicas de *deep learning*. En concreto, en [8] se propone usar autoencoders, entrenados sobre el dominio fuente y objetivo para extraer nuevas features que contienen información común a ambos dominios y que se pueden usar para entrenar otro modelo cualquiera.

También existe una aplicación de boosting [9] a este ámbito que aprovecha el esquema de pesos e introduce asimetría en la actualización de estos para dar más importancia a instancias del dominio objetivo y descartar las del dominio fuente que perjudican el rendimiento de los clasificadores.

Al margen de las diferencias en la ejecución, una de las cuestiones más importantes para la aplicación de métodos de transfer learning son los requerimientos impuestos en cuanto a la caracterización de los dominios.

Muchos métodos [7], [6], [8] consideran que el espacio de features de fuente y objetivo es el mismo. Entre la literatura reciente, solo [5] parte de la hipótesis de que puede haber disparidad. Esto supone una limitación porque encontrarse con conjuntos de datos caracterizados de la misma forma es poco

común, así que para alinear fuente y objetivo es probable que haya que descartar features presentes en uno y no en otro, lo que conlleva una pérdida de información.

Por otro lado, la presencia de etiquetas también supone un factor limitante. Las técnicas más flexibles [8], [7] no requieren de etiquetas en ninguno de los dominios, basándose únicamente en las propias features. Por una parte, esto es positivo porque permite hacer uso de conjuntos de datos masivos no etiquetados. Por otra parte, esto exige al problema planteado una relación clara entre espacios de features y distribuciones de probabilidad de los datos fuente y objetivo, y que no se puede determinar a priori. En general, asumiremos como caso de referencia que existen etiquetas en el dominio fuente [10] para facilitar la transformación de los datos.

En el caso del dominio objetivo, no se suele asumir la presencia de etiquetas. Solo [9] y de forma opcional [5] consideran su uso explícito. Esto se plantea como situación experimental, pero en general se van a necesitar aunque solo sea con propósito de testear la eficacia del método.

III. Trabajo futuro y conclusiones

Las técnicas de transfer learning han encontrado aplicaciones, principalmente, en el campo del procesado de imágenes apoyado sobre redes neuronales de convolución. Existen aplicaciones de estos estudios al campo de la ciberseguridad como [11, 12]. En estos artículos se extraen capas preentrenadas de una red neuronal que se fijan para el entrenamiento sobre los datos del problema. Existen otros procedimientos basados en el uso de redes generativas adversariales [13] que realizan *data augmentation* para entrenar una red neuronal sobre instancias originales y otras generadas sintéticamente. Estos métodos quedan fuera del alcance de este trabajo, que pretende realizar una evaluación de otras técnicas más genéricas, que son aplicables a otros modelos como por ejemplo árboles de decisión.

Todos los estudios realizados sobre transfer learning proporcionan el pseudo-código de los algoritmos desarrollados o utilizados. Sin embargo, la implementación de éstos no suele ser de libre acceso, o si lo es suele darse en lenguajes que requieren de una licencia, principalmente MATLAB. Es por ello que la primera fase del presente estudio ha consistido en la implementación de una serie de algoritmos de transfer learning [6], [8], [5], [9], en un lenguaje de código abierto como es Python. En particular, se realizan implementaciones de los métodos descritos en [5–9, 14].

El objetivo final de esta investigación será determinar si la aplicación de estas técnicas de transfer learning a problemas de ciberseguridad puede resultar beneficiosa y en qué medida. El análisis se realizará tanto desde el punto de vista predictivo, incluyendo métricas que complementes a la tasa de acierto (curvas ROC, curvas de aprendizaje, sensibilidad, precisión etc), como en términos computacionales (tiempos de ejecución, recursos consumidos etc).

Agradecimientos

Este estudio se enmarca dentro del proyecto X54 financiado por el Instituto Nacional de Ciberseguridad (INCIBE) y desarrollado por el Instituto de Investigación en Ciencias Aplicadas a la Ciberseguridad (RIASC) de la Universidad de León.

Referencias

[1] A. A. Aburomman and M. B. I. Reaz, "A novel svm-knn-pso ensemble method for intrusion detection system," *Applied Soft Computing*, vol. 38, pp. 360 – 372, 2016.

[2] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, "Madam: Effective and efficient behavior-based android malware detection and prevention," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 83–97, Jan 2018.

[3] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018.

[4] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[5] J. Li, K. Lu, Z. Huang, L. Zhu, and H. Tao Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Transactions on Cybernetics*, vol. PP, pp. 1–12, 04 2018.

[6] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 850–863, Feb 2016.

[7] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 2058–2065.

[8] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 4119–4125.

[9] S. Al-Stouhi and C. K. Reddy, "Transfer learning for class imbalance problems with inadequate data," *Knowledge and Information Systems*, vol. 48, no. 1, pp. 201–228, Jul 2016.

[10] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016.

[11] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. d. Geus, "Malicious software classification using vgg16 deep neural network's bottleneck features," in *Information Technology - New Generations*, S. Latifi, Ed. Cham: Springer International Publishing, 2018, pp. 51–59.

[12] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. de Geus, "Malicious software classification using transfer learning of resnet-50 deep neural network," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2017, pp. 1011–1014.

[13] M. . Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 469–477.

[14] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, Sep. 2017.

Análisis de las Técnicas de Detección Automática de Pornografía en Vídeos

Jenny Alexandra Cifuentes, Esteban Alejandro Armas Vega, Ana Lucila Sandoval Orozco,
Luis Javier García Villalba*
Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain
Email: jacifuentesq@gmail.com, esarmas@ucm.es {asandoval, javiergv}@fdi.ucm.es

***Resumen*—El análisis forense digital ha surgido como una disciplina para enfrentar diversos tipos de delitos informáticos y cibernéticos. En particular, teniendo en cuenta el aumento del contenido pornográfico no restringido en Internet y la difusión de casos de distribución de material de abuso sexual infantil, hay una creciente necesidad de herramientas informáticas eficientes para la detección y/o el bloqueo automático de contenido pornográfico. El objetivo de este estudio es revisar las diferentes estrategias disponibles en la literatura para la detección de pornografía, específicamente en vídeos, e identificar brechas de investigación. Este trabajo muestra que las técnicas basadas en aprendizaje profundo detectan vídeos pornográficos con mayor precisión que otras estrategias de detección convencionales. La precisión de las estrategias reportadas en este trabajo depende de las técnicas de extracción de características, la arquitectura y los algoritmos finales de clasificación. Finalmente, se detallan algunas áreas complementarias de investigación en la detección de vídeos pornográficos.**

***Index Terms*—Detección de Pornografía, Clasificación de Vídeos, Investigación Forense Digital, Aprendizaje Profundo, Características de Movimiento**

Tipo de contribución: *Investigación original*

I. Introducción

Debido al rápido crecimiento en el intercambio de contenido multimedia en las redes sociales y dispositivos móviles, la cantidad de contenido inapropiado o ilegal en Internet también ha aumentado sustancialmente. Es así como la pornografía infantil, por ejemplo, se ha convertido en un tema común de preocupación debido al hecho de que este mercado podría fomentar el incremento de material relacionado con abuso sexual [1] y es un claro ataque a la dignidad de los niños presentándolos como objetos sexuales [2]. De esta manera, la detección automática de pornografía ha recibido una atención considerable en las medidas asociadas al cumplimiento de las leyes y en diversas actividades forenses. Estas herramientas computacionales no solo podrían evitar la carga en servidores o el acceso a material no deseado para usuarios (en el caso de menores) o ubicaciones particulares (por ejemplo en instituciones de educación infantil o lugares públicos), sino que también el análisis eficiente de material pornográfico en las escenas del crimen podría llevar a la detención inmediata de posibles delincuentes. De esta manera, una vez que se realiza el proceso de detección de pornografía, se pueden implementar estrategias adicionales para llevar a cabo el reconocimiento de rostro y edad, la identificación del modelo y la ubicación de la cámara y la detección de la manipulación de la imagen, entre otras, lo que podría ayudar a filtrar vídeos y obtener diversos tipos de evidencias durante una investigación.

Con el fin de abordar este problema, se han desarrollado soluciones comerciales para regular el acceso a este tipo de contenidos ([3], [4]). Estos productos ofrecen una metodología de filtrado de información mediante el uso de listas blancas y negras, las cuales se basan en la información obtenida de los metadatos. Sin embargo, estas estrategias no han sido efectivas porque cierto contenido inapropiado puede adjuntarse maliciosamente a textos inofensivos, lo que hace que estas etiquetas de texto vinculadas no sean suficientes para realizar el análisis [5]. En consecuencia, el procesamiento de la información visual se convierte en una característica fundamental para obtener una clasificación confiable de los contenidos pornográficos.

Los primeros enfoques desarrollados para la detección de vídeos pornográficos incorporan principalmente la segmentación de la región del color de la piel para caracterizar la desnudez ([6], [7], [8], [9], [10]). En general, este tipo de soluciones toman en cuenta esa información para distinguir el conjunto de píxeles y las distribuciones espaciales que describen a las personas desnudas. Sin embargo, la exposición de la piel en sí no es un indicador confiable, ya que grandes áreas de la piel no son explícitamente pornográficas (por ejemplo, con el uso de trajes de baño e imágenes relacionadas con deportes) que podrían causar muchos falsos positivos. Además, mediante este método no es posible analizar imágenes en escala de grises.

A diferencia de las técnicas enfocadas en la detección de la piel, otras estrategias alternativas se centran en los descriptores de imagen para realizar la tarea de reconocimiento. En estos enfoques, se extraen las características locales y, en una etapa de descripción intermedia, se cuantifican por medio de un libro de códigos ('*codebook*'), lo cual define los modelos de Bolsas de Palabras (*Bag of Words* - BoW) ([11], [12], [13]). Aunque estos métodos han proporcionado resultados muy prometedores en este tema de investigación, los descriptores locales calculados durante el análisis requieren una alta carga computacional y conducen a vectores de dimensiones considerables. Además, los enfoques son muy sensibles a la elección de las palabras clave, el tamaño del libro de códigos y los algoritmos de agrupación y codificación, lo que resulta en el ajuste de una cantidad significativa de

parámetros, necesarios para lograr un buen desempeño. La mayoría de trabajos de investigación generalmente extienden las soluciones, que han sido aplicadas a imágenes a contenidos de vídeo, mediante el análisis de fotogramas individuales y de una implementación final de un umbral a cada una de las muestras [7]. Sin embargo, la información espacio-temporal disponible en los vídeos puede ofrecer características adicionales que mejoren la precisión en la clasificación.

Por tanto, se han desarrollado métodos de detección de pornografía multimodal (información visual y de movimiento) con el fin de buscar soluciones más efectivas y eficientes. Entre estas estrategias, se han propuesto Trayectorias densas [14], Puntos Temporales de Interés [15], Características Temporales Robustas [16]. De hecho, en [16] se demostró que este tipo de análisis multimodal logra una mayor precisión en la detección de pornografía en vídeos. Sin embargo, a pesar de que estas estrategias tienen un porcentaje de falsos positivos más baja en comparación con otros trabajos, aún no pueden detectar eventos como la masturbación, donde personas vestidas podrían realizar una acción sexual con movimientos que pueden llegar a ser estáticos [17].

Por este motivo y considerando el reciente éxito de las estrategias basadas en Aprendizaje Profundo (*'Deep Learning'*), diversos investigadores han propuesto esta metodología para la detección de pornografía en vídeos. Es así como se han presentado arquitecturas de Redes Neuronales Convolucionales (CNN) y Redes Neuronales Recurrentes (RNN) orientadas a brindar soluciones particularmente en este campo ([18], [19], [20], [21]). Los resultados asociados a estos trabajos han demostrado que las arquitecturas basadas en Aprendizaje Profundo pueden aumentar considerablemente las tasas de clasificación de pornografía en vídeos.

El objetivo principal de este estudio consiste en revisar las técnicas propuestas en la literatura para la detección de pornografía en vídeos, con el fin de identificar posibles trabajos de investigación. Diversas evaluaciones comparativas entre algunos de los métodos presentados en este trabajo han sido reportadas en conjunto para imágenes y vídeos para estrategias particulares ([12], [20], [22]). Sin embargo, en este artículo se abordan las estrategias enfocadas al análisis específico en vídeos, describiendo todos los trabajos relevantes que hasta el momento, a nuestro saber, han sido propuestos en este campo junto con sus arquitecturas y los algoritmos respectivos de extracción de fotogramas y clasificación final del contenido multimedia. Este documento está organizado como sigue: los enfoques basados en fotogramas claves, descriptores de imagen, análisis de movimiento y Aprendizaje Profundo se describen en las Secciones II, III, IV y V, respectivamente. Por cada tipo de enfoque se comparan las técnicas analizadas. Finalmente, las conclusiones y el trabajo futuro se presentan en la Sección VI.

II. Técnicas basadas en el Análisis de Características Visuales

La mayoría de los trabajos sobre detección de pornografía han examinado el reconocimiento de la piel humana. En particular, las estrategias basadas en este enfoque se fundamentan en la premisa de que los contenidos pornográficos a color incluyen una gran parte de áreas asociadas a la piel. De esta manera, en este caso particular, la clasificación se realiza en función de características de bajo nivel, como color, forma o patrones de distribución generales. Esta metodología ha sido ampliamente explorada en aplicaciones sobre imágenes ([23], [24], [25], [26], [27]).

Sobre esta base, los métodos tradicionales implementados para clasificar vídeos pornográficos incluyen la extracción de estas características visuales directamente de los fotogramas clave. En esta línea, Wang *et al.* segmentaron el flujo de vídeo en tomas y fotogramas clave, que posteriormente utilizaron para la extracción del cuerpo humano desnudo [6]. Específicamente, los autores tomaron en cuenta la propiedad convergente de la distribución del color de la piel para proponer el modelo gaussiano asociado a una distribución particular en el espacio de color YCbCr, clasificando finalmente los píxeles de la piel mediante un método bayesiano. Los resultados experimentales mostraron una precisión en la detección de vídeos pornográficos del 89.2 % y del 90.3 % para vídeos de toma larga y corta, respectivamente.

Lee *et al.*, por otra parte, presentaron un enfoque que contiene dos características visuales de extracción y, con base en esos resultados, llevaron a cabo un proceso de aprendizaje por medio de Máquinas de Vectores Soporte (Support Vector Machines - SVM) [9]. La primera variable se basa en una decisión basada en un solo fotograma, en la cual se calcula la probabilidad gaussiana de que un píxel tenga el color característico de la piel. La segunda es una variable de decisión basada en un Grupo de Fotogramas (Group of Frames - GoF) la cual caracteriza la representación conjunta de características basadas en color para múltiples fotogramas. El cálculo conlleva a la cuantización del espacio HSV en 256 celdas, adicionadas a partir de los múltiples fotogramas de un vídeo, y al promedio de los valores de la celda correspondiente. En estos experimentos, se analizaron 1200 archivos de vídeo con resultados que involucran precisiones de 100 % y 96.6 % durante el entrenamiento y la validación, respectivamente.

Monteiro y Polastro realizaron la detección de pornografía infantil en vídeos mediante una herramienta orientada al reconocimiento en imágenes llamada NuDetective y un método alternativo para la segmentación de fotogramas [8]. La herramienta forense NuDetective [28] puede detectar automáticamente la desnudez en imágenes con base en una relación de umbral del espacio de color RGB, lo cual significa satisfacer la relación descrita en la Ecuación 1. Así mismo, la etapa de clasificación se lleva a cabo utilizando el algoritmo desarrollado en [29], el cual define un umbral para el área cubierta por regiones con piel. La segmentación de vídeo para obtener los fotogramas que permitirán realizar el análisis, se enfoca en una función logarítmica que representa el muestreo adaptativo del vídeo. El índice de precisión para un análisis de 149 vídeos es 85.9 % en un lapso de 170 segundos, lo cual es 0.2 % más preciso y 44.8 % más rápido en comparación con los resultados preliminares reportados en [7], en los cuales se utiliza un método de muestreo uniforme para la extracción de los fotogramas.

$$\begin{aligned} &R > 95 \text{ and } G > 40 \text{ and } B > 20 \text{ and} \\ &\text{máx}\{R,G,B\} - \text{mín}\{R,G,B\} > 15 \text{ and} \\ &|R-G| > 15 \text{ and } R > G \text{ and } R > B \end{aligned} \tag{1}$$

Tabla I
ARTÍCULOS REPRESENTATIVOS BASADOS EN EL ANÁLISIS DE CARACTERÍSTICAS VISUALES

Ref.	Tamaño del Conjunto de Datos	Algoritmo de Extracción de Fotogramas	Algoritmo de Reconocimiento	Características	Algoritmo de Clasificación	Medidas de Evaluación
[6]	20 vídeos largos 112 vídeos cortos	Diferencia de Color [31]	Modelo Gausiano en YCbCr	Color de la Piel Textura de la piel Morfología	Bayesiano	Precisión: Largo: 89.2 %, Corto: 90.3 % Recall: Largo: 92.5 %, Corto: 91.5 %
[9]	1200 vídeos	Muestreo Uniforme	Individual: Gausiano Global: Discriminante de color HSV	Color de la Piel	SVM	Precisión-validación: 96.6 % Recall-validación: 86.19 %
[7]	149 vídeos	Muestreo Uniforme	Umbral RGB	Color de la Piel	Umbral de Área con Piel [29]	Precisión: 85.7 % Recall: 84.9 %
[8]	149 vídeos	Función Logarítmica	Umbral RGB	Color de Piel	Umbral de Área con Piel [29]	Precisión: 85.9 % Recall: 87.3 %
[10]	986 Imágenes 253 Vídeos	Muestreo Uniforme [30]	Umbral YCbCr Filtro Gausiano Pasa-Bajo	Color de la Piel Textura de la Piel	Umbral de Áreas con Piel [29]	Precisión: 90.33 %

Más recientemente, García *et al.* desarrollaron una aplicación para detectar vídeos pornográficos también basados en el análisis de píxeles del color de la piel, esta vez en el espacio YCbCr [10]. Así mismo, la extracción de fotogramas se realizó utilizando el kit de herramientas de vídeo PHP [30]. Con base en estos resultados, los fotogramas de vídeo se descompusieron en píxeles, la detección del nivel de piel se ajustó mediante un umbral y se calcularon los porcentajes de piel y no-piel en relación al fotograma analizado. Finalmente, la aplicación de filtrado de desnudos clasificó el archivo como desnudo si cumplía con las condiciones de umbral de regiones desnudas propuesto en [29]. La aplicación se evaluó en un conjunto de datos de 1.239 archivos multimedia web (Imágenes = 986; Vídeos = 253), obteniendo una precisión del 90.33 % y una exactitud del 80.23 %.

La tabla I muestra un resumen de las propuestas de investigación basadas en el análisis de características visuales. Si bien, algunos de los trabajos expuestos anteriormente tienen muy buenas tasas de precisión, estos enfoques dependen en gran medida del recuento de píxeles de color piel, lo cual genera errores considerables en escenas con muchas personas o con materiales cuyos colores son similares a los de la piel humana. Además, los algoritmos son difíciles de generalizar en función de los cambios de color que pueden tener diferentes grupos étnicos, o producidos por variaciones de luminosidad.

III. TÉCNICAS BASADAS EN EL ANÁLISIS CON DESCRIPTORES DE IMAGEN

Teniendo en cuenta las desventajas inherentes asociadas a las características visuales de bajo nivel, una metodología alternativa llamada Bolsa de Palabras (BoW) se ha utilizado considerablemente. La idea fundamental de esta propuesta es minimizar la brecha semántica existente entre las características visuales de bajo nivel (por ejemplo, el color de los píxeles) y los conceptos de alto nivel sobre pornografía. En este caso, una adaptación del modelo BoW, utilizado en la recuperación de información, se implementa por medio de información extraída de histogramas asociados a parches locales. Es así como antes de la etapa de reconocimiento de características, se genera un libro de códigos de palabras visuales utilizando un algoritmo de agrupamiento para cada parche. Sobre la base de estos resultados, el vector de características utilizado en la clasificación es un histograma obtenido con la frecuencia en la que aparece cada palabra visual en la imagen [32]. Esta estrategia se ha implementado en gran medida para clasificar imágenes en diferentes categorías de pornografía ([33], [5], [34], [35]).

La aplicación de esta metodología en vídeos ha sido explorada en algunas investigaciones. En esta línea, Lopes *et al.* propusieron un enfoque basado en BoW mediante el descriptor de color HueSIFT (Hue Scale-Invariant Feature Transform) con el fin de detectar desnudos en vídeos [13]. En este caso, los fotogramas individuales están representados por una Bolsa de Características Visuales (BoVF) y son clasificados por un modelo lineal de SVM. Con base en estos resultados, se implementó un esquema de votación por mayoría para mejorar la clasificación de los segmentos de vídeo. Los resultados experimentales muestran que el algoritmo logra, en el mejor de los casos, una tasa de clasificación del 93.2 %.

Avila *et al.* presentaron dos representaciones de medio nivel basadas en una distribución de distancias entre palabras clave llamadas BOSSA [36] y BossaNova [37]. En estos trabajos, los autores mejoran la representación por medio de un histograma obtenido con base en las distancias calculadas entre los descriptores en el fotograma y los extraídos del libro de códigos. En ambas investigaciones, se aplicó un esquema de votación para la clasificación final de los fotogramas del vídeo. Los experimentos de validación se llevaron a cabo usando el conjunto de datos de Pornography-800 [37] obteniendo una tasa de precisión de 87.1 % y 89.5 % para los estrategias basadas en BOSSA y BossaNova, respectivamente.

De forma similar, Caetano *et al.* [11] proponen una nueva solución, aplicada a vídeos, empleando descriptores binarios locales junto con BossaNova. El enfoque descrito por los autores tiene la ventaja de que es independiente de cualquier detector de piel o de forma para clasificar el contenido pornográfico. Para validar el rendimiento del algoritmo, los autores analizaron el conjunto de datos Pornography-800 [37], que está compuesto por casi 80 horas de 400 vídeos pornográficos y 400 no pornográficos, separados en 16.727 fotogramas claves de vídeo. Las tasas de exactitud en esta investigación alcanzan el 90.9 %. En un trabajo posterior, Caetano *et al.* [12] presentan una extensión del modelo BoW el cual preserva con mayor precisión la información visual. En particular, describen dos enfoques de descripción de vídeo basados en la combinación del descriptor de Vídeo BossaNova (BNVD) y el Descriptor de Vídeo BoW (BoW-VD). Con base en esta propuesta, los autores mejoraron la exactitud de la clasificación agregando la información de las

Tabla II
ARTÍCULOS REPRESENTATIVOS BASADOS EN EL ANÁLISIS CON DESCRIPTORES DE IMAGEN

Referencia	Tamaño del Conjunto de Datos	Algoritmo BoVW	Descriptor de Características	Algoritmo de Clasificación	Medidas de Evaluación
[13]	179 segmentos de vídeo	Estándar	HueSIFT	SVM (Kernel Lineal)	Exactitud: 93.2 %
[36]	800 vídeos	BOSSA	HueSIFT	SVM (Kernel No Lineal)	Exactitud: 87.1 %
[37]	800 vídeos	BossaNova	HueSIFT	SVM (Kernel Lineal)	Exactitud: 89.5 %
[11]	800 vídeos	BossaNova	Descriptores Binarios	SVM (Kernel No Lineal)	Exactitud: 90.9 %
[12]	800 vídeos	BossaNovaVD	Descriptores Binarios	SVM (Kernel No Lineal)	Exactitud: 92.4 %
[32]	3595 vídeos	Estándar	Vectores de Movimiento	SVM(Kernel No Especificado)	Índice de Error: 6.04 %
[38]	3300 vídeos	Estándar	Vectores de Movimiento, Audio Color de la Piel	SVM (Kernel No Lineal) (kernel RBF)	Índice de Error: 5.92 %
[39]	800 vídeos	Estándar	STIP	SVM (Kernel No Lineal)	Exactitud: 91.9 %
[40]	800 vídeos	Estándar	Color STIP	SVM (Kernel Lineal)	Exactitud: 91.0 %
[16]	2000 vídeos web	Vector de Fisher	TRoF	SVM (Kernel Lineal)	Exactitud: 95.0 %

características de medio nivel de todos los fotogramas de vídeo a una sola representación, en lugar de implementar una bolsa individual para cada fotograma de vídeo extraído. Los resultados reportan una exactitud del 92.4 % analizando el mismo conjunto de datos del trabajo anterior.

Relativamente pocos trabajos han incluido características espacio-temporales en los algoritmos de detección de vídeos pornográficos. Esta idea ha sido desarrollada por Jansohn *et al.*, la cual combina un modelo de BoVW con un análisis estadístico de vectores de movimiento MPEG-4 específicamente para realizar la detección en vídeos [32]. La respectiva validación se llevó a cabo con 932 vídeos web para adultos y 2.663 clips con contenido inofensivo adquiridos del portal web YouTube. Los resultados mostraron que la inclusión del análisis de movimiento en la etapa de reconocimiento reduce el índice de error (EER) de 9.9 % a 6.0 %, en comparación con los enfoques tradicionales de BoVW. En un trabajo posterior, Ulges *et al.* propusieron un enfoque multimodal que incluye diferentes características como el color de la piel, los coeficientes discretos de transformación del coseno de diversos parches de la imagen (basados en un modelo de BoVM), histogramas de movimiento e información de audio procesada mediante coeficientes cepstrales [38]. En particular, los autores evaluaron la combinación de todas las características, obteniendo un incremento considerable en la exactitud, reduciendo el error en un 36-56 % en comparación con el mejor sistema uni-modal. El conjunto de datos analizado en esta investigación incluye alrededor de 500 horas de vídeo (1.000 vídeos pornográficos y 2.300 vídeos de YouTube). Si bien el error de igualdad se reduce en este trabajo, el costo computacional involucrado es considerablemente alto.

Como otra alternativa de análisis, Valle et al. aplicaron el descriptor espacio-temporal STIP [15] junto con un modelo estándar de BoVW, implementando un enfoque de muestreo aleatorio para la generación del libro de códigos [39]. En la aplicación en vídeos pornográficos, los experimentos reportados examinaron un conjunto de datos de 77 horas para 800 vídeos con una exactitud general del 91.9 %. Siguiendo una dirección similar, Souza *et al.* evaluaron el desempeño de la familia de STIPs basados en color (HUESTIP, COLORSTIP y HUE-COLORSTIP), usando una estrategia tradicional basada en BoVW [40]. El mejor resultado de exactitud registrado en este trabajo, para el mismo conjunto de datos descrito en [39], fue de 91 % mediante el descriptor COLORSTIP.

Más recientemente, Moreira *et al.* introdujeron un descriptor de puntos de interés espacio-temporales llamado 'Característica Robusta Temporal' (Temporal Robust Features - TRoF) [16], el cual tiene en cuenta la información relevante sobre el movimiento en el análisis de vídeo. Específicamente, los autores agregaron la información local obtenida, al usar la TRoF, en una representación de medio nivel utilizando vectores de Fisher. La validación de esta estrategia se realizó, contrastándola con las soluciones basadas en BoVW descritas anteriormente. De esta manera, se evaluó el rendimiento utilizando el conjunto de datos Pornography-2k, que está compuesto por 2000 vídeos web recopilados de Internet. Los resultados experimentales muestran una exactitud el 95 %, superando a estrategias basadas en BoW consideradas en esta Sección. La tabla II resume la estructura de estos enfoques, mostrando las medidas de desempeño utilizadas para la evaluación de cada estrategia.

Una de las desventajas asociadas a este enfoque incluye la alta complejidad del modelo, debido a la gran cantidad de algoritmos implementados para generar las palabras clave y los numerosos esfuerzos para encontrar los hiper-parámetros adecuados para lograr un buen desempeño de la estrategia propuesta.

IV. ENFOQUES BASADOS EN EL ANÁLISIS DE MOVIMIENTO

La mayoría de las investigaciones sobre detección de vídeos pornográficos se han centrado en los enfoques de segmentación de fotogramas implementados junto con el análisis de las características de bajo o medio nivel extraídas de cada uno de ellos. Sin embargo, los conjuntos de datos de vídeo involucran información adicional (datos de audio y movimiento) que podrían usarse para mejorar la precisión en la etapa de clasificación. En este sentido, Rea *et al.* propusieron utilizar la estimación del color de la piel junto con los patrones periódicos encontrados en las señales de audio en vídeos pornográficos [41]. Específicamente, la periodicidad de audio se calcula a través de la ubicación de los máximos y mínimos en la autocorrelación de la señal de energía correspondiente. En este trabajo, el enfoque propuesto se evaluó en una película de muestra, pero no se validó en un conjunto de datos más amplio, lo que impide la posibilidad de tener resultados de clasificación generales.

Otro trabajo de investigación que examina las señales de audio fue presentado por Zuo *et al.*. En particular, ellos proponen la clasificación de vídeos pornográficos utilizando un modelo de mezcla gaussiana (GMM), de una clase, que analiza un vector de características de 13 dimensiones (12

Tabla III
ARTÍCULOS REPRESENTATIVOS CON RELACIÓN AL ANÁLISIS DE MOVIMIENTO

Ref.	Tamaño del Conjunto de Datos	Características Analizadas	Algoritmo de Clasificación	Medidas de Evaluación
[41]	1 Muestra de Vídeo	Periodicidad del Audio	Umbral de Periodicidad	No especificado
[42]	889 vídeos	12 MFCC + Término de Energía Contorno del Cuerpo	GMM Clasificador Bayes	Precisión: 92.3 % Recall: 98.3 %
[43]	3255 vídeos	Vectores de Movimiento Momentos de Forma	Coincidencia de Forma [44]	Exactitud: 96.5 %
[45]	750 vídeos	Vectores de Movimiento	Umbral de Estimación Espectral	No especificado
[46]	100 vídeos	Vectores de Movimiento (Fuerza y dirección)	2 Umbrales de Características de movimiento	Exactitud: 90.0 %
[47]	4000 vídeos	Características de Periodicidad y Movimiento	SVM (Kernel Lineal)	Exactitud: 95.44 %
[48]	19813 escenas	Magnitud y frequencia de movimiento periódico Vector de Color de Piel	SVM (Kernel RBF)	No especificado

Coeficientes Cepstrales en las Frecuencias de Mel -MFCC- más un término de energía) obtenido de la señal de audio de los vídeos [42]. Además, se incluye un algoritmo de reconocimiento de imágenes pornográficas basadas en el contorno generalizado y un clasificador de Bayes para completar el reconocimiento mediante el uso de datos de audio y vídeo. En estos experimentos, 352 películas pornográficas y 537 películas regulares se analizaron con una precisión del 92.3 %. Durante este análisis, la granularidad de los fotogramas de audio fue muy pequeña y la componente de periodicidad no se tuvo en cuenta.

Kim *et al.*, por otra parte, realizan la detección de movimientos globales con base en los vectores de movimiento asociados a cada fotograma individual. En caso de que se reconozca un movimiento local, el algoritmo realiza la detección de regiones de la piel mediante descriptores de momentos invariantes [43]. Finalmente, la clasificación se basa en el algoritmo de coincidencia de formas descrito en [44]. Durante el proceso de validación, se analizaron 2275 vídeos convencionales y 980 vídeos obscenos con una tasa final de exactitud del 96.5 %.

Un enfoque alternativo fue propuesto por Endeshaw *et al.* [45]. En este trabajo, los autores presentan un algoritmo de detección de periodicidad de movimiento con el fin de identificar patrones repetitivos representativos en vídeos pornográficos. Este análisis se realiza con base en el cálculo de espectros del movimiento dominante, en una banda de frecuencia específica, durante intervalos de 16 segundos de la secuencia de vídeo. Los resultados experimentales solo se presentan de forma indicativa en función de la cantidad limitada de material de vídeo utilizado para la validación.

En una línea de investigación similar, Zhiyi *et al.* realizaron la extracción del vector de movimiento del flujo de vídeo MPEG y lo suavizaron mediante un filtro de mediana [46]. El método calcula la fuerza y la dirección del respectivo vector de movimiento y la detección de pornografía se realiza en base a una variable umbral. En este trabajo, se clasificaron 30 vídeos pornográficos y 70 vídeos regulares con una exactitud general del 90.0 %. Este algoritmo tiene el problema de no poder detectar vídeos pornográficos que involucren movimientos globales o en los cuales, la componente de movimiento sea muy escasa.

Behrad *et al.* propusieron la localización de la sección de color de piel más grande y, con base en estos resultados, realizaron la extracción de 6 características basadas en su movimiento, utilizando la transformada de Fourier de los coeficientes de autocorrelación entre fotogramas [47]. La clasificación de vídeos pornográficos se llevó a cabo utilizando un clasificador SVM con una tasa de reconocimiento promedio de 95.44 % para 2.000 episodios de vídeos obscenos y 2.000 episodios de vídeos regulares.

Alternativamente, Jung *et al.* usaron los patrones de movimiento espacio-temporal para realizar la respectiva detección de vídeos pornográficos [48]. En este caso, las características extraídas se basan en la magnitud y la frecuencia del movimiento periódico y el vector de color de la piel, obtenidos de los histogramas de matiz y saturación. Durante los experimentos, se utilizaron 1500 segmentos de vídeo (500 de vídeos para adultos y 1000 de documentales) como conjunto de entrenamiento. La prueba se realizó en 18313 escenas (1103 escenas con al menos una unidad de vídeo de una escena adulta). En comparación con el enfoque propuesto, los resultados mostraron que los métodos descritos anteriormente en ([47], [38]) son menos efectivos en el análisis de escenas pornográficas que incluyen una pequeña porción de color de piel en cada fotograma. Además, la metodología propuesta en este trabajo también es más precisa en ausencia del color de piel, en contraste con los métodos que no incluyen específicamente el color de la piel como un componente de clasificación, como por ejemplo los resultados descritos en [45]. Asimismo, durante esta evaluación, se evaluaron los tiempos de procesamiento de los trabajos desarrollados en ([47], [38], [45]) y el enfoque propuesto. Los resultados muestran que la estrategia desarrollada por Endeshaw *et al.* [45] es más eficiente. La Tabla III presenta una descripción general de las soluciones basadas en el análisis de movimiento encontradas para el problema de detección de vídeos pornográficos.

V. TÉCNICAS BASADAS EN APRENDIZAJE PROFUNDO

Teniendo en cuenta la dificultad para definir umbrales adecuados en los algoritmos de detección basados en la región de piel y el movimiento, la falta de características adecuadas que describan el movimiento y el reciente éxito generalizado de los métodos de Aprendizaje Profundo, estos últimos han sido considerados como una solución relevante en el campo de la detección de pornografía. En particular, las Redes Neuronales Convolucionales (CNN) se han aplicado satisfactoriamente al reconocimiento de imágenes pornográficas ([49], [50], [51]).

En esta línea de investigación, Moustafa presentó el primer intento de evaluar el uso de Redes Neuronales Profundas en el problema de detección de vídeos pornográficos [18]. En este estudio, el autor propone una combinación de dos arqui-

Tabla IV
ARTÍCULOS REPRESENTATIVOS RELACIONADOS CON ESTRATEGIAS DE APRENDIZAJE PROFUNDO

Referencia	Tamaño del Conjunto de Datos	Arquitectura de la CNN	Algoritmo de Clasificación	Medidas de Evaluación
[18]	800 vídeos [37]	Fusión (CNN AlexNet y CNN GoogleNet)	Votación por Mayoría	Exactitud: 94.1 %
[20]	800 vídeos [37] 2000 vídeos [16]	CNN basada en GoogleNet	SVM (Kernel Lineal)	Exactitud: 97.9 % Exactitud: 96.4 %
[21]	800 vídeos [37]	Fusión (CNN ResNet y CNN GoogleNet)	LSTM-RNN	Exactitud: 95.6 %
[19]	800 vídeos [37]	CNN VGG-C3D CNN ResNet R(2+1)D CNN	SVM (Kernel Lineal) Clasificador SoftMax	Exactitud: 95.1 % Exactitud: 91.8 %

tecturas CNN diferentes (AlexNet [52] y GoogLeNet [53]) para clasificar pornografía a partir de fotogramas de vídeo convencionales. Los experimentos se llevaron a cabo sobre el conjunto de datos desarrollado en [37], el cual contiene 400 vídeos para adultos y 400 regulares. Para clasificar las secuencias de vídeo, los fotogramas clave se extraen y se prueban individualmente, y el etiquetado final se basa en un conteo de mayoría de votos. El enfoque propuesto obtuvo una exactitud del 94.1 %, superando los resultados obtenidos en [37] y [11]. Sin embargo, este trabajo no exploró el uso de la información de movimiento.

Perez *et al.* proponen una estrategia basada en la CNN GoogleNet [53], la cual se empleó para todos los tipos de datos: estáticos (fotogramas en bruto) y de movimiento (flujo óptico y vectores de movimiento) [20]. La configuración experimental incluyó los conjuntos de datos pornography-800 y pornography-2k detallados en [37] y [16], obteniendo exactitudes generales del 97.9 % y del 96.4 %, respectivamente. Estas tasas de reconocimiento son más altas comparadas con los resultados presentados anteriormente en las Secciones II, III, IV y V.

Recientemente, Wehrmann *et al.* presentaron un enfoque alternativo compuesto por combinaciones de las arquitecturas GoogleNet [53] y ResNet [54] para la extracción de características y de una Red Neural Recurrente (RNN), basada en unidades LSTM (Long Short Term Memory), para la clasificación de pornografía en vídeos [21]. El enfoque propuesto se llama ACORDE (Adult Content Recognition with Deep Neural Networks) y se evaluó en el conjunto de datos pornography-800 [37], con una tasa de exactitud del 95.6 % para la mejor configuración (ResNet-101).

Da Silva y Marana evaluaron dos CNN espacio-temporales 3D: la CNN VGG-C3D [55] y la CNN ResNet R(2+1)D [56] para la detección de pornografía en vídeos [19]. La arquitectura propuesta para la CNN VGG-C3D incluye 8 capas convolucionales, 5 capas de agrupación, 2 capas completamente conectadas y una capa de salida softmax. Los filtros de convolución 3D asociados a esta CNN son de dimensión $3 \times 3 \times 3$ con pasos de $1 \times 1 \times 1$. La CNN ResNeT R(2+1)D, por otro lado, factoriza la convolución 3D en dos operaciones individuales y sucesivas: una convolución espacial 2D y una convolución temporal 1D. Esta arquitectura permite la adición de funciones de activación de rectificación no lineal (ReLU) entre las convoluciones 2D y 1D, lo cual aumenta el número de no linealidades en el modelo. La evaluación de estas alternativas se desarrolló en el onjunto de datos pornography-800 con exactitudes del 95.1 % y del 91.8 %, para las CNN VGG-C3D y ResNetR(2+1)D, respectivamente. En general, para este tema de investigación, se han desarrollado cuatro investigaciones centradas en diversas arquitecturas de CNN y bajo diferentes combinaciones de clasificadores supervisados, como se muestra en la Tabla IV.

VI. CONCLUSIONES

En este trabajo de revisión se describen los algoritmos de detección de pornografía utilizados específicamente en vídeos. Teniendo en cuenta la creciente cantidad de vídeos pornográficos encontrados en Internet y las tasas alarmantes de casos de explotación pornográfica infantil, la identificación eficiente de contenidos pornográficos presentes en diversas bases de datos multimedia se ha convertido en un tema vital en el análisis forense. En este contexto, se han propuesto diferentes estrategias para abordar este problema. Es así como se han presentado algoritmos basados en características de bajo y medio nivel (visuales y descriptores de imagen), en el análisis de movimiento y en estrategias relacionadas con Aprendizaje Profundo. Entre ellas, los métodos de Aprendizaje Profundo detectan vídeos pornográficos con mayor precisión que otros modelos convencionales. En particular, la investigación desarrollada por [20] muestra los mejores resultados de precisión hasta ahora para una solución basada en redes profundas, en el campo de la detección de vídeos pornográficos. Es importante tener en cuenta que los conjuntos de datos estándar desarrollados en este campo (pornography-800 y pornography-2k) han facilitado la comparación entre las diversas propuestas de investigación.

Teniendo en cuenta el éxito alcanzado por las técnicas basadas en Aprendizaje Profundo, se espera el desarrollo de nuevos estudios que tengan en cuenta las combinaciones de arquitecturas basadas en redes profundas espacio-temporales supervisadas y no supervisadas con el fin de estudiar la configuración más eficiente en el contexto particular de la pornografía. Así mismo, se espera que las señales de audio y las características de flujo óptico sean evaluadas en conjunto con las visuales sobre estos enfoques con el fin de determinar el impacto en las tasas respectivas de exactitud y precisión. Finalmente, el análisis de redes profundas clásicas, como los autoencoders y las máquinas de Boltzmann restringidas serían de gran interés en este campo de aplicación.

AGRADECIMIENTOS

Este proyecto ha sido financiado por el programa de investigación e innovación de la Unión Europea Horizonte 2020 en virtud del acuerdo de subvención No 700326. Sitio web: http://ramses2020.eu

REFERENCIAS

[1] S. Ost, "Children at risk: Legal and societal perceptions of the potential threat that the possession of child pornography poses to society," *Journal of Law and Society*, vol. 29, no. 3, pp. 436–460, 2002.

[2] K. Warner, "Sentencing for child pornography," *Australian Law Journal*, vol. 84, no. 6, pp. 384–395, 2010.

[3] TapTap Software, "Media Detective," www.mediadetective.com, February 2019.

[4] Hyperdyne Software, "Snitch Plus," www.hyperdynesoftware.com, February 2019.

[5] A. P. Lopes, S. E. de Avila, A. N. Peixoto, R. S. Oliveira, and A. d. A. Araújo, "A bag-of-features approach based on hue-sift descriptor for nude detection," in *Proceedings of the 17th European Signal Processing Conference*. IEEE, 2009, pp. 1552–1556.

[6] D. Wang, M. Zhu, X. Yuan, and H. Qian, "Identification and annotation of erotic film based on content analysis," in *Proceedings of IV Electronic Imaging and Multimedia Technology*, vol. 56. International Society for Optics and Photonics, 2005, pp. 88–95.

[7] M. de Castro Polastro and P. M. da Silva Eleuterio, "A statistical approach for identifying videos of child pornography at crime scenes," in *Proceedings of the Seventh International Conference on Availability, Reliability and Security*. IEEE, 2012, pp. 604–612.

[8] P. M. da Silva Eleuterio, M. de Castro Polastro, and B. F. Police, "An adaptive sampling strategy for automatic detection of child pornographic videos," in *Proceedings of the Seventh International Conference on Forensic Computer Science, Brasilia, DF, Brazil*, 2012, pp. 12–19.

[9] H. Lee, S. Lee, and T. Nam, "Implementation of high performance objectionable video classification system," in *Proceedings of 8th International Conference Advanced Communication Technology*, vol. 2. IEEE, 2006, pp. 959–961.

[10] M. B. Garcia, T. F. Revano, B. G. M. Habal, J. O. Contreras, and J. B. R. Enriquez, "A Pornographic Image and Video Filtering Application Using Optimized Nudity Recognition and Detection Algorithm," in *Proceedings of 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, 2018, pp. 1–5.

[11] C. Caetano, S. Avila, S. Guimaraes, and A. d. A. Araújo, "Pornography detection using bossanova video descriptor," in *Proceedings of 22nd European Signal Processing Conference*. IEEE, 2014, pp. 1681–1685.

[12] C. Caetano, S. Avila, W. R. Schwartz, S. J. F. Guimarães, and A. d. A. Araújo, "A mid-level video representation based on binary descriptors: A case study for pornography detection," *Neurocomputing*, vol. 213, pp. 102–114, 2016.

[13] A. P. B. Lopes, S. E. de Avila, A. N. Peixoto, R. S. Oliveira, M. d. M. Coelho, and A. d. A. Araújo, "Nude detection in video using bag-of-visual-features," in *Proceedings of XXII Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2009, pp. 224–231.

[14] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[15] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[16] D. Moreira, S. Avila, M. Perez, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Pornography classification: The hidden clues in video space–time," *Forensic science international*, vol. 268, pp. 46–61, 2016.

[17] K. H. Song and Y.-S. Kim, "Pornographic Video Detection Scheme Using Multimodal Features," *Journal of Engineering and Applied Sciences*, vol. 13, no. 5, pp. 1174–1182, 2018.

[18] M. N. Moustafa, "Applying deep learning to classify pornographic images and videos," *arXiv preprint arXiv:1511.08899*, 2015.

[19] M. V. da Silva and A. N. Marana, "Spatiotemporal CNNs for Pornography Detection in Videos," in *Proceedings of Iberoamerican Congress on Pattern Recognition*. Springer, 2018, pp. 547–555.

[20] M. Perez, S. Avila, D. Moreira, D. Moraes, V. Testoni, E. Valle, S. Goldenstein, and A. Rocha, "Video pornography detection through deep learning techniques and motion information," *Neurocomputing*, vol. 230, pp. 279–293, 2017.

[21] J. Wehrmann, G. S. Simões, R. C. Barros, and V. F. Cavalcante, "Adult content detection in videos with convolutional and recurrent neural networks," *Neurocomputing*, vol. 272, pp. 432–438, 2018.

[22] A. Gangwar, E. Fidalgo, E. Alegre, and V. González-Castro, "Pornography and child sexual abuse detection in image and video: A comparative evaluation," in *Proceedings of the 8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017)*. IET, 2017, pp. 37–42.

[23] M. M. Fleck, D. A. Forsyth, and C. Bregler, "Finding naked people," in *Proceedings of European Conference on Computer Vision*. Springer, 1996, pp. 593–602.

[24] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[25] A. Carlsson, A. Eriksson, and M. Isik, "Automatic detection of images containing nudity," *Master thesis in intelligent systems Design*, 2008.

[26] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern recognition*, vol. 40, no. 3, pp. 1106–1122, 2007.

[27] C. Platzer, M. Stuetz, and M. Lindorfer, "Skin sheriff: a machine learning solution for detecting explicit images," in *Proceedings of the 2nd international workshop on Security and forensics in communication systems*. ACM, 2014, pp. 45–56.

[28] M. de_Castro Polastro and P. M. da Silva Eleuterio, "Nudetective: A forensic tool to help combat child pornography through automatic nudity detection," in *Proceedings of Workshop on Database and Expert Systems Applications*. IEEE, 2010, pp. 349–353.

[29] R. Ap-Apid, "An algorithm for nudity detection," in *Proceedings of the 5th Philippine Computing Science Congress*, 2005, pp. 201–205.

[30] O. Lillie, "PHPVideoToolkit," https://github.com/buggedcom/phpvideotoolkit-v2, 2017.

[31] M.-L. Zhu, "Video stream segmentation method based on video page," *JOURNAL OF COMPUTER AIDED DESIGN AND COMPUTER GRAPHICS*, vol. 12, no. 8, pp. 585–589, 2000.

[32] C. Jansohn, A. Ulges, and T. M. Breuel, "Detecting pornographic video content by combining image features with motion information," in *Proceedings of the 17th ACM international conference on Multimedia*. ACM, 2009, pp. 601–604.

[33] T. Deselaers, L. Pimenidis, and H. Ney, "Bag-of-visual-words models for adult image classification and filtering," in *Proceedings of 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[34] A. Ulges and A. Stahl, "Automatic detection of child pornography using color visual words," in *Proceedings of IEEE International Conference on Multimedia and Expo*. IEEE, 2011, pp. 1–6.

[35] J. Zhang, L. Sui, L. Zhuo, Z. Li, and Y. Yang, "An approach of bag-of-words based on visual attention model for pornographic images recognition in compressed domain," *Neurocomputing*, vol. 110, pp. 145–152, 2013.

[36] S. Avila, N. Thome, M. Cord, E. Valle, and A. d. A. Araújo, "Bossa: Extended bow formalism for image classification," in *2011 18th IEEE International Conference on Image Processing*. IEEE, 2011, pp. 2909–2912.

[37] S. Avila, N. Thome, M. Cord, E. Valle, and A. D. A. AraúJo, "Pooling in image representation: The visual codeword point of view," *Computer Vision and Image Understanding*, vol. 117, no. 5, pp. 453–465, 2013.

[38] A. Ulges, C. Schulze, D. Borth, and A. Stahl, "Pornography detection in video benefits (a lot) from a multi-modal approach," in *Proceedings of the 2012 ACM international workshop on Audio and multimedia methods for large-scale video analysis*. ACM, 2012, pp. 21–26.

[39] E. Valle, S. de Avila, A. d. Luz Jr, F. de Souza, M. Coelho, and A. Araújo, "Content-based filtering for video sharing social networks," *arXiv preprint arXiv:1101.2427*, 2011.

[40] F. Souza, E. Valle, G. Cámara-Chávez, and A. Araújo, "An evaluation on color invariant based local spatiotemporal features for action recognition," *IEEE SIBGRAPI*, 2012.

[41] N. Rea, G. Lacey, R. Dahyotit, and R. Dahyot, "Multimodal periodicity analysis for illicit content detection in videos," in *Proceedings of the 3rd European Conference on Visual Media Production (CVMP 2006)-Part of the 2nd Multimedia Conference 2006*. IET, 2006, pp. 106–114.

[42] H. Zuo, O. Wu, W. Hu, and B. Xu, "Recognition of blue movies by fusion of audio and video," in *Proceedings of IEEE International Conference on Multimedia and Expo*. IEEE, 2008, pp. 37–40.

[43] C.-Y. Kim, O.-J. Kwon, W.-G. Kim, and S.-R. Choi, "Automatic system for filtering obscene video," in *Proceedings of 10th International Conference on Advanced Communication Technology*, vol. 2. IEEE, 2008, pp. 1435–1438.

[44] J. Z. Wang, J. Li, G. Wiederhold, and O. Firschein, "System for screening objectionable images," *Computer Communications*, vol. 21, no. 15, pp. 1355–1360, 1998.

[45] T. Endeshaw, J. Garcia, and A. Jakobsson, "Classification of indecent videos by low complexity repetitive motion detection," in *Proceedings of 37th IEEE Applied Imagery Pattern Recognition Workshop*. IEEE, 2008, pp. 1–7.

[46] Q. Zhiyi, L. Yanmin, L. Ying, J. Kang, and C. Yong, "A method for reciprocating motion detection in porn video based on motion features," in *Proceedings of 2nd IEEE International Conference on Broadband Network & Multimedia Technology*. IEEE, 2009, pp. 183–187.

[47] A. Behrad, M. Salehpour, M. Ghaderian, M. Saiedi, and M. N. Barati, "Content-based obscene video recognition by combining 3D spatio-temporal and motion-based features," *EURASIP Journal on Image and Video Processing*, vol. 2012, no. 1, p. 23, 2012.

[48] S. Jung, J. Youn, and S. Sull, "A real-time system for detecting indecent videos based on spatiotemporal patterns," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 696–701, 2014.

[49] F. Nian, T. Li, Y. Wang, M. Xu, and J. Wu, "Pornographic image detection utilizing deep convolutional neural networks," *Neurocomputing*, vol. 210, pp. 283–293, 2016.

[50] P. Vitorino, S. Avila, M. Perez, and A. Rocha, "Leveraging deep neural networks to fight child pornography in the age of social media," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 303–313, 2018.

[51] Y. Wang, X. Jin, and X. Tan, "Pornographic image recognition by strongly-supervised deep multiple instance learning," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 4418–4422.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[53] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[55] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[56] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

Visualización y Análisis de Tráfico Móvil para la Securización de Redes y Sistemas

J.A. Gómez Hernández, J. Camacho, P. García Teodoro, G. Maciá-Fernández,
M. Robles Carrillo, A. Muñoz Ropa, J.A. Holgado Terriza
Network Engineering & Security Group - CITIC - Universidad de Granada
{jagomez, josecamacho, pgteodor, gmacia, mrobles, aropa, jholgado}@ugr.es

***Resumen*—Dado el creciente uso de dispositivos de usuario móviles en entornos de red corporativos, junto con la también creciente existencia de vulnerabilidades en estos dispositivos, se hace precisa la adecuada protección de este tipo de entornos y sistemas. A este fin, los autores están desarrollando un proyecto de investigación fundamentado en la monitorización de dispositivos y usuarios finales para provisionar un control de acceso dinámico que reduzca los riesgos del entorno. El diseño y despliegue de la propuesta científico-técnica requiere un marco de experimentación que permita validar los desarrollos realizados. El presente trabajo recoge una base de datos de tráfico móvil adquirida en el entorno de red de la Universidad de Granada, sobre la cual se lleva a cabo un análisis preliminar del comportamiento de este tipo de usuarios como paso previo para alcanzar los objetivos del proyecto. Para una mejor comprensión, dicho análisis se apoya en la herramienta de visualización Gephi.**

***Index Terms*—Control de acceso, Comportamiento, Dispositivo móvil, Monitorización, Visualización**

Tipo de contribución: *Investigación original*

I. Introducción

Según diversos estudios, el número de dispositivos móviles (*smartphones* y *tablets*) supera ya el de otros dispositivos más tradicionales como sobremesa e incluso portátiles [1], [2], [3]. En consonancia con ello, en torno a la mitad del tráfico Internet actual está asociado a dispositivos móviles [4], no limitándose el empleo de estos a servicios de ocio o entretenimiento personal, sino que resulta cada vez mayor su operación en entornos de trabajo corporativos [5].

Por otro lado, es manifiesto el elevado (y creciente) número de vulnerabilidades de este tipo de dispositivos, en particular por lo que respecta a Android [6], [7]. Ello va de la mano del también significativo incremento de vulnerabilidades y ataques a dispositivos IoT [8].

En este contexto, parece evidente la necesidad de securizar este tipo de sistemas y entornos. A este fin, en el proyecto de investigación TIN2017-83494-R del que los autores son miembros del equipo de trabajo, se propone el desarrollo de un sistema de control de acceso y monitorización dinámico de dispositivos móviles a fin de determinar la potencial ocurrencia de situaciones de riesgo para el entorno y, en su caso, modificar las políticas de acceso concretas a aplicar.

La propuesta se denomina MDSM (de *Mobile device Dynamic Security Management*) y se sustenta sobre un esquema de acceso basado en atributos (ABAC, *Attribute Based Access Control*) [9], siendo su flujo operacional como se muestra en la Figura 1:

1. La red debe conocer el nivel de seguridad del usuario/dispositivo de cara a la provisión de acceso a servicios y recursos. Para ello se estima dinámicamente (mediante procedimientos ML, *machine learning* [10]) un *perfil de seguridad* para el usuario/dispositivo basado en ciertos atributos de seguridad derivados de su configuración y operación en el tiempo.
2. Dicho perfil de seguridad se monitoriza a lo largo del tiempo, tanto al inicio de la comunicación (a través de mensajes *Access Request*) como durante ella. De este modo, el acceso se concede (si procede) al principio y se renueva periódicamente en base al nivel de seguridad obtenido. En caso de determinarse la existencia de algún riesgo de seguridad, el acceso podrá ser restringido e incluso denegado. En todo caso, antes de ello el sistema podrá notificar al usuario/dispositivo para que resuelva los problemas detectados.

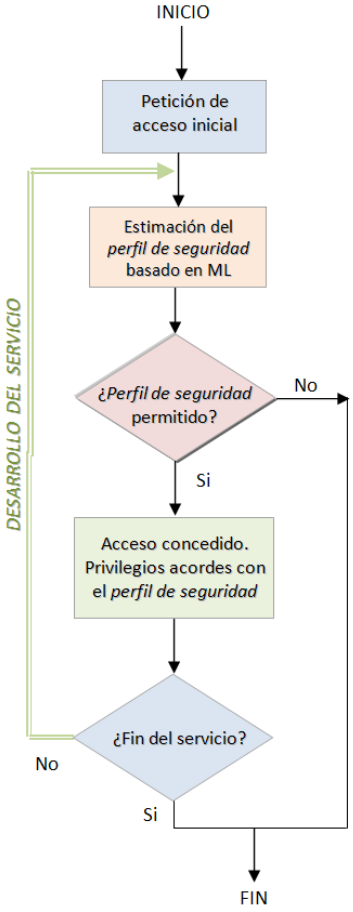


Figura 1. Flujo operacional de MDSM.

3. MDSM respeta la privacidad del usuario/dispositivo en la estimación del perfil de seguridad asociado y en la toma de decisiones de acceso correspondientes.

Como para cualquier otra propuesta de índole científico-técnica, es precisa la disposición de un marco experimental que permita la validación de los desarrollos realizados. A este fin, a la fecha se ha adquirido una base de datos de tráfico móvil preliminar sobre la que iniciar los primeras propuestas de MDSM en la línea anteriormente apuntada. Dicha base de datos se describe en la Sección II. Seguidamente, en la Sección III se describe el empleo de herramientas de visualización de redes para la ayuda en el estudio de patrones y comportamientos. De entre otras posibles, en la Sección IV se hará uso de Gephi para el análisis del tráfico móvil recopilado, en el ánimo de extraer perfiles que nos den una orientación diferenciada acerca de este tipo de usuarios. Finalmenete, la Seccion V resume la aportación del trabajo.

II. Base de datos de tráfico móvil

Para la adquisición de la base de datos se ha realizado primeramente un análisis de los requerimientos que dichos datos debían cumplir para ser válidos en los estudios a realizar posteriormente. Dichos requerimientos son los siguientes:

- *Req1*: Se debe obtener tráfico de usuarios móviles.
- *Req2*: Se debe seleccionar una red en la que posteriormente se puedan implementar los sistemas de control previstos en el proyecto MDSM.
- *Req3*: El proceso de recolección del tráfico debe estar conforme con la normativa vigente.
- *Req4*: Los datos deben contener características del tráfico móvil, que además deben estar suficientemente anonimizadas para no violar la privacidad de los usuarios.
- *Req5*: Los datos deben ser suficientemente representativos para estudiar aspectos como la evolución temporal y la cicloestacionariedad.
- *Req6*: Los datos deben permitir la creación de modelos de perfil según el tipo de usuario e incluso también el tipo de dispositivo utilizado para acceder a la red.

La red elegida en este punto es la de la Universidad de Granada (UGR). En ella, los usuarios móviles (*Req1*) se conectan a través de la infraestructura *Eduroam*. Se ha elegido esta red debido a que posteriormente se podrán implementar en ella mecanismos de control para el tráfico de los usuarios, tal y como está previsto en el proyecto MDSM (*Req2*).

Para la recolección de tráfico en esta red se han seleccionado un total de 35 usuarios voluntarios para participar en el proceso que se ha conectado mediante 73 dispositivos (*smartphone* y portátiles). En media podemos indicar que cada usuario accedería a la red con un *smartphone* y un portátil. El primer paso ha sido identificar el marco jurídico de aplicación para garantizar la conformidad del trabajo previsto con la normativa en vigor al respecto (*Req3*); en concreto, en tres ámbitos: *a)* protección de datos personales; *b)* privacidad de las comunicaciones electrónicas; y *c)* seguridad de redes y sistemas de información.

a) El régimen jurídico establecido en materia de protección de datos a partir del Reglamento General de Protección de Datos establece el principio del consentimiento expreso como fundamento para cualquier tratamiento de datos personales. El uso de los mismos con fines de investigación científica, como en este caso, debe realizarse en las condiciones y con las garantías adecuadas establecidas en esa disposición y en la normativa interna de aplicación, que es la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y Garantía de los Derechos Digitales. Aunque en el momento de solicitar el consentimiento para la recolección de datos esta ley aún no se había adoptado, se trabajó sobre la base de los proyectos de ley existentes y de la normativa específica adoptada por la Universidad de Granada para diseñar dos formularios: un modelo especial, que permitiese un amplio margen de uso de los datos, y un modelo básico, para quienes solo aceptasen un uso limitado y al que finalmente no fue necesario recurrir. El acceso a los datos de UGR estuvo condicionado a la presentación de los formularios de manifestación del consentimiento. Sobre este punto, el proyecto se desarrolla respetando la normativa europea y nacional relativa a la elaboración de perfiles con fines de investigación científica.

b) La normativa sobre privacidad de las comunicaciones electrónicas sigue basada en la Directiva ePrivacy porque, aunque se ha adoptado en diciembre de 2018 la Directiva por la que se establece Código Europeo de las Comunicaciones Electrónicas, no se ha modificado aún la parte relativa a privacidad. El modelo de consentimiento especial elaborado para este proyecto incluye los aspectos relativos a la privacidad y la confidencialidad.

c) La aplicación de las normas sobre seguridad de las redes y sistemas de información viene determinada por el hecho de que el Real-Decreto Ley 12/2018 reconoce al conjunto de la administración pública la condición de operador de servicios esenciales a los efectos de la Directiva (UE) 2016/1148. Ello ha requerido la elaboración de un modelo de monitorización del respeto de los requisitos de seguridad y notificación establecidos en esa normativa, en proceso de definición, porque sus contenidos dependen en gran medida de la revisión de la Estrategia de Ciberseguridad Nacional que se ha puesto en marcha en agosto de 2018 pero aún no está concluida.

La infraestructura existente en la red de la UGR permite la monitorización de tráfico de usuarios móviles (*Req4*), obteniéndose dicha información en formato *Netflow* (no incluye la captura del *payload* de los paquetes de red para mantener la privacidad de los usuarios). Así, los diferentes puntos de acceso de la red móvil se han configurado para enviar a un colector *Netflow* la información correspondiente a los flujos de los usuarios preseleccionados cuando se autentican en *Eduroam*. Las características consideradas para cada flujo han sido: direcciones IP, puertos origen y destino, protocolo, *flags* TCP observados en dicho flujo, tipo de servicio (ToS), y número de paquetes y *bytes* en ambos sentidos de la comunicación. El resultado de la extracción de los flujos a partir de la información *Netflow* proporciona un *listado de flujos*.

Adicionalmente a esta información, también el sistema de autenticación de la red permite identificar los instantes temporales en los que se realiza la autenticación de un usuario, identificando sus direcciones MAC e IP. De esta forma, para los usuarios participantes en la monitorización, se obtiene

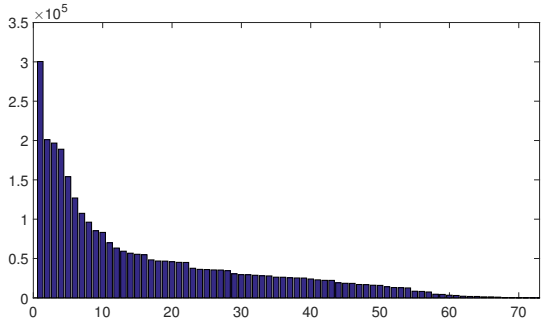


Figura 2. Número de flujos por dispositivo del tráfico total.

Tabla I
CARACTERÍSTICAS CONSIDERADAS EN LOS FLUJOS NETFLOW

Variable	**Número de características → valores**
Source IP	2 → *public, private*
Destination IP	2 → *public, private*
Source port	50 → *specific services, Other*
Destination port	50 → *specific services, Other*
Protocol	5 → *TCP, UDP, ICMP, IGMP, Other*
Flags	6 → *A, S, F, R, P, U*
ToS	3 → *0, 192, Other*
# Packets in	5 → *very low, low, medium, high, very high*
# Packets out	5 → *very low, low, medium, high, very high*
# Bytes in	5 → *very low, low, medium, high, very high*
# Bytes out	5 → *very low, low, medium, high, very high*

un *listado de accesos* a la red identificando el instante de tiempo de dicho acceso y las direcciones MAC e IP asignada. Nótese que la información sobre el usuario concreto que está accediendo queda oculta en este proceso, para así garantizar el anonimato. Además, en este intercambio de información se anonimizan también la direcciones IP y las MAC.

A partir de los dos listados disponibles (flujos y accesos), se realiza un proceso de correlación para identificar qué dispositivos (identificados por su dirección MAC) han generado los diferentes flujos del listado de flujos. Para ello, se utiliza la información común en ambos listados, esto es, las direcciones IP y los instantes temporales. El resultado final es un *listado de conexiones* donde cada flujo (anonimizado) tiene etiquetada también su dirección MAC (anonimizada).

Para hacer posible el estudio temporal de la evolución del tráfico (*Req5*) se monitoriza a los usuarios durante 165 días, obteniendo un total de 3.103.771 conexiones. La Figura 2 muestra el número de conexiones monitorizadas para cada una de las 73 direcciones MAC monitorizadas cuya evolución temporal se puede ver en la Figura 3, donde se muestran con barras verticales las diferentes semanas. Se pueden observar las bajadas del tráfico en los periodos de fin de semana y también en las semanas correspondientes al mes de agosto.

Nótese que esta información (*listado de conexiones*) no solo permite evaluar la evolución temporal de las conexiones, sino también identificar patrones de comportamiento de los diferentes dispositivos (*Req6*).

Un último paso en el procesado de la base de datos es la extracción de características agrupadas en intervalos temporales. Para ello se definen intervalos de 1 minuto de duración, y se obtienen las 138 características que se recogen en la Tabla I utilizando la herramienta *FCParser* (https://github.com/josecamachop/FCParser). Como ejemplo, la característica *pub_sourceIP* contiene el número de direcciones IP origen públicas observadas durante 1 minuto. Para los diferentes valores de puerto se establecen también 49 variables cada una correspondiente a un puerto[1], y una más para los puertos restantes. Por ejemplo, la variable *sport_ssh* contiene el número de observaciones de conexiones desde el puerto 22 durante 1 minuto.

III. CIBERSEGURIDAD Y VISUALIZACIÓN

La provisión de seguridad en redes y sistemas se sustenta en gran medida en la monitorización y supervisión del entorno, de manera que, más allá de la necesaria adopción de mecanismos preventivos (uso de criptografía en las comunicaciones, implementación de cortafuegos, etc.), seamos capaces de determinar en un momento dado la ocurrencia de actividades o situaciones potencialmente peligrosas para, en su caso, adoptar las medidas correctoras oportunas.

Los entornos de redes y comunicaciones actuales se caracterizan por una creciente complejidad, tanto desde el punto de vista de la heterogeneidad como del número de subsistemas y dispositivos base componentes. Adicionalmente a ello, también la velocidad de cómputo y de transmisión resultan significativas. Como resultado, la cantidad de información y datos resultante del proceso de monitorización de un entorno dado es de alta complejidad desde distintos puntos de vista: volumen, variedad, velocidad, ... Es lo que viene en denominarse Big Data.

En este contexto, la visualización de datos constituye una herramienta de alta relevancia en el análisis exploratorio de información, en particular en el campo de la seguridad [11], [12]. Con el objetivo principal de encontrar y mostrar la existencia de ciertos patrones y comportamientos en la información analizada, los tipos de visualizaciones disponibles son diversos [13]:

- Temporal, donde se considera la cronología temporal a la hora de representar los datos. Algunos ejemplos son las líneas y series temporales, los diagramas de Gantt, los diagramas de arco o los diagramas aluviales.
- Espacial, donde se hace uso de mapas coloreados función de los valores de las variables estadísticas consideradas. Es el caso de los cartogramas, los mapas de distribución de puntos o los mapas de contorno
- Volumétrico, donde se usa uan visualización 3D para los datos. Aunque este tipo de visualización suele emplearse en entornos como el médico o el científico, también puede ser de interés en el campo de la seguridad en redes y sistemas.
- Jerárquico, donde se muestran los datos y variables dentro de una organización. Ejemplos de ello son los diagramas en árbol, los mapas en árbol, los árboles radiales o los dendrogramas.
- Multidimensional, donde se recurre al empleo de indicadores separados para las variables analizadas. Este el caso de los gráficos de burbujas, de barras o radiales, los diagramas de araña o las coordenadas paralelas.
- Red, donde se proporciona un gráfico formado por una

[1]Se han seleccionado los puertos más relevantes: 22, 53, 80, 443, etc.

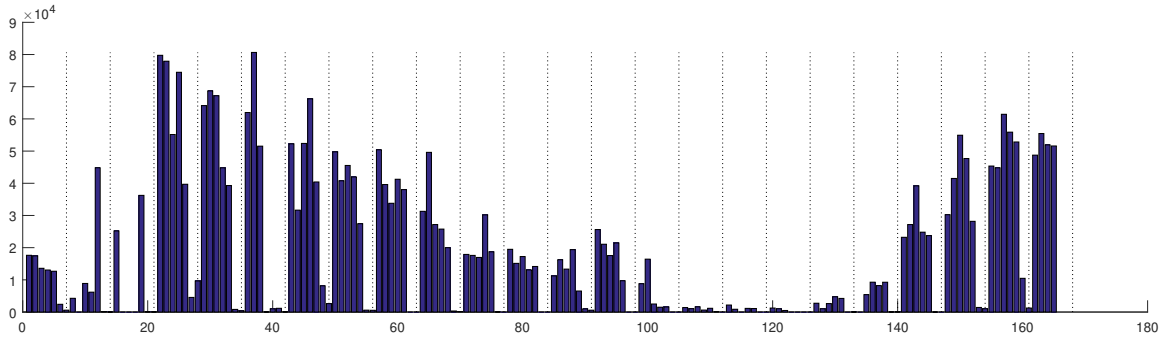


Figura 3. Número de flujos por día.

serie de nodos con enlaces entre ellos. Este tipo de visualización suele ser de amplio uso en el contexto que nos ocupa y algunos ejemplos son los diagramas de matrices, los grafos de dependencias o los esquemas de colmena.

Por lo que respecta a la disposición de herramientas de visualización, en http://www.wikiviz.org/wiki/Tools se pueden encontrar referencias a varias, desde específicas para datos textuales hasta *kits* para representaciones varias. Aunque son numerosas las herramientas que puede emplearse para la visualización de datos relacionados con redes y sistemas, algunas de las más adoptadas se citan a continuación.

Cytoscape (https://cytoscape.org/) es una plataforma de código abierto para visualizar redes complejas e integrarlas con cualquier tipo de datos, disponiendo de una variedad de *plugins* para dominios diversos. *Guess* (http://graphexploration.cond.org/) es una herramienta de análisis y visualización para grafos y redes, que contiene un lenguaje embebido llamado Gython. *Graphviz* (http://www.graphviz.org/) es también un software de visualización de gráficos de código abierto, el cual permite la descripción de gráficos en un lenguaje texto simple y realiza diagramas en forma de imágenes y SVG para páginas web, PDF o postscripts par a su inclusión en otros documentos, o hace visualizaciones web interactivas. *GVF* (http://gvf.sourceforge.net/) es un conjunto de patrones de diseño y aproximaciones para la visualización y manejo de estructuras gráficas, donde se dispone del formato de intercambio GraphXML. *Walrus* (http://www.caida.org/tools/visualization/walrus/) permite la visualización de grandes gráficos en formato 3D, para lo cual incorpora un ojo de pez que posibilita una visión global y detallada de forma simultánea. *Gephi* (https://gephi.org/) es una plataforma de visualización y exploración para todo tipo de redes y sistemas complejos, siendo además de código abierto, gratuita y disponible para Windows, Linux y Mac.

Algunos estudios como [14], [15], [16] abordan la comparación de distintas herramientas de visualización. Aunque todas ellas presentan pros y contras, una de las más aceptadas es Gephi [17]. Las razones de ello son un buen comportamiento en características, el tiempo de ejecución implicado, el soporte de formatos de archivos, informe y realimentación para el usuario, los métodos de distribución permitidos, la interacción con nodos individuales y la calidad de la visualización. Además, es capaz de manejar *datasets* de tamaño elevado. A modo de ejemplo, en la Figura 4 se muestran visualizaciones de Gephi. Es por ello que, sin ánimo de minusvalorar el posible empleo de otras herramientas de visualización, será Gephi la que aquí usaremos para analizar nuestra base de datos de tráfico móvil.

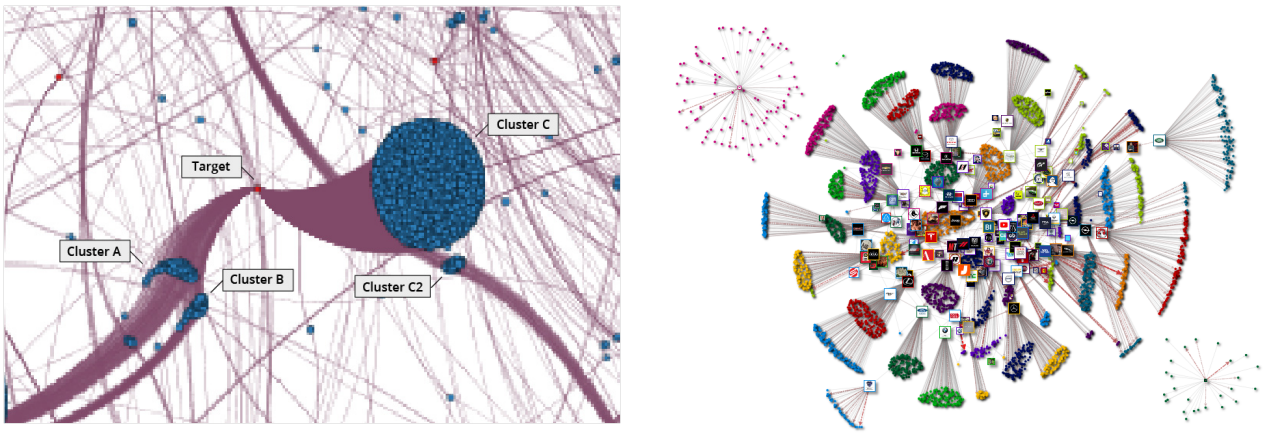
IV. Análisis de datos y derivación de comportamiento

Como se indicaba en el Apartado II, del procesamiento de la base de datos (Tabla I), y para que la visualización sea útil, nos centraremos en tres parámetros de cara a simplificar la visualización a realizar en lo que sigue con Gephi: la dirección *MAC*, el *puerto* de comunicación del equipo que actúa como servidor de la comunicación y el número de comunicaciones entres ambos, que denominaremos *peso*.

Con estos datos, transformamos el espacio del problema en un grafo bipartito dirigido $\overrightarrow{G}$=<V,$\overrightarrow{E}$>, donde los vértices V son bien del tipo V1 que representa direcciones MAC, bien al conjunto V2 que representa los puertos, V1,V2$\subseteq$V. Además se cumple que V=V1$\bigcup$V2 y V1$\bigcap$V2 =ϕ. Los bordes $\overrightarrow{e}$=(v1,v2)$\in\overrightarrow{E}$, que unen un vértice tipo V1 (fuente) y otro de tipo V2 (destino), representan la conexión entre una dirección MAC y el puerto remoto al que accede. Además, este enlace viene cuantificado por un peso que indica el número de ocurrencias de esa conexión. En nuestro caso, V1={*MAC-1*, . . . ,*MAC-73*} y V2={0, . . . ,65535}. Para mejorar la visualización se ha optado por colorear los nodos en función del peso de los mismos y los casos más significativos además se han etiquetado con el nombre asociado que los identifica. En nuestro estudio, el grafo tiene un grado medio de 2.891 y un grado medio con pesos de 224.502.

Dadas las características de los grafos que vamos a construir, algunas métricas útiles en otro tipo de redes como la autoridad, modularidad, proximidad al la centralidad, etc., no resultan útiles. En nuestro caso, usaremos principalmente la métrica del rango *grado con pesos* (suma de los bordes de un nodo más sus pesos) o sus extensiones para un grafo dirigido: grado de entrada con pesos (número de aristas de entrada a un vértice) y grado de salida con pesos (ídem de salida). Estas últimas métricas permitirán resaltar en el grafo unos u otros tipos de vértices en función de las necesidades.

Otro aspecto clave es seleccionar un algoritmo de modelado de distribución [18] de entre los múltiples que permite la herramienta. En nuestro caso, en base a [19] y tras varias pruebas empíricas, observamos que el algoritmo que mejor se



(a) (b)

Figura 4. Ejemplos de visualizaciones con Gephi: (a) botnets, (b) RRSS.

adapta es el de Yifan Hu [20], que genera un gráfico estéticamente comparable al resto de distribuciones pero menos conservador y con un menor tiempo de visualización. Este algoritmo facilita las visualizaciones ya que, en mayor medida que otros, empuja los nodos con un bajo numero de aristas hacia la periferia.

Dada la gran cantidad de días, direcciones MAC y puertos monitorizados, debemos establecer una estrategia de visualización. Basándonos en el hecho de que parámetros como las conexiones de mayor duración, las mayores cantidades de datos transferidos y un número elevado de conexiones pueden mostrar posibles comportamientos y anomalías en los flujos capturados, establecemos el siguiente procedimiento:

1. Detectar los nodos que han generado la principal cantidad de tráfico.
2. Ver la evolución temporal de los nodos detectados en el punto anterior.
3. Analizar los comportamientos concretos de esos nodos.

Para ver los nodos que han tenido más trafico, Punto 1, hemos realizado una primera visualización con el tráfico total hacia Internet de todos los nodos durante el periodo completo sumando todos los flujos y sus pesos hacia cualquier puerto por cada nodo. El resultado puede verse en la Figura 5, donde, con la funcionalidad de la herramienta para representar el tamaño de un nodo en función de su peso, se han resaltado los 5 nodos (*MAC-1* a *MAC-5*) que suponen el 50 % del tráfico analizado. Concretamente, el nodo *MAC-1* supone un 25 % del trafico total.

Una vez seleccionados los nodos a observar y para representar la evolución temporal de tráfico, Paso 2, realizamos una visualización dinámica de todo el periodo monitorizado. Para realizarla se parte del *dataset* anterior donde se añade a cada par <*MAC, puerto-servidor*> una marca temporal de inicio y otra de fin correspondientes al día al que pertenece dicho par mediante un *script* para automatizar el proceso. El grafo original que se obtiene es complejo de visualizar pues el número de puertos usados por todas las direcciones MAC es muy elevado, lo que hace que tengamos un grafo poco útil.

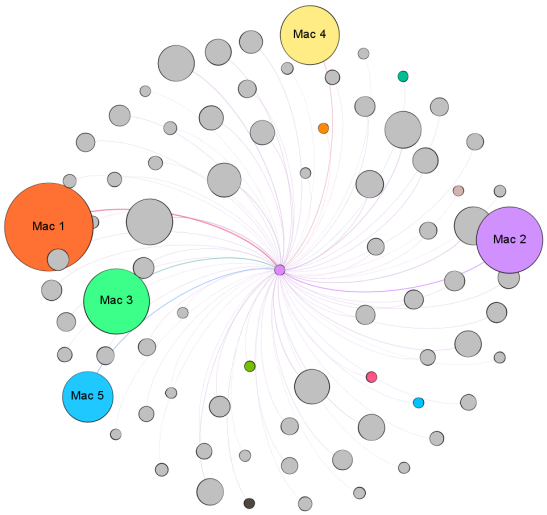


Figura 5. Flujos totales por equipo hacia Internet.

De cara a destacar patrones de comportamiento, haremos uso de concepto de *entropía* del grafo $\overrightarrow{G}=(V,\overrightarrow{E})$ definida como la entropía de Shannon (H) [21]:

$$H(G) = -\sum_{v \in V \neq} p(v) log_2(p(v))$$

donde $p(v)$ es la probabilidad de encontrar un nodo con grado v entre todos los valores de grado distintos ($V\neq$ es el conjunto de valores de grados distintos de V). Como define la ecuación, un valor alto de entropía $H(G)$ del grafo $\overrightarrow{G}=(V,\overrightarrow{E})$ indicará una alta variabilidad en la distribución de nodos y, a la inversa, el significado de una baja entropía es una baja variabilidad.

Dado que nuestro objetivo es obtener un grafo con la mayor entropía posible (no la máxima) que nos permita identificar patrones de comportamiento, vamos a utilizar la capacidad de filtrado de la herramienta Gephi en función del grado

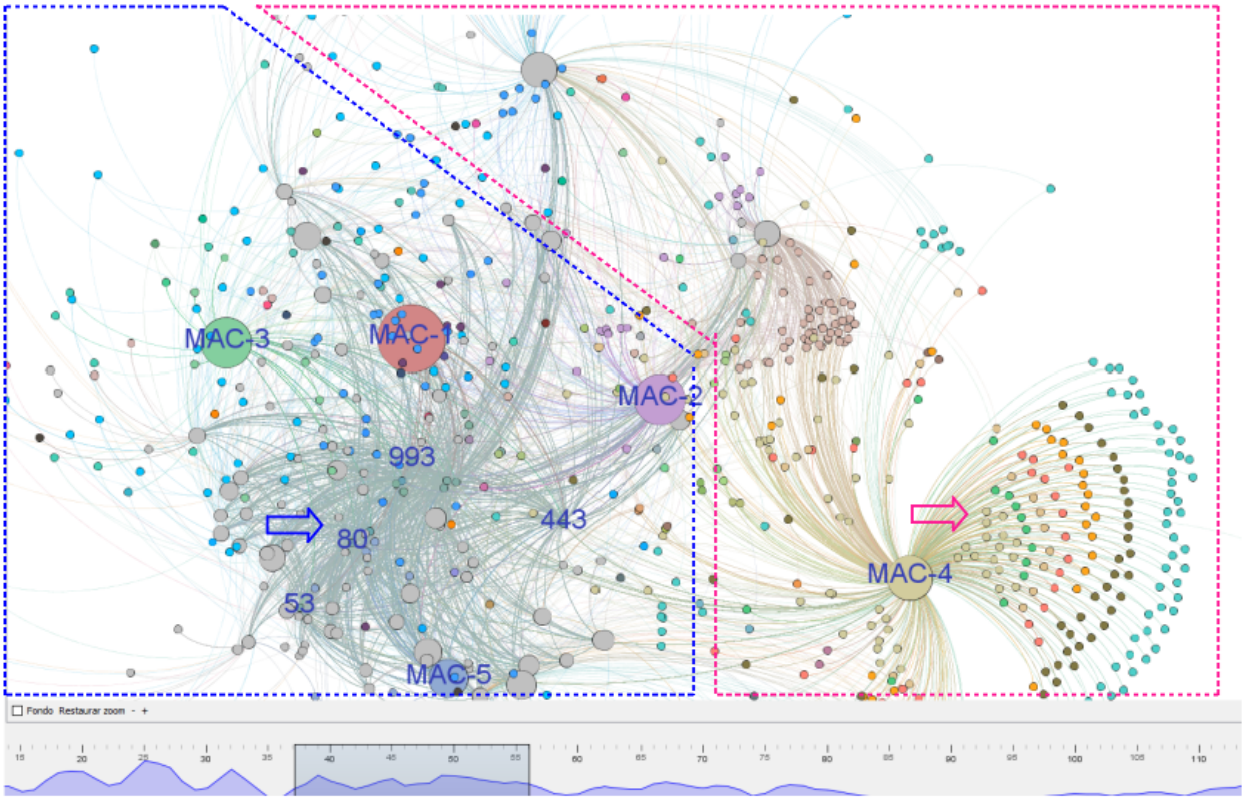


Figura 6. Flujos de varios días dentro en la serie temporal completa.

de pesos para reducir la entropía. En nuestro caso, solo es necesario filtrar por grado 3 para visualizar un patrón de comportamiento. La Figura 6 muestra una instantánea de tráfico global que comprende los días 37 a 56, de los 126 de que disponemos. En la serie temporal hemos marcado aquellas direcciones MAC detectadas en el Paso 1 de cara a facilitar su identificación.

En la Figura 6 hay dos patrones que debemos resaltar en relación a las direcciones MAC en estudio. Podemos ver que las aristas correspondientes a algunas de las MAC analizadas, concretamente la zona demarcada con la línea discontinua azul (donde se encuentran las direcciones *MAC-1*, *MAC-2*, *MAC-3*, y *MAC-5*) que confluyen hacia una zona común marcada con una flecha azul. El conjunto denso de estas aristas, zona de aristas de color verdoso, apuntan a los vértices *puerto* muy utilizados por las direcciones MAC, entre los que destacan los puertos más esperados: 53, 80, 443 y 993 (marcados con la flecha azul, como puede verse en la Figura 7 y que se resalta cuando aplicamos un filtro con valor 80.000 para el rango de pesos con objeto de eliminar puertos poco usados que introducen ruido). Para llevar a cabo un estudio más detallado, vamos a descartar este tráfico que representa un volumen elevado a puertos de uso común.

La otra zona a destacar, zona demarcada por la línea discontinua roja de la gráfica anterior, corresponde a los bordes que salen de la *MAC-4* hace un amplio abanico de vértices puerto (flecha roja en la Figura 8). Como se puede observar, el volumen de tráfico no es muy elevado pero los destinos son un amplio número de *puertos* distintos con

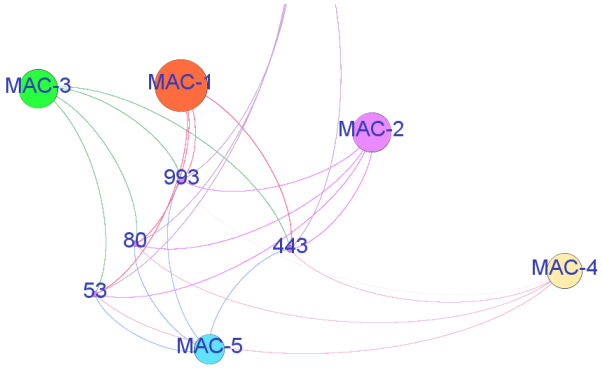


Figura 7. Puertos de servidores con mayor volumen de tráfico.

un rango de pesos muy variado (bandas de nodos *puertos* con diferentes colores: azul para el peso 3, negro para 4, naranja 5, etc.). Este en un aspecto no común al resto de vértices de direcciones MAC de la visualización, donde como indicábamos en el párrafo anterior, lo más normal son flujos a puertos más comunes, y esporádicamente a otros puertos.

Por otra parte, la barra en la parte inferior de la Figura 6 muestra la evolución temporal del número de aristas (flujos). En el intervalo representado, podemos ver que el número de aristas tiene un valor intermedio relativamente constante durante los días seleccionados, frente al resto de días, donde encontramos un número mayor de fluctuaciones.

Para el estudio del Paso 3 propuesto con anterioridad y de

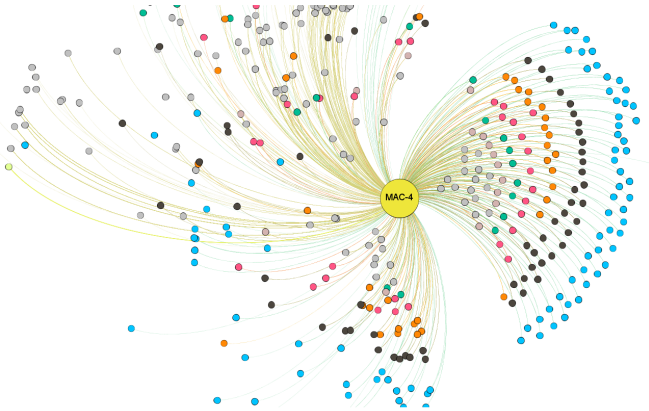


Figura 8. Patrón de acceso a puertos de servidores del nodo *MAC-4*.

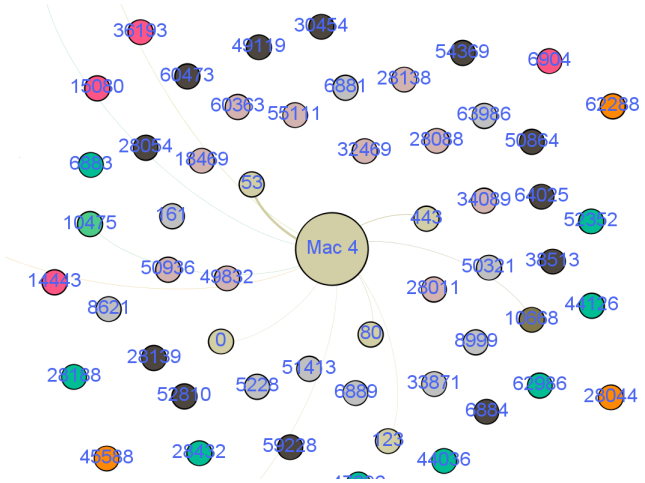


Figura 9. Un instante en la serie temporal del nodo bajo estudio.

cara a analizar con detalle los flujos del equipo con dirección *MAC-4*, hemos realizado una evolución temporal de dicho nodo en el periodo indicado anteriormente. Un instante de esta serie temporal se muestra en la Figura 9 e incide en el comportamiento singular del nodo ya que, además del número elevado de puertos en uso, los puertos más comunes (53, 80, 443, ...) son accedidos de forma periódica, pero además hay tráfico de baja intensidad pero constante hacia servicios como Bittorrent (6881-6889), iTunes (puertos 8000–8999) y un alto número de puertos identificados como *unassigned* por la IANA (10475, 28011, 32469, ...).

Como se desprende del análisis anterior, hemos detectado claramente dos tipos de comportamientos. Por un lado, un amplio número de usuarios/equipos mantiene tráfico que podemos denominar convencional, donde el uso de Internet por parte de los mismos está principalmente enfocado al acceso a correo electrónico cifrado, web (http o https), y, por supuesto, un uso intensivo de DNS. De otro lado, algunos equipos acceden a un número elevado y variado de puertos no registrados o asignados. El hecho de que veamos dos perfiles, uno de tráfico tradicional cliente/servidor (puertos 80, 443, etc.) y otro donde hay muchos puertos destino, nos hace hipotetizar que el segundo perfil puede deberse a tráfico P2P.

Para profundizar en el análisis de esos dos tipos de nodos, hemos realizado una nueva visualización de los nodos con direcciones MAC-1 y MAC-4. Ahora representamos, a partir de la base de datos de flujos, un grafo para el rango de días bajo estudio en el que los bordes representan la conexión entre las direcciones IP destino de la comunicación con el nodo en estudio y el puerto del servidor usado en dicha conexión. El resultado se muestra en la Figura 10.

Es inmediato ver que la entropía del grafo (b) de la Figura 10 es mayor que la del grafo (a), y no solo en la periferia del grafo. Como se puede observar, en los anillos de nodos interiores marcados en colores según su peso, apreciamos menor entropía en el caso (a) que en caso (b). Esto, junto con el uso de puertos ligados a redes P2P para el caso del grafo (b), corrobora los estudios [22], [23], [24], donde se demuestra cómo las comunicaciones en redes P2P muestran mayor entropía, es decir, una mayor variabilidad en las direcciones IP y puertos a las que se conecta un equipo. En el caso de los puertos, esa variabilidad supone no solo el amplio uso de puertos conocidos con un peso relevante sino también en el uso de puertos arbitrarios típico de las actuales aplicaciones P2P.

En el estudio presentado, los grafos permiten detectar o identificar de forma relativamente rápida nodos por categoría (tráfico total, acceso a servicios comunes o no) que podrán ser analizados a posteriori usando todos los parámetros obtenidos en el primer pre-procesamiento, o usando directamente la base de datos de flujos.

V. Conclusiones

En el marco del proyecto TIN2017-83494-R, estamos desarrollando un trabajo para la monitorización de dispositivos y usuarios para la provisión de un control de acceso dinámico. En una primera fase se ha adquirido una bases de datos de tráfico móvil de un conjunto de voluntarios dentro de la Universidad de Granada que se ha analizado con Gephi.

A partir de esa base de datos se han realizado varios pre-procesamientos de los datos para seleccionar un conjunto reducido de parámetros como son las direcciones MAC, el puerto remoto, número de flujo sobre ese puerto, al objeto de realizar visualizaciones que nos den una aproximación al comportamiento de los usuarios y al tráfico generado.

Como trabajo siguiente, prevemos la implementación de esquemas automáticos que consideren los parámetros apuntados para la clasificación.

Agradecimientos

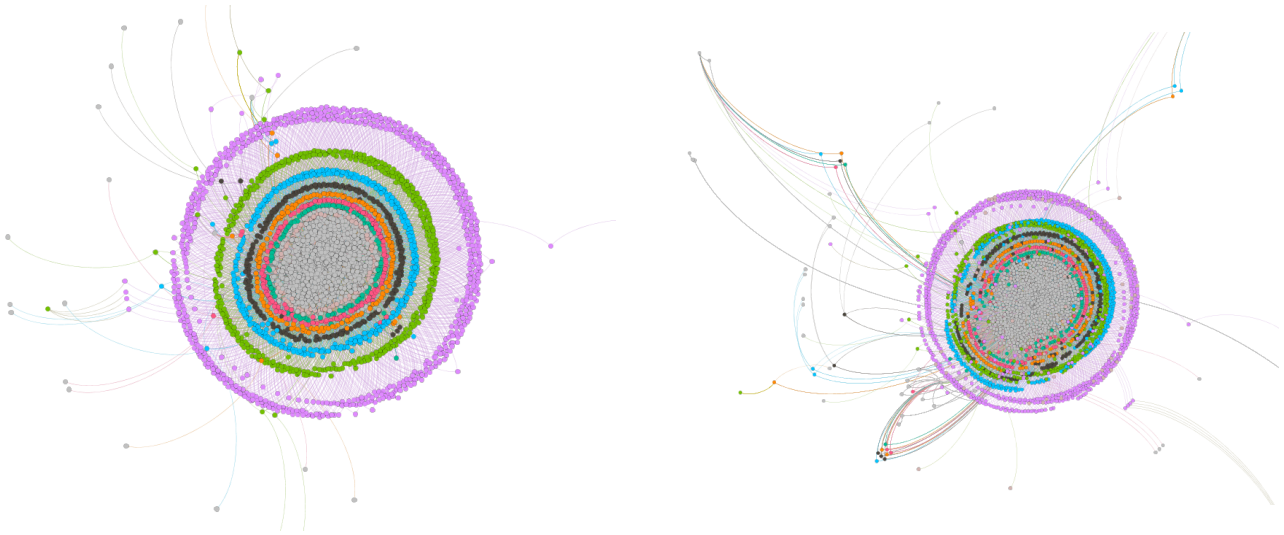
Agradecemos al Centro de Servicios de Informática y Redes de Comunicaciones (*CSIRC*) de la Universidad de Granada su colaboración en la recogida de los datos relativos al tráfico de los voluntarios del proyecto.

Este trabajo ha sido financiado parcialmente por el Gobierno de España, con fondos FEDER, a través del proyecto TIN2017-83494-R.

Referencias

[1] SmartInsights: “Mobile Marketing Statistics compilation”, en *https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/*, Julio, 2018.

[2] GlobalStats: “Desktop vs Mobile vs Tablet Market Share Worldwide”, en *http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet*, Febrero, 2019.



(a) (b)

Figura 10. Patrones de comunicaciones con servicios de red: (a) uso convencional, (b) uso de redes P2P.

[3] Google: "Consumer Barometer with Google", en *https://www.consumerbarometer.com/en/trending/?countryCode=UK&category=TRN-NOFILTER-ALL*, Febrero, 2019.

[4] HostingFacts: "Consumer Barometer with Google", en *https://www.consumerbarometer.com/en/trending/?countryCode=UK&category=TRN-NOFILTER-ALL*, Diciembre, 2018.

[5] Wikipedia: "Bring your own device", en *https://en.wikipedia.org/wiki/Bring_your_own_device*, Marzo, 2019.

[6] Check Point: "2018 Security Report", en *https://www.checkpoint.com/downloads/product-related/report/2018-security-report.pdf*, 2018.

[7] Symantec: "2018 Internet Security Threat Report", en *https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-executive-summary-en.pdf*, 2018.

[8] Sophos, "SophosLabs 2019 Threat Report", en *https://www.sophos.com/en-us/medialibrary/pdfs/technical-papers/sophoslabs-2019-threat-report.pdf*, Noviembre, 2018.

[9] V.C. Hu, D.R. Kuhn, D.F. Ferraiolo, J. Voas, "Attribute-based access control". *Computer, vol. 48, n. 2*, pp. 85-88, 2015.

[10] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, G. Maciá-Fernández: "PCA-based Multivariate Statistical Network Monitoring for Anomaly Detection". *Computers & Security, vol. 59*, pp. 118-137, 2016.

[11] R. Marty: *Applied Security Visualization*. Addison Wesley Professional, 2008.

[12] B. Balakrishnan: "Security Data Visualization". *SANS InstituteInformation Security Reading Room*, 2008. Disponible en https://www.sans.org/reading-room/whitepapers/metrics/security-data-visualization-36387

[13] A. Zoss, "Data Visualization". Disponible en https://guides.library.duke.edu/datavis/vis_types.

[14] Y. Uchida, S. Matsuno, Y. Iha, M. Sakamoto: "Comparison of tools for visualization of big data in SMEs". *Journal of Scientific Research and Development, vol. 3, n. 3*, pp. 97-103, 2016.

[15] E.G. Caldarola, A.M. Rinaldi: "Big Data Visualization Tools: A Survey. The New Paradigms, Methodologies and Tools for Large Data Sets Visualization". *6th International Conference on Data Science, Technology and Applications*, pp. 296-305, 2017.

[16] M.A.M. Faysal, S. Arifuzzman, "A Comparative Analysis of Large-scale Network Visualization Tools", *2018 IEEE International Conference on Big Data (Big Data)*, pp 4837-4843, 10-13 Dic., 2018.

[17] B. Bastian, S. Heymann, M. Jacomy: "Gephi: An Open Source Software for Exploring and Manipulating Networks". *Third International ICWSM Conference*, pp. 361-362, 2009.

[18] Ken Cherven, *Matering Gephi Network visualization*, Packt Publishing, 2015.

[19] Georgios A. Pavlopoulos, David Paez-Espino, Nikos C. Kyrpides, y Ioannis Iliopoulos, "Empirical Comparison of Visualization Tools for Larger-Scale Network Analysis", *Advances in Bioinformatics (Hindawi), Volume 2017*, 8 pages, 2017.

[20] Yifan Hu, "Efficient, High-Qality Forced-Directed Graph Drwaing", *The Mathematica Journal, vol. 10:1*, pp.37-71 2006.

[21] Sanuel de Sousa y Walter G. Kropatsch, "Data Graph Formulation as the Minimum-Weigth Maximum-Entropy Problem", *LNCS, 9069*, pp. 13-22, 2015.

[22] F. Constantinou y P. Mavrommatis, "Identifying Known and Unknown Peer-to-Peer Traffic", *Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)*, 2006.

[23] J. V. P. Gomes et al., "Analysis of Peer-to-Peer Traffic Using a Behavioural Method Based on Entroy", *IEEE International Conference on Performance, Computing and Communications (IPCCC)*, Jan 2008.

[24] M. Bhatia y M.K. Rai, "Identifying P2P traffic: A survey", *Peer-to-Peer Netw. Appl. (2017) 10*:1182–1203, 2017.

MSNM-S: An Applied Network Monitoring Tool for Anomaly Detection in Complex Network Environments

Roberto Magán Carrión
Dpt. of Computer Engineering
University of Cádiz
Network Engineering & Security Group
University of Granada
{roberto.magan@uca.es, rmagan@ugr.es}

José Camacho
Dpt. of Signal Theory,
Telematics & Communications,
Network Engineering & Security Group
University of Granada
{josecamacho@ugr.es}

Gabriel Maciá-Fernández
Dpt. of Signal Theory,
Telematics & Communications,
Network Engineering & Security Group
University of Granada
{gmacia@ugr.es}

Ismael Jerez Ibáñez
Dpt. of Computer Engineering
University of Cádiz
{ismael.jerezibanez@uca.es}

***Abstract*—Recent forecasts predict a huge number of devices inter-connected by 2021. In this scenario, new applications and services are devised mainly focused on improving people's daily life. Monitoring devices, applications and the involved information to detect security events is a great challenge in such a complex environment. Because of that, new methods, algorithms and tools must be developed. In this work, we introduce the recently released MSNM-S tool which is able to carried out the monitoring and detection of security events in this kind of scenario.**

***Index Terms*—Monitoring, anomaly detection, diagnosis, communication networks, multivariate analysis**

I. Motivation

Several technical reports forecast of 30 billion IoT devices around the world by 2021 and [1] more than 3 billions of M2M connections by 2022 [2]. This scenario offers new services and applications for a more comfortable people's life. However, it arises security concerns too. How to monitor and control what is happening in this scenario is a great challenge since the attack exposure surface grows almost exponentially with the number of devices inter-connected. Additionally, how to manage the generated data coming from different information sources like the applications or networking devices and communications is something to take into account. This way, key aspects like managing volume, veracity or velocity of the data are of upmost importance for quick and efficient detection and reaction against security attacks. In fact, these aspects could limit the practical application of the solutions, even more in the current scenario.

To address the previous issues, we developed MSNM-S (Multivariate Statistical Network Monitoring-Sensor) a tool for monitoring and anomaly detection that:

1) Drastically reduces the network monitoring traffic but keeping the detection performance.
2) Adds privacy in communications.
3) Is scalable, versatile, distributed and dynamically adaptable to changes in the environments.
4) It is ready to be used in complex network and systems scenarios.

MSNM-S was firstly introduced in [3] and is based on the work done by Maciá-Fernández *et al.* [4]. The idea behind the solution is to reduce the network monitoring traffic in hierarchical systems by using the so-called monitoring statistics Q-st [5] y D-st [6] widely used in MSPC (Multivariate Statistical Process Control) for industrial processes.

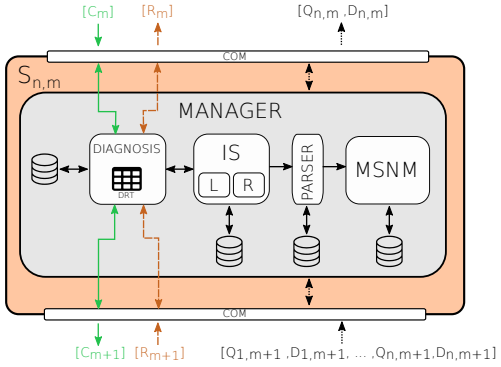


Fig. 1. Functional modules $S_{n,m}$ where n is the sensor ID and m is hierarchical level ID where the sensor is deployed. Black lines corresponds with the monitoring information flow while red and brown lines with the diagnosis one.

The MSNM-S v1.0 version is released under GPL license and we encourage the readers to be an active part of the project which is available at [7].

II. Tool description: main characteristics and operating modes

The developed system is based on spreading the so-called MSNM-Ss (Multivariate Statistical Network Monitoring-Sensors) throughout a hierarchical composition of interconnected devices for, mainly, monitoring and detection of security events. MSNM-S acronym comes from the MSNM approach [8] which is the core of each sensor.

Figure 1 shows the involved MSNM-S functional modules. All together provide the sensor of ways to:

- Easily and dynamically get information from heterogeneous data sources (e.g., *netflow, syslog, IDS logs, etc*) through the IS (Information Source) and PARSER (FaaC (Feature-as-a-Counter approach [9])) modules.
- Monitor complex networks and systems by sending computed statistics from MSNM module to other sensors.
- Talk to each other thanks to the COMmunication module.
- Diagnose the source of a suspicious behavior observed with the DIAGNOSIS module.

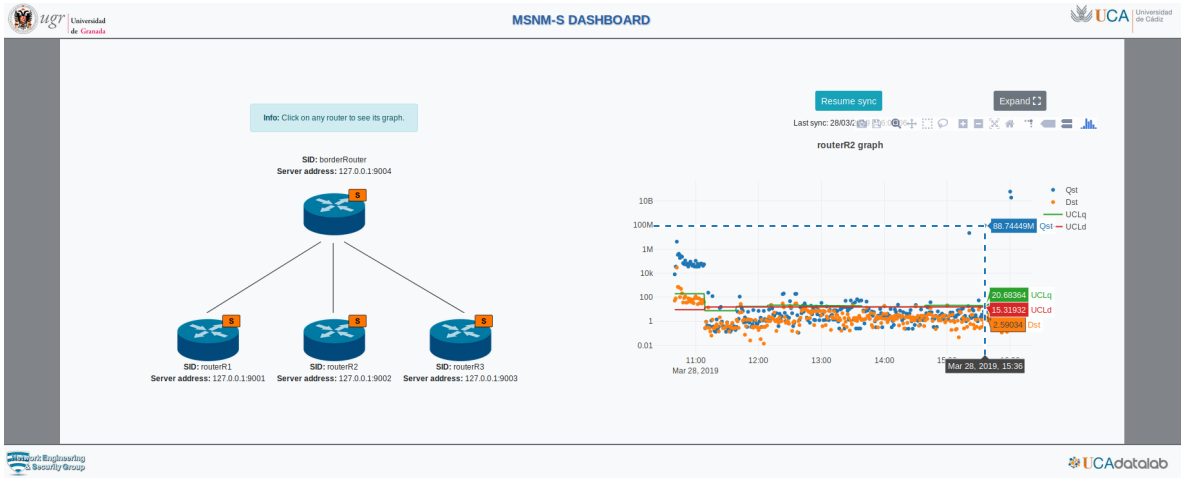


Fig. 2. MSNM-S dashboard showing a typical networking scenario where the sensors (orange boxes) are deployed. The monitoring graph showing the evolution of statistics with time is shown on the right.

Certainly, its main strength relies on how to manage big and complex network architectures, systems and data to perform the monitoring and detection procedures. However, the tool can also be used as is, standalone. Although in its very early development phase, a dashboard to monitor and detect anomalies is provided. Figure 2 shows a specific example of a simple network architecture with two hierarchical levels. The deployed MSNM-Ss (orange boxes) at the lower network level (SIDs: `routerR1, routerR2, routerR3`) are getting and processing *netflow* traffic data to compute the monitoring statistics to be afterwards sent to the one located at the upper network level (SID: `borderRouterr`). The `borderRouter` sensor aggregates the received monitoring information and computes new statistics. By means of inspecting them, a security analyst can determine if and anomaly is taking place when the control limits are exceeded. To this end, the dashboard also provides a monitoring graph (right side of the figure) representing the evolution of statistics with time at each sensor (blue and orange dots for Q-st and D-st statistics, respectively) and the proposed control limits as well (green and red lines for $UCLq$ and $UCLd$ control limits, respectively). It is worth to noting the dynamically adaptation of the system to the environment which lead the system to compute different control limits with time. For that, we currently use the EWMA (Exponentially Weighted Moving Average) approach to dynamically calibrate the sensors[1].

Once the anomaly is detected, a deeper inspection should be done to determine, for example, where the anomaly comes from and why. This is the so-called diagnosis procedure. It is of particular relevance in complex and hierarchical environments where what we only have at the top level is two numerical values. At the time of writing this article the diagnosis procedure is still on development.

III. Conclusions & Future Work

The MSNM-S monitoring and detection tool is introduced here. Although MSNM-S is on its early development stages and much work should be done, we strongly think that solutions like the one proposed here are needed to tackle security concerns in new and complex environments found nowadays.

Future developments will be focus on to automatize the device discovering and sensor deployment throughout a network environment; to still continue with the implementation of the diagnosis procedure; to improve the dashboard interaction and capabilities; and, finally, to validate the tool in real production environments which is not commonly addressed in research works.

[1]How to deploy and run the experiment is explained at the official MSNM-S code repository [7].

Acknowledgment

This work has been partially supported by Spanish MINECO (Ministerio de Economía y Competitividad) through project TIN2014-60346-R and FEDER funds.

References

[1] A. Nordrum, “Popular internet of things forecast of 50 billion devices by 2020 is outdated.” [Online; Accessed 6-March-2019] https://bit.ly/2kVkk9A.

[2] Cisco Systems, “Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022 white paper,” [Online; Accessed 6-March-2019] https://bit.ly/2SNKAhz.

[3] J. M.-F. G. Magán-Carrión, R. Camacho and M. Fuentes-García, “Esquema jerárquico de monitorización y detección de anomalías en red: Aplicación práctica,” in *JNIC2017 III Jornadas Nacionales de Investigación en Ciberseguridad*, October 2017, pp. 184–185.

[4] G. Maciá-Fernández, J. Camacho, P. García-Teodoro, and R. A. Rodríguez-Gómez, “Hierarchical PCA-based multivariate statistical network monitoring for anomaly detection,” in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, diciembre 2016, pp. 1–6.

[5] J. E. Jackson and G. S. Mudholkar, “Control Procedures for Residuals Associated with Principal Component Analysis,” *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[6] H. Hotelling, *Multivariate Quality Control. Techniques of Statistical Analysis*. MacGraw-Hill, 1947.

[7] R. Magán-Carrión, J. Camacho, and G. Maciá-Fernández, “MSNM-S: Multivariate Statistical Network Monitoring-Sensor - Github,” [Online; Accessed 6-March-2019] https://github.com/nesg-ugr/msnm-sensor.

[8] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, “PCA-based multivariate statistical network monitoring for anomaly detection,” *Computers & Security*, vol. 59, pp. 118–137, June 2016.

[9] A. Pérez-Villegas, J. García-Jiménez, and J. Camacho, “FaaC (feature-as-a-counter) parser - Github,” [Online; Accessed 6-March-2019] https://github.com/josecamachop/FCParser.

Evaluación de mejoras en la monitorización estadística multivariante para la detección de anomalías en tráfico ciclo-estacionario

Marta Fuentes-García
Teoría de la Señal,
Telemática y Telecomunicaciones
Universidad de Granada
Email:nmfuentes@ugr.es

José Camacho
Teoría de la Señal,
Telemática y Telecomunicaciones
Universidad de Granada
Email: josecamacho@ugr.es

Gabriel Maciá-Fernández
Teoría de la Señal,
Telemática y Telecomunicaciones
Universidad de Granada
Email: gmacia@ugr.es

***Resumen*—El tráfico de red tiene un claro carácter ciclo-estacionario (por ejemplo, ciclos día/noche o laborables/fines de semana). Esto hace que se puedan identificar patrones de comportamiento distintos dentro de ciertos intervalos temporales: el comportamiento de la red puede variar según las horas dentro de un mismo día. Por otra parte, estos mismos patrones se repiten de forma periódica: por ejemplo, el tráfico de red es similar todas las mañanas los días laborables. Esta particularidad hace más compleja la creación de modelos de normalidad adecuados para la detección de anomalías, así como la aplicación de técnicas que capturen estas dinámicas de manera adecuada sin generar una alta tasa de falsos positivos.**
Nuestro trabajo actual está centrado en evaluar la aplicación de distintas alternativas de detección de anomalías dentro del enfoque de monitorización de redes estadística multivariante (MSNM). En concreto, nuestro objetivo es mejorar el área bajo la curva (AUC) y garantizar así un elevado número de verdaderos positivos a la par que se reducen los falsos positivos.

***Index Terms*—Detección de Ataques, Detección de Anomalías, PCA, MSNM, Análisis Multivariante, Ciberseguridad, Ciclo-estacionario**

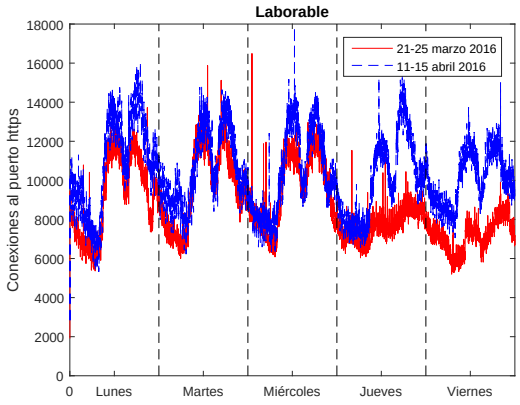
Tipo de contribución: *Investigación en desarrollo*

I. Introducción

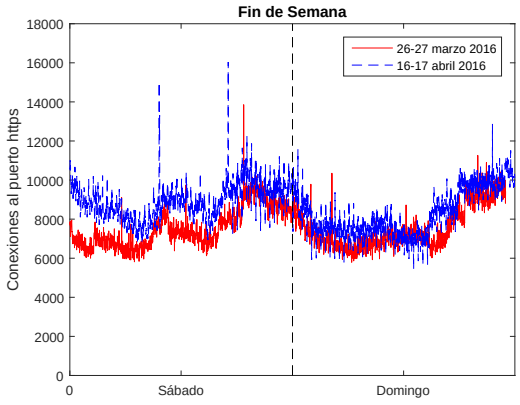
El tráfico de una red puede llegar a ser muy complejo de analizar. En función del tipo de red, existen distintos perfiles o patrones de comportamiento. Recientemente se presentó el conjunto de datos UGR'16 [1]. Este conjunto de datos contiene registros de tráfico netflow de un proveedor de servicios de nivel 3 capturados durante cuatro meses. Durante el cuarto mes se capturó tráfico de la misma red mientras se ejecutaban ataques de forma controlada. Dado que se trata de una captura extensa de tiempo, una de las principales características de este *dataset* es que permite observar distintos tipos de ciclos en el tráfico (como el diario o semanal).

La Fig. 1 muestra el flujo HTTPS de dos semanas distintas obtenido de UGR'16. Se puede observar la diferencia entre días laborables (Fig. 1 (a)) y fines de semana (Fig. 1 (b)). Por otra parte, si se observan individualmente los gráficos, se aprecia la periodicidad del tráfico para las distintas horas del día.

La aplicación de MSNM [2] consiste en parsear y fusionar los datos originales, aplicar *Phase I* [3] para crear un modelo de calibración con datos de operación normales (del inglés, NOC), *Phase II* [3][4][2] para monitorizar los nuevos datos



(a)



(b)

Figura 1. Conexiones de red con puerto destino HTTPS obtenidas de UGR'16. Diferencias y similitudes entre (a) laborables y (b) fines de semana.

detectando anomalías en los mismos y, por último, diagnosticar las anomalías identificadas [3][5][6][7].

Tras la aplicación de MSNM sobre UGR'16 [1][8], parte de nuestra investigación actual está centrada en continuar evolucionando esta metodología en sus distintas etapas para tratar de manera adecuada datos ciclo-estacionarios. Recientemente se han publicado algunas alternativas para la ejecución de MSNM. En concreto, relativas a la construcción del modelo

de calibración, detección de anomalías [9][10], y diagnóstico de las mismas [11].

El objetivo del presente trabajo es evaluar y comparar dichas alternativas, así como otras no publicadas utilizando UGR'16. La Sección II muestra los resultados preliminares obtenidos para algunas de las variaciones de MSNM descritas, mientras que la Sección III presenta las conclusiones sobre el trabajo actual y futuro.

II. Resultados preliminares

Lo habitual es disponer de registros de netflow capturados en distintos sensores. En la práctica, las capturas de UGR'16 proceden un único router (sensor). Sin embargo, estas capturas se pueden organizar en routers virtuales, atendiendo a los rangos de IP implicados. En este estudio hemos divido la captura original [1] en tres routers virtuales: VR1, VR2 y VR3.

La Fig. 2 muestra los resultados obtenidos, en los que hemos estudiado el efecto de:

- La **organización** de los datos antes de la fase de pre-procesamiento. Estándar: *i)* Esquema **G**, consiste en agregar las variables extraidas de los distintos routers [VR1+VR2+VR3] (captura original), y *ii)* Esquema **S**, consiste en concatenar las variables de los distintos routers [VR1 VR2 VR3]. Jerárquica: Esquema **H**, consiste en crear modelos a partir de las variables de los distintos routers y combinar los estadísticos obtenidos en distintos niveles [9].
- El **pre-procesamiento**. *i)* Auto escalado (AS), y *ii)* XPA, que es una variación de AS en la que se utilizan más observaciones para calcular la media y la desviación típica [12]).

Las alternativas MSNM se evalúan haciendo uso del área bajo la curva (del inglés, AUC) [13], [14]. Esta es una medida típica para clasificadores de una clase, donde se encuentran los IDSs basados en detección de anomalías, como MSNM. Un clasificador ideal presenta un $AUC = 1$, mientras que un $AUC \approx 0{,}5$ denota que el clasificador es similar a un clasificador aleatorio [15].

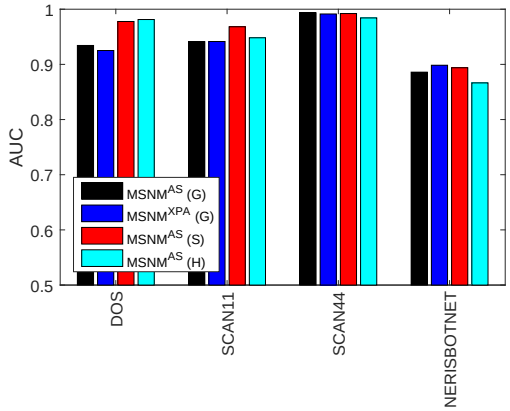


Figura 2. Evaluación de distintas versiones de MSNM. Valores del área bajo la curva (del inglés, AUC) para los ataques introducidos en UGR'16.

III. Conclusiones y trabajo futuro

Este trabajo evalúa y compara diferentes propuestas de aplicación de la metodología MSNM sobre datos ciclo-estacionarios (UGR'16).

El objetivo final es identificar cuál o cuáles de las propuestas presentadas ofrecen mejores resultados en términos de detección (idealmente, alto porcentaje de verdaderos positivos y bajo número de falsos positivos).

Agradecimientos

Este trabajo de investigación está financiado por el Ministerio de Economía y Competitividad y los fondos FEDER a través de los proyectos TIN2014-60346-R y TIN2017-83494-R.

Referencias

[1] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón Sánchez, "UGR'16: a new dataset for the evaluation of cyclostationarity-based network IDSs," *Computer & Security*, November 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817302353

[2] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based multivariate statistical network monitoring for anomaly detection," *Computers & Security*, vol. 59, pp. 118–137, 2016.

[3] P. Nomikos and J. F. MacGregor, "Multivariate Statistical Process Control Charts for Monitoring Batch Processes," *Technometrics*, vol. 37, no. 1, pp. 41–59, 1995.

[4] A. Ferrer, "Latent Structures-Based Multivariate Statistical Process Control: A Paradigm Shift," *Quality Engineering*, vol. 26, no. 1, pp. 72–91, 2014.

[5] B. M. Wise, N. L. Ricker, D. F. Veltkamp, and B. R. Kowalski, "Theoretical basis for the use of principal component models for monitoring multivariate processes," *Process Control and Quality*, vol. 1, no. 1, pp. 41–51, 1990.

[6] C. F. Alcala and S. J. Qin, "Reconstruction-based contribution for process monitoring," *Automatica*, vol. 45, no. 7, pp. 1593–1600, 2009.

[7] J. Camacho, "Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models," *Journal of Chemometrics*, vol. 25, no. 11, pp. 592–600, 2011.

[8] J. Camacho, P. García-Teodoro, and G. Maciá-Fernández, "Traffic Monitoring and Diagnosis with Multivariate Statistical Network Monitoring: A Case Study," *IEEE Security & Privacy International Workshop on Traffic Measurements for Cybersecurity (WTMC 2017)*, 2017.

[9] G. Maciá-Fernández, J. Camacho, P. García-Teodoro, and R. A. Rodríguez-Gómez, "Hierarchical PCA-based multivariate statistical network monitoring for anomaly detection," in *Information Forensics and Security (WIFS), 2016 IEEE International Workshop on*. IEEE, 2016, pp. 1–6.

[10] J. Camacho, G. Maciá-Fernández, N. M. Fuentes-García, and E. Saccenti, "Semi-supervised multivariate statistical network monitoring for learning security threats," *IEEE Transactions on Information Forensics and Security*, vol. PP, pp. 1–1, 01 2019.

[11] M. Fuentes-García, G. Maciá-Fernández, and J. Camacho, "Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control," *Chemometrics and Intelligent Laboratory Systems*, vol. 172, pp. 194 – 210, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169743917302046

[12] N. M. Fuentes-García, González-Martínez, G. Maciá-Fernández, and J. Camacho, "PARAMO: Enhanced Data Pre-processing in Batch Multivariate Statistical Process Control," *Submitted to Journal of Chemometrics*, 2019.

[13] C. E. Metz, "Basic principles of roc analysis," *Seminars in Nuclear Medicine*, vol. 8, no. 4, pp. 283 – 298, 1978. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001299878800142

[14] J. Hanley and B. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve," *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[15] E. Alpaydin, *Introduction to Machine Learning*, 2nd ed. The MIT Press, 2010.

DarkNER: A Platform for Named Entity Recognition in Tor Darknet

Mhd Wesam Al-Nabki
Dept. IESA.
Universidad de León
Researcher at INCIBE
mnab@unileon.es

Eduardo Fidalgo
Dept. IESA.
Universidad de León
Researcher at INCIBE
eduardo.fidalgo@unileon.es

Javier Velasco Mata
Dept. IESA.
Universidad de León
Researcher at INCIBE
jvelm@unileon.es

***Abstract*—In this paper, we introduce DarkNER, an application of Named Entity Recognition (NER) based on neural networks to identify six categories of named entities: Location, Person, Products, Corporation, Group, and Creative-work, in onion domains on the Tor network. The presented NER model is trained on the W-NUT-2017 dataset and tested on manually tagged samples of Tor hidden services. The experiments show the adaptability and effectiveness of neural networks models in detecting new textual entities, such as drugs names and weapons brands. The proposed application could help the authorities in filtering and monitoring the contents of the Tor domains.**

***Index Terms*—Named Entity Recognition, Darknet, Tor Network, Cybersecurity, Hidden Services**

Contribution Type: *Ongoing research*

Named Entity Recognition (NER) aims to identify different types of entities, such as a location, people names or products, within a given text. Those entities can be useful for several Natural Language Processing (NLP) tasks such as web contents filtering and monitoring, entity-based trend detection, and content mining [1], [2]. Recently, The Onion Router (Tor) network has become a safe shelter for practicing suspicious activities on the Internet, like drugs trading, weapons markets, or child pornography forums, far from the authorities' monitoring tools [2]. The conventional methods for monitoring the online textual contents, such as using predefined lists of keywords, allow to detect those keywords only, it is hard to maintain or keep it updated with emerging terms. Moreover, the problem becomes more complicated when those lists need to be created and updated in several languages. Hereafter, it would be useful to build an automatic system to recognize textual entities, with the capability to adapt to the dynamic content of the Tor domains.

Building a NER system is challenging due to the limited amount of supervised training samples and the possibility of having multiple meanings for a given word. Besides, the quality of the input text has a significant impact on the performance of the system . For example, a well-structured text quoted from a newspaper, where the capital letters and the punctuation marks are carefully reviewed, would be easier to understand rather than a short text written in slang words that may contain syntax mistakes.

In this paper, we present DarkNER, a platform to detect six categories of named entities (NE): Locations, Person, Creative-work, Group, Product, and Corporation, in the hidden services of the Tor network. In particular, we used the neural network model proposed by Aguilar et al. [8], since achieves the state of the art performance on W-NUT-2017 dataset[1]. To the best of our knowledge, the W-NUT-2017 is the most recent dataset for NE for noisy user-generated text.

Thanks to the experience we earned during labeling Darknet Usage Text Addresses (DUTA) dataset [3], we observed that the W-NUT-2017 dataset mimics the nature of the contents of the Tor network domains in terms of the quality of the text. Both datasets hold noisy user-generated text, which is rich with slang words, acronyms, abbreviations, together with the presence of emerging terms that have never been seen before. We propose to use the trained NER model of Aguilar et al. to detect NE on onion domains sampled from DUTA dataset.

The rest of the paper is organized as follows: Section I presents the related work. Then, Section II introduces the used neural network structure. After that, we explore the conducted experiments on DUTA samples in Section III. Finally, Section IV presents the conclusions by pointing out to our ongoing research on the field.

I. Related Work

The NER task has been a hot research topic for a long time. Before the rise of the deep learning techniques, the proposed methods mainly depended on manually extracted features from the input text. McCallum et al. [4] used handcrafted features, such as words prefix or suffix, and capital letters. However, the automatic feature extraction carried out using deep learning has pushed the state-of-the-art score strongly in NER systems. Lample et al. [5] proposed a neural network model with F1 score of 90.94% on Conll2003[2] dataset. Ma et al. [6] designed a model similar to Lample, but they used a Convolutional Neural Network (CNN) instead of Bi-LSTM for the characters sequences and had an F1 score of 91.21%. Although the neural network models have a high F1 score, the performance drops sharply in the case of the user-generated text like users' tweets on Twitter. Von Däniken et al. [7] used Transfer Learning (TL) and achieved F1 score of 40.78%. Aguilar et al. [8] presented a neural network model and trained it over the W-NUT-2017 dataset with F1 score of 41.86%.

II. Methodology

In this section, we describe briefly the neural network model proposed by Aguilar et al. [8]. The model extracts

[1]http://noisy-text.github.io/2017/emerging-rare-entities.html
[2]https://www.clips.uantwerpen.be/conll2003/ner

features from (i) word characters: an orthographic encoder is used to represent the characters. The encoded characters are embedded into a $\mathbb{R}^{d\times t}$ embedding space, where d is the dimension of the features per character and t denotes the word length threshold. Then, the result is passed into 2-stacked convolutional layers with a global average pooling. (ii) Word context: each word is represented using two codifications. First, pre-trained words embedding to capture latent semantics of words. Second, an embedding for the part-of-speech (POS) tags that were generated using the CMU Twitter POS tagger[3]. These embeddings are concatenated to form the final representation of the input word and passed into a Bidirectional Long Short-Term Memory (Bi-LSTM). (iii) A gazetteer, an external resource of knowledge. The gazetteer vector of a single word is a binary vector of n dimensions whereas n refers to the number of the categories in the dataset, i.e. 6 dimensions in the W-NUT-2017 dataset. The length of the gazetteer is equal to the size of the dataset's vocabulary. Next, the extracted features are fed into a multi-task network. The first task has a sigmoid activation function to identify whether the input token is an entity or not, while the second one has a softmax activation function to decide the category of the tag. Finally, the model is attached with a Conditional Random Field (CRF) to account for the sequential constraints in the input text.

III. Experimental Results

A. Tor Domains Entities Recognition

The W-NUT-2017 dataset has six types of NE that are encoded using: Begin, Inside, and Outside (BIO) tags such that the *B* in BIO refers to the beginning of a tag, the *I* refers to inside of a tag, and the *O* refers to non-entity words. Since there is no training dataset for the Tor hidden services, we used the W-NUT-2017 dataset for training. Later, to test the performance of the trained model, we manually labeled the NE tags of 15 onion domains which were randomly sampled from the categories Drugs and Violence in DUTA dataset. We found that the trained model can detect drugs names and weapons brands as *Products*, marketplaces names as a *Corporation*, names of cities and countries as *Location*, and people names as *Person*. Table I reports the performance of the model in terms of Precision, Recall, and their harmonic mean F1 score measures along with examples of the recognized entities per category.

TABLE I
EXAMPLES OF THE RECOGNIZED NE IN 15 SAMPLES ONION DOMAINS ALONG WITH PRECISION/ RECALL AND THEIR HARMONIC MEAN F1 SCORE

Categories	% Precision	% Recall	% F1 Score	NE Examples
Corporation	35.19	17.76	23.60	Alpha Pharma, Heckler & Koch
Creative-work	37.36	12.01	18.18	Danaucolt Ghost
Group	49.43	21.08	29.55	American brands
Location	53.80	52.94	53.37	Barcelona, Amsterdam
Person	68.35	51.37	58.65	Alex Grey, Vin Mariani
Product	56.48	20.85	30.46	marijuana, Purple Kush, S&W 10mm, Ruger M77
Average	56.72	30.47	39.65	

The results show that the system is capable of detecting NE with a low recall but with high precision, relatively. The decrease in the recall value could reflect the difficulty of recognizing emerging or rare terms in the test set. The model of Aguilar et al. depended on an external resource of knowledge that was built manually to fit Twitter dataset, and this could justify this decrease. To overcome this limitation, our ongoing research focuses on replacing the gazetteer with a dynamic feature that could be calculated based on the given training set, and consequently, making the network end-to-end, without any dependency of external resources of knowledge.

[3]https://www.cs.cmu.edu/~ark/TweetNLP/

IV. Conclusions and Future Work

In this paper, we demonstrated the effectiveness of a NER system in detecting textual entities in the Tor onion domains. Also, we pointed out the drawback of the state of the art model which allowed us to focus our ongoing research on making the model as an end-to-end network. We found that even if we train the NER model with a dataset that is not related to the final application, it is still capable of detecting useful entities that are related to suspicious activities. In addition to introducing a new dynamic feature that replaces the gazetteer, we plan to build a customized NER model for Tor domains that might help the authorities in monitoring and analyzing the Tor Darknet content. Hereafter, those automatically recognized entities can serve as an input for our previous work in [1] to build a fully-automatic tool for detecting the emerging products in the Tor Darknet.

Acknowledgement

This research is supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01. We acknowledge NVIDIA Corporation with the donation of the Titan XP GPU used for this research.

References

[1] Al-Nabki, M., Fidalgo, E., Alegre, E., and Gonzalez-Castro, V., "Detecting Emerging Products in Tor Network Based on K-Shell Graph Decomposition", *III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC)*, vol. 1, no. 1, pp. 24-30, 2017.

[2] Al-Nabki, M., Fidalgo, E., Alegre, E., and Fernández-Robles, L. "Torank: Identifying the most influential suspicious domains in the Tor network". *Expert Systems with Applications*, vol. 123, pp.212–226, 2019.

[3] Al-Nabki, M., Fidalgo, E., Alegre, E., and de Paz, I. "Classifying Illegal Activities on Tor Network Based on Web Textual Contents", *European Chapter of the Association for Computational Linguistics, 2017.*

[4] McCallum, A., and Li, W. (2003, May). "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons". *In Proceedings of the seventh conference on Natural language learning at HLT-NAACL. Association for Computational Linguistics.*

[5] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. "Neural architectures for named entity recognition". *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.As-sociation for Computational Linguistics*, 2016, pp. 260–270.

[6] Ma, X., and Hovy, E. (2016). "End-to-end sequence labeling via bi-directional lstm-cnns-crf." *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* pp. 1064-1074

[7] von Däniken, P., and Cieliebak, M. (2017, September). "Transfer learning and sentence level features for named entity recognition on tweets." *In Proceedings of the 3rd Workshop on Noisy User-generated Text* pp. 166-171.

[8] Aguilar, G., Maharjan, S., Monroy, A. P. L., and Solorio, T. (2017). "A Multi-task Approach for Named Entity Recognition in Social Media Data". *In Proceedings of the 3rd Workshop on Noisy User-generated Text*

A Review of Anomaly-based Exploratory Analysis and Detection of Exploits in Android

Guillermo Suarez-Tangil, Santanu Kumar Dash
University College London (UK)

Pedro García-Teodoro
University of Granada (Spain)

José Camacho
University of Granada (Spain)

Lorenzo Cavallaro
Royal Holloway, University of London (UK)

***Abstract*—Smartphone platforms are becoming increasingly complex, which gives way to software vulnerabilities that are difficult to identify. In this work we present *CoME*, an anomaly-based methodology aiming at detecting software exploitation in Android systems. *CoME* models the normal behavior of a given software component or service and it is capable of identifying any unanticipated behavior. To this end, we first monitor the normal operation of a given exploitable component through lightweight virtual introspection. Then, we use a multivariate analysis approach to estimate the normality model and detect anomalies. We evaluate our system against one of the most critical vulnerable and widely exploited services in Android, *i.e.*, the `mediaserver`. Results show that our approach can not only provide a meaningful explanatory of discriminant features for illegitimate activities, but it can also be used to accurately detect malicious software exploitations at runtime.**

Tipo de contribución: *Investigación ya publicada.*

I. Introduction

The increasing popularity of tablets and smartphones has led to an exponential growth of the number of risks and vulnerabilities in them. As malware becomes sophisticated, it is infeasible to eke out the malicious action from the app in a controlled setting. Thus, the best option is to resort an online monitor that keeps continuously checking for anomalous actions that are indicative of malice. The difficulty in designing any anomaly detection system is coming up with a model of normality capable of scaling as the number of features increases. Additionally, most anomaly detection techniques yield black box models with an uninterpretable linkage between input and output data.

In this paper, we introduce a novel anomaly-based approach. We advocate the use of Multivariate Statistical Network Monitoring (MSNM) [1] to address aforementioned problems. In particular, we apply the NSNM approach to detect anomalous actions intending to cause some harm to Android systems. To this end, we show how the proposed anomaly detector effortlessly extracts events and features of interest by using the well-known Principal Component Analysis (PCA) as a building block. We also show that MSNM not only provides a means to easily process a multitude of features but also provides an easily interpretable model of normality and anomaly. When using MSNM, we do not need to perform a feature selection on first place, avoiding the risk of discarding useful information for anomaly detection. Our main contributions are:

- The proposed anomaly-based methodology relies on the estimation of a *normality model* for a given Android service. For that, the normal expected operation of a given service is first estimated by collecting system information at multiple levels of granularity. CopperDroid [2] will be used as the monitoring system to collect such information.
- The multivariate statistical approach proposed in [1] (MSNM) supports the overall anomaly detection process. Despite the inherent capabilities of such techniques to handle a number of features of diverse nature and origin, they are not commonly used for malware detection for the time being, which constitutes a novelty in the field.
- A relevant and well publicised vulnerability in the Android libraries is used to test our proposal: the *Stagefright* bug[1]. For that, we first estimate the normal operating conditions (NOC) of the system's `mediaserver`—the sub-system responsible for processing media file in Android. Then, we feed the `mediaserver` with both legitimate and crafted files aimed at exposing the vulnerability. By analyzing the associated behavior in all the cases, we shall show the capacity of MSNM in detecting the malicious ones.

We next present a summary of *CoME*. We refer the reader to [2], [1], [3] for extended details.

II. CoME: Anomaly Detection in Android

In this section, we present a novel proposal aimed at detecting anomalies in Android platforms. For that, we used the following methodology:

- *Monitoring*: The target system is monitored in order to collect information regarding the overall activity taking place on it. This information is parameterized to represent (usually in terms of a feature vector) the 'state' or 'behavior' of the system at a given instant. This way, a sequence of observations are disposed as the system operates and evolves over time.
- *Training*: Provided a set of observations corresponding to the 'normal/legitimate' operation of the system, a 'normality' model is first estimated

[1] http://www.androidcentral.com/stagefright

by considering some mathematical theory (*e.g.*, Markov models, fuzzy theory, neural networks, etc.).

- *Detection*: Further observations gathered from the monitored system are subsequently evaluated by using the 'normality' model in order to estimate a deviation score. From that, we conclude that an anomaly is occurring if the deviation score obtained surpasses a given threshold. Otherwise, the observation (or sequence or observations) analyzed is classified as 'normal'.

CoME is based on the combined use of two core building blocks: CopperDroid and MEDA/MSNM. The first one is used as a monitoring tool for Android devices [2]. The second one, MSNM, is part of the functionality programmed in the Multivariate Exploratory Data Analysis (MEDA) toolbox [1] as a detection approach proposed and successfully applied to detect anomalies in network environments.

III. Experimental Evaluation

For our evaluation, we collected a dataset of normal media files (*goodware*) and a dataset of crafted media files (*malware*). For the *goodware*, we queried GooglePlay and retrieved about 15,000 apps from which we extracted all media files and we randomly selected a total number of 264 MP4 files. For the *malware*, we obtained a number of crafted media files released by Zimperium exploiting recent vulnerabilities affecting most Android versions (e.g., CVE-2015-1538 or CVE-2017-0809).

CopperDroid generates a set of features at a constant sampling rate during the execution of media files. For the experimentation of this paper, we consider a total of 692 features. These features contained all behaviors reported after monitoring the `mediaserver` process. Our data consisted of a total of 71,336 time observations of the 692 features derived from the execution of 298 files (264 goodware and 34 malware samples). This execution was of variable duration.

We mainly observed features related to: (i) *File Access* to different libraries and data files, (ii) low-level *System Calls*, and (iii) different *Binder* transactions. With these features we then leverage two statistical metrics called D-st and Q-st from MEDA [1] as detailed in [3] to build the following systems:

Off-line system: for aiding security experts on the identification of relevant discriminant features. The result obtained is shown in Figure 1. We can see that the Q-st is highly efficient in discriminating goodware and malware. Control limits are adjusted at a 99% confidence level, so that 99 out of 100 calibration goodware files yield statistics below the limits. It can be seen that this adjustment holds for the tested goodware, since only one test goodware file out of 64 exceeds the limits, and they are exceeded by a reduced margin.

On-line system: for accurate runtime monitoring, analysis, and detection. Results are shown in Figure 2, where we included the results three detection systems: two that leverage the D-st and Q-st metrics, and a

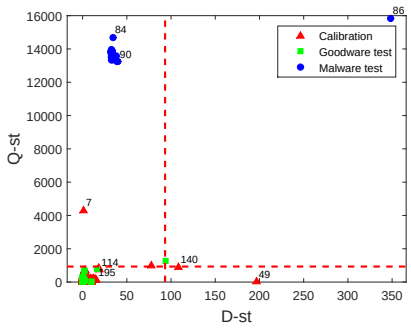


Fig. 1: MSNM monitoring chart: Q-st vs D-st. Each point represents a file.

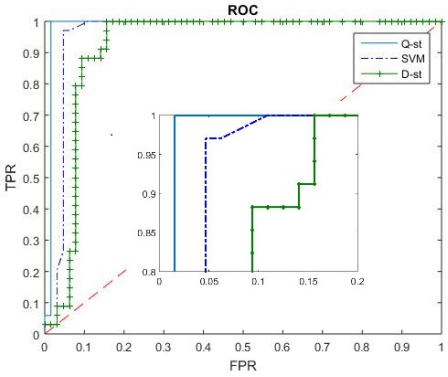


Fig. 2: ROC curves for the MSNM statistics in test files.

third one-class support vector machine (OCSVM) classifier presented as reference. According to these ROC curves, the detection performance of the online MSNM system is impressive. While the D-st is outperformed by the one class SVM classifier, the Q-st clearly outperforms the latter. Overall, the performance of the online MSNM system shows that we can make a clear discrimination between malware and goodware — except for test goodware file number 51. Interestingly, this false positive was also highlighted in our off-line analysis as described in the extended version [3].

IV. Conclusion

Using lightweight introspection to perform transparent behavioral analysis at runtime, we have shown how anomaly-based multivariate systems can effectively be used to identify technical software exploitations based on unknown vulnerabilities. Our extended work is described in [3] and it has been partially supported by the Spanish MIMECO and FEDER funds TIN2014-60346-R and TIN2017-83494-R, and the UK EPSRC grant EP/L022710/1.

References

[1] J. Camacho, A. Pérez-Villegas, R. Rodríguez-Gómez, E. J.-M. nas, Multivariate Exploratory Data Analysis (MEDA) Toolbox for Matlab, Chemometrics and Intelligent Lab Systems.

[2] K. Tam, S. Khan, A. Fattori, L. Cavallaro, CopperDroid: Automatic Reconstruction of Android Malware Behaviors, in: NDSS, 2015.

[3] G. Suárez-Tangil, S. K. Dash, P. García-Teodoro, J. Camacho, L. Cavallaro, Anomaly-based exploratory analysis and detection of exploits in android mediaserver, IET Information Security 12 (5) (2018) 404–413.

Un resumen de "Aplicación de técnicas de compresión de información a la identificación de anomalías en fuentes de datos heterogéneas: análisis y limitaciones"

Gonzalo de la Torre-Abaitua
Escuela Politécnica Superior
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
gonzalo.torre@estudiante.uam.es

Luis F. Lago-Fernández
Escuela Politécnica Superior
Departamento de Ingeniería Informática
Universidad Autónoma de Madrid
luis.lago@uam.es

David Arroyo
Instituto de Tecnologías Físicas
y de la Información - ITEFI
Consejo Superior de Investigaciones
Científicas - CSIC
david.arroyo@iec.csic.es

***Resumen*—La interconexión y heterogeneidad de los diferentes sistemas de información de la actualidad hacen que la ciberseguridad haya evolucionado desde la clásica clasificación basada en logs y listas, hacia enfoques de carácter integral que consideran otros factores como las redes sociales, foros de discusión o mensajes de correo. Esto hace necesario disponer de un mecanismo que pueda analizar de forma agnóstica esta amplia variedad de registros de actividad y de eventos de seguridad. Partiendo de la base de que todos estos registros contienen información textual, hemos explorado el uso de la distancia de compresión normalizada (NCD) para establecer una metodología capaz de trabajar con fuentes heterogéneas de información. En este sentido, hemos partido de una contribución propia en el campo de la detección de anomalías en HTTP y la hemos extendido a la detección de dominios generados mediante DGAs (Domain Generation Algorithms) y de spam en SMS. Los diversos experimentos confirman que la metodología tiene un rendimiento aceptable de acuerdo con el estado del arte. En este punto, cabe subrayar la ventaja de nuestra propuesta en términos de simplicidad y de capacidad de ser aplicada de modo general, al margen del formato de codificación de los datos. Asimismo, también se ha observado que se alcanzan resultados positivos utilizando menos datos de entrenamiento que los usados en otras aproximaciones a los tres problemas considerados.**

***Index Terms*—NCD, spam, DGA, IDS, detección de anomalías, fuentes heterogéneas de datos**

Tipo de contribución: *Investigación ya publicada (límite 2 páginas)*

I. Introducción

Hoy en día vivimos en un mundo interconectado en el que se han ampliado el tipo y variedad de dispositivos conectados a Internet. Esto ha provocado que la ciberseguridad se haya convertido en un área compleja con grandes retos [1]. Así, no basta con realizar la búsqueda de amenazas de seguridad mediante el análisis de logs o en base a listas blancas/negras de actividad en nuestra red de trabajo. En efecto, se ha de tener en cuenta que la detección temprana de ciberamenazas puede demandar la observación y procesamiento de otras fuentes de información. De hecho, recientemente se han efectuado esfuerzos significativos orientados a identificar fugas de información en compañías, a identificar vulnerabilidades de los sistemas de información y de comunicación de una organización, o a descubrir zero-days mediante la exploración de fuentes abiertas de datos [2]. Como punto común, estos eventos pueden ser interpretados como texto codificado como lenguaje natural o, al menos, estructurado. Esta característica permite diseñar una metodología para comparar eventos de diversa naturaleza mediante un mismo tipo de procedimiento para el tratamiento de información. En específico, en [3], [4] hemos propuesto el uso de técnicas de compresión de información como base de tal metodología. En concreto, en nuestros trabajos previos hemos mostrado la idoneidad de la NCD [5] como métrica de detección de anomalías.

Frente a otras metodologías que se centran en un único tipo de fuentes de datos [6], nuestro trabajo ha estado centrado en el desarrollo de una metodología basada en la NCD de forma que la caracterización de los datos se puede efectuar de modo automático e independientemente del tipo de dato considerado. Con el objeto de analizar tanto la bondad como el carácter neutral del método, en nuestros trabajos previos hemos llevado a cabo la validación del método en los problemas de la detección de dominios DGA, de spam en SMS [3] y de anomalías en peticiones web [4]. A continuación, explicamos la metodología considerando como casos de uso la identificación de dominios DGA, la detección de spam en SMS y la detección de anomalías en peticiones HTTP.

II. Metodología y resultados

Tradicionalmente la detección de anomalías se ha planteado de manera diferente en función del tipo de problema. Así, la detección de DGAs se ha abordado bien mediante el análisis del tráfico DNS, o bien considerando el nombre de dominio [7]. En el caso de la detección de spam normalmente se efectúa la extracción de distintas características para, posteriormente, aplicar diferentes algoritmos y técnicas de análisis de texto [8]. Mientras que en las peticiones web se suelen extraer características de la petición y aplicar diferentes algoritmos de aprendizaje automático [9].

Nuestra metodología, por el contrario, plantea el uso de un único formalismo que no depende del tipo de problema. Parte del tratamiento de una muestra dada (petición HTTP, nombre de dominio, mensaje SMS) como una cadena de caracteres, de forma que independientemente del tipo de problema todos son tratados como texto. Esta cadena de texto es codificada mediante un vector de características que mide la distancia de

dicha cadena a distintas agrupaciones de cadenas (generadores de características). Para ello, el *dataset* se divide en dos grupos disjuntos. El primero se utiliza para obtener k generadores de características, conjuntos de cadenas pertenecientes a la misma clase. El segundo contiene cada una de las cadenas que se utilizarán para entrenar el clasificador final. Cada una de estas cadenas es representada mediante un vector de k componentes, en el que cada componente representa la distancia entre la cadena y uno de los k generadores de características. La distancia entre la cadena b y el generador a viene dada por [10]:

$$D(a,b) = \frac{C(b|a)}{C(b)} = \frac{C(ab) - C(a)}{C(b)} \quad (1)$$

donde ab representa la concatenación de a y b y $C(x)$ es el tamaño en bytes de x tras aplicar el algoritmo de compresión gzip.

II-A. Preparación de los datos

Para el problema de identificación de dominios DGA hemos utilizado un *dataset* compuesto por 800000 dominios normales y 800000 dominios DGA pertenecientes a diferentes familias de malware. De estos, hemos utilizado 1600 dominios para entrenar el clasificador y el resto para construir los generadores de características. En el problema de detección de spam, hemos utilizado 1494 mensajes (747 normales y 747 spam). El grupo de entrenamiento está formado por 200 mensajes de cada clase, quedando los 1094 restantes para construir los generadores de características. Finalmente, para el problema de detección de anomalías en URLs hemos usado 9600 peticiones normales y anómalas. De estas, 800 de cada clase forman el grupo de entrenamiento y las 4000 restantes de cada clase se usan para construir los k generadores.

II-B. Resultados

Una vez que los datos están preparados hemos aplicado una máquina de vectores de soporte (SVM) con kernel RBF y validación cruzada para la clasificación. En todos los casos hemos realizado experimentos para $k \in \{8, 16, 80, 160\}$. Los resultados obtenidos se pueden ver en la tabla I. Para spam y anomalías web se obtienen resultados comparables al estado del arte, donde se indica un $accuracy$ (ACC) entre 0,85 y 0,97, mientras que en DGA son inferiores a otros publicados en la literatura con un ACC de 0,91 [7]. Sin embargo, nuestro planteamiento tiene la ventaja de que no es necesario realizar ingeniería de características ni ajustar múltiples parámetros.

III. Conclusiones

En este trabajo hemos explorado un procedimiento que utiliza la NCD para construir un conjunto de atributos que habilita la clasificación de datos mediante máquinas de vectores de soporte. Dicho procedimiento ha sido aplicado en diferentes problemas heterogéneos que pueden ser representados como texto. En el caso de la detección de spam y la detección de anomalías en URLs, el rendimiento alcanzado es positivo y establece la base para ampliar el uso del mecanismo estudiado a otras tareas relevantes dentro del análisis de lenguaje natural y/o estructurado. Sin embargo en el caso de la detección de dominios DGA el rendimiento, sin ser malo, se muestra inferior al divulgado en la literatura. Aún así, la simplificación en la fase de extracción de características y el menor número de parámetros a entrenar e hiperparámetros a fijar compensan el inferior rendimiento. A su vez, cabe recordar que estos rendimientos se han alcanzado utilizando un número menor de patrones para entrenar el modelo. Igualmente, es reseñable el hecho de que para los casos estudiados el mecanismo aplicado ha sido el mismo independientemente de la naturaleza intrínseca de cada problema concreto.

Tabla I
Resultados *spam* SMS, DGA y anomalías URL. k, número de atributos; Acc., *accuracy*; AUC, área media bajo la curva ROC.

k	DGA		spam		URL	
	Acc.	AUC	Acc.	AUC	Acc.	AUC
8	0,717	0,785	0,783	0,85	0,85	0,93
16	0,77	0,84	0,827	0,89	0,89	0,95
80	0,82	0,88	0,888	0,95	0,95	0,94
160	0,795	0,869	0,9	0,96	0,94	0,97

Los resultados obtenidos muestran la versatilidad y polivalencia de la metodología y son un indicador de su potencial extensibilidad a otras tareas de clasificación binaria, como podrían ser, clasificación de textos con diferentes longitudes y características, clasificación de texto estructurado de diferente longitud y clasificación de texto de lenguaje natural.

Agradecimientos

Este trabajo ha sido financiado por la Comunidad de Madrid (España) dentro del proyecto CYNAMON (P2018/TCS-4566) con apoyo de fondos FSE y FEDER de la Unión Europea, y por el proyecto MINECO/FEDER TIN2017-84452-R del Gobierno español.

Referencias

[1] Y. Harel, I-B. Gal e Y. Elovici: "Cyber Security and the Role of Intelligent Systems in Addressing Its Challenges", en *ACM Trans. Intell. Syst. Technol.*, vol. 8, n. 4, pp. 1-49, 2017.

[2] C. Sabottke, O. Suciu y T. Dumitraş: "Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-world Exploits", en *Proceedings of the 24th USENIX Conference on Security Symposium*, pp. 1041-1056, 2015.

[3] G. de la Torre-Abaitua, L-F. Lago-Fernández y D. Arroyo: "Aplicación de técnicas de compresión de información a la identificación de anomalías en fuentes de datos heterogéneas: análisis y limitaciones" en *XV Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, 2018

[4] G. de la Torre-Abaitua, L-F. Lago-Fernández y D. Arroyo: "On the application of Compression Based Metrics to identifying anomalous behaviour in web traffic" en *Logic Journal of the IGPL*, Artículo Aceptado (Pendiente de publicación)

[5] R. Cilibrasi y P. Vitanyi: "Automatic Extraction of Meaning from the Web" en *IEEE International Symposium on Information Theory*, pp. 2309-2313, 2006.

[6] S. Nilizadeh, F. Labreche, A. Sedighian, A. Zand, J-M. Fernandez, C. Kruegel, G. Stringhini y G. Vigna: "POISED: Spotting Twitter Spam Off the Beaten Paths", en *CoRR*, pp. 1159–1174, 2017.

[7] P. Lison y V. Mavroeidis: "Automatic Detection of Malware-Generated Domains with Recurrent Neural Models", en *CoRR*, 2017.

[8] M. Prilepok, P. Berek, J. Platos y V. Snasel: "spam detection using data compression signatures", en *Cybernetics and Systems*, vol. 44, pp. 533-549, 2013.

[9] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández y E. Vázquez: "Anomaly-based network intrusion detection: Techniques, systems and challenges" en *Computers and Security*, vol. 28, pp. 18-28, 2009

[10] R. Cilibrasi y P. Vitányi: "Clustering by compression", en *IEEE Transactions on Information Theory*, vol. 51, n. 4, pp. 1523-1545, 2005.

[11] T-A. Almeida, J-M-G. Hidalgo y A. Yamakami: "Contributions to the Study of SMS Spam Filtering: New Collection and Results", en *Proceedings of the 11th ACM Symposium on Document Engineering*, pp. 259-262, 2011.

A Review of "What did Really Change in the new App Release?"

Paolo Calciati[♠♡], Konstantin Kuznetsov[♣], Xue Bai[♢], Alessandra Gorla[♠]
[♠]*IMDEA Software Institute, Spain* [♡]*Universidad Politécnica de Madrid, Spain* [♣]*CISPA, Saarland University, Germany* [♢]*Beijing Institute of Technology, China*

***Abstract*—As the mobile app market is evolving at a very fast pace, users and market managers might have a hard time understanding what really changed in a new release and whether updating the app is recommendable or could pose a security and privacy threat. We propose a ready-to-use framework to analyze the evolution of Android apps. Our framework extracts and visualizes various information —such as how an app uses sensitive data, which third-party libraries it relies on, etc.— and combines it to create a comprehensive report on how apps evolve.**

We perform an empirical study on 235 applications using our framework. Our analysis reveals that Android apps tend to have more leaks of sensitive data over time, and that API calls tend to have the corresponding dangerous permission already granted when added to the code.

***Index Terms*—Android; app evolution; behavior change**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

In this paper we present the results of our research that we previously published at the Mining Software Repositories (MSR) conference in 2018 [1].

To remain appealing for end-users and avoid them migrating to competing apps, developers have to continuously update their apps. They have to provide new features, and address bug fixes as fast as possible, causing Android apps to have a very frequent release cycle. This can cause problems to both market managers – such as the Google Play Store – and final users. The former need to analyze every version before publishing it, while the latter receive updates for the apps installed on their devices transparently, as by default the Android system notifies them only when there are substantial changes in the list of permissions that the app requests. Most users, however, cannot easily understand how the behavior of an app changed with a new app release, as most of the changes happen beyond the user interface and the list of requested permissions, which is what users can easily analyze.

This paper presents Cartographer, a ready-to-use framework for users and market managers to analyze the evolution of an Android application. Cartographer extracts and visualizes various information: 1) it shows how an app uses sensitive data, thanks to a custom static data flow analysis; 2) it aims to identify the list of third-party libraries that the app uses, even if obfuscated; 3) it extracts the network traffic to have a list of hosts the application talks to; 4) it statically extracts sensitive Android APIs the application uses. Cartographer runs these analyses separately and combines the results to create a comprehensive report on how the app evolved.

While many of the existing research papers regarding Android application evolution focus only on dangerous permissions, in this paper we take a wider view to have a more in-depth understanding of the changes across different releases. We use Cartographer to empirically analyze 14,880 releases belonging to 235 applications with at least 50 releases each.

Our analysis reveals that Android applications tend to have more leaks of sensitive data over time, in line with previous literature [2], [3]. However, the growth is largely determined by third-party libraries, used by the apps. Our study also shows that the majority of API calls relative to dangerous permissions are added to the code in releases posterior to the one where the corresponding permission was requested, and that the vast majority of data flows only exists for a limited number of versions.

II. Cartographer

Our framework aims to thoroughly analyze the behavior of an app keeping into account several aspects, each of them requiring a specific feature, such as network traffic or list of data flows. We then visualize the output of each analysis to make it easy for the users to spot any significant difference. The workflow is divided into three logical parts:

a) APK Crawl: The first part aims to retrieve a significant amount of releases for the app of interest, or for a set of releases specifying particular conditions.

b) Information Extraction: The second part comprises different modules, implemented on top of Luigi, which analyze each APK in isolation using various techniques and tools, and report the information for the evolution of the analyzed app, and showing it to the user in the form of a heatmap, such as shown in Figure 1. Each module can be executed separately, though, they are dependent on each other. Modules use the JSON format to exchange information.

Cartographer supports the extraction of the the following data:

- **App Metadata**: the Android *aapt* tool extracts basic data, e.g. package name, version code and permissions;
- **Libraries**: we resort to *LibRadar* to detect third-party libraries used the app, and resort to our own heuristic to complement the information extracted;
- **Apk Content**: we extract the resources contained in the app with *Apktool*, showing both changes in code and UI elements, and in the Android Manifest, such as activities, permissions and services;
- **URLs**: we extract the URLs to which the app connects both by dynamically executing the app using Monkey and logging the network traffic using *tpdump*, and by running *Stringoid* [4] to statically extract constructed URL strings in the app's code. In the second case, Cartographer differentiates whether the strings are in the app code or in a library;

- **API Evolution**: by means of *Soot* [5], we extract the list of API calls related to dangerous permissions that the app makes to the Android framework;
- **Data Leaks**: we use a modified version of *Flowdroid*[6] to identify possible data leaks.

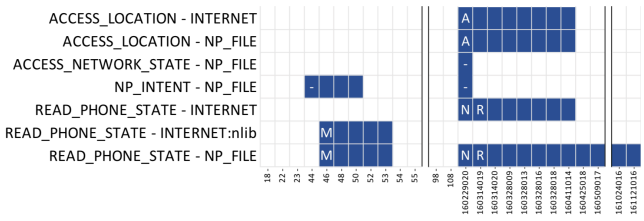


Fig. 1. Information flows in TripAdvisor

c) Data Analysis: In the third part we combine information from previous modules to provide a more in-depth understanding of the changes. One example of combining results from different scripts is Figure 1, where we combine data gathered from the FlowDroid and App Info tasks: in the heatmap we show on the y-axis the data flows of the application, enhancing the data with information regarding the status of the permission needed by the flow source (*A*: permission already asked in previous version, *N*: permission newly asked, -: the flow does not require any dangerous permission, *M*: permission missing, *R*: permission revoked). We can see that the READ_PHONE_STATE to INTERNET flow, which appears in version 160229020, has the READ_PHONE_STATE permission newly asked (N) in the same release. However, the permission is revoked (R) in the following version and never added back. With this information we can understand that, despite finding the data flow with FlowDroid, it can only be exploited in the first version it appeared, as the required permission is no longer requested afterwards. This allowed us to discover that FlowDroid reports unfeasible flows, since the supposedly leaked data is protected by a permission which the application has not requested.

The code of Cartographer is open source and available at: https://github.com/gorla/appmining

III. Empirical Study

For our empirical study we sample 235 applications with at least 50 releases – 14,880 in total – from Androzoo and use Cartographer to see on how Android apps change across different releases. In the following paragraphs we present the results for the research questions we consider.

a) RQ1: How does a new data flow correlate to other changes in the release?: Cartographer can identify 202 new flows in our dataset, out of which 70% leak data from a new source, while 30% leak data from an already leaked source. Only 15% of the new flow sources are protected by a newly asked permission, while for the rest either the leaked data is already used inside the app in the previous version, or developers add over-privileged permissions from the start for a later use. Finally, when a new flow is added, the layout changes 82.67% of the cases: in about one in five cases the addition of a new flow is completely transparent to the user as there are no visible changes in the UI: we see this as a possible privacy threat.

TABLE I
Most common flow patterns

Pattern	Count	Frequency (%)
0+ → 1+ → 0+	314	39.15
0+ → 1+	168	20.95
0+ → (1 → 0)+ → 0+	107	13.35
0+ → (1 → 0)+ → 1+	80	9.98
(1 → 0)+	60	7.49
1+	12	1.50

b) RQ2: How do web domains relate to layout changes?: In the dynamic analysis we identify 57,183 new domain connections; 62.7% of the times there is a layout change, 29.8% there are no changes, and 7.5% the Apk Content analysis did not generate a layout folder. We have similar results for 20,742 URLs identified by Stringoid, with the proportion more in favor of layout changes: 80.1% layout changed, 19.3% layout unchanged and 0.6% unknown. We checked all the domains with VirusTotal and analyzed the 16 reported as malicious by at least 3 VirusTotal sources, ending up with a final list of 8 potentially malicious domains.

c) RQ3: How do information flows evolve during the lifetime of an application? How do third party libraries play a role into the app evolution?: We collected the most common flow patterns (using a 1% frequency threshold) in Table I, using 1 to report the presence of flows and 0 for their absence. The reported patterns count a total of 802 flows, out of which 84% had both a source and a sink inside library code, while the remaining 16% had at least one of them in the app code. We note that while the number of flows tend to increase in third party libraries, it remains fairly constant in the app code.

d) RQ4: How do API Calls Evolve?: We found 1,047 APIs for which the related permission is requested in the same version as the API is added, while it is already requested in a previous version for 9,360 newly added APIs. From our data we can see that it is 9 times more frequent to have the permission already asked when a new API is added. We also discovered that most of the newly added APIs (88%) related to dangerous permissions are added in libraries.

Acknowledgments

This work was supported by the EU FP7-PEOPLE-COFUND project AMAROUT II (n. 291803), by the Spanish project DEDETIS, by the Spanish Government through the SCUM grant RTI2018-102043-B-I00 and by the Madrid Regional projects N-Greens Software (n. S2013/ICE-2731), BLOQUES and MadridFlightOnChip.

References

[1] P. Calciati, K. Kuznetsov, B. Xue, and A. Gorla, "What did really change with the new release of the app?" in *MSR 2018*, 2018, pp. 142–152.
[2] P. Calciati and A. Gorla, "How do apps evolve in their permission requests? a preliminary study," in *MSR 2017*, 2017, pp. 37–41.
[3] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, "Permission evolution in the android ecosystem," in *ACSAC 2012*, 2012, pp. 31–40.
[4] M. Rapoport, P. Suter, E. Wittern, O. Lhoták, and J. Dolby, "Who you gonna call?: analyzing web requests in android applications," in *MSR 2017*, 2017, pp. 80–90.
[5] R. Vallée-Rai, P. Co, E. Gagnon, L. Hendren, P. Lam, and V. Sundaresan, "Soot – a Java bytecode optimization framework," in *CASCON '99*. IBM Press, 1999, pp. 13–23.
[6] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps," in *PLDI 2014*, 2014, pp. 259–269.

A Review of Scalable Detection of Botnets Based on DGA

Mattia Zago, Manuel Gil Pérez and Gregorio Martínez Pérez

Department of Communications and Information Engineering, University of Murcia, 30100 Murcia, Spain
Email: *mattia.zago@um.es*, *mgilperez@um.es*, *gregorio@um.es*

***Abstract*—This conference article is a review of a work previously published by the authors and contains a summary of the identified challenges and proposed solution. The research navigates the state-of-the-art concerning the Machine Learning (ML) approaches to the detection of botnets based on Domain Generation Algorithms (DGAs), with a critical review of the existing frameworks in terms of algorithms and features used. The research aims to polish the data exploration process with a comparative analysis of those ML features presented in the state-of-the-art. The main results are several, starting with the creation of a common ground that enables future researches to focus on the study and design of new detection solutions instead of centre on the feature discovery process; following with the experimental demonstration that detection frameworks can be effective without harming user privacy; and, finally, by presenting multiple research challenges that can be exploited by future researches.**

***Index Terms*—Botnet, Domain Generation Algorithm, DGA, Machine Learning, Natural Language Processing**

***Tipo de contribución*—Investigación ya publicada**

I. Introduction

The habitual disregard of cybersecurity principles in computer networks is, up to this time, representing a serious and major threat. In fact, malwares are infecting hosts and on-demand services, often assembling coordinated armies that are commonly defined as botnets, and, when belonging to a botnet, a compromised host is called *zombie* or *bot*. In order to tackle this cyber threat, both the industry and the scientific community have studied and developed a number of Intrusion Detection Systems (IDSs) that attempts to identify and prevent malware infections. Because of this combined effort, malwares are implementing evasive techniques that aim to disguise their presence on the compromised bots and their communications to the Command & Control (C&C) server(s) [1]. The first and most important category of concealing techniques is represented by the Domain Generation Algorithms (DGAs), i.e., pseudo-random domain names generators that can be used to establish unpredictable *rendezvous* points with the C&C servers.

This article presents and summarises the findings described in [2]; in there, an analysis of the security challenges related to the research area of DGA-based botnet detection was presented. That research highlights how Machine Learning (ML) techniques are may represent a useful resource to tackle the threat represented by DGA-based botnets. Specifically, two ML problems are identified and analysed, i.e., given a set of Fully Qualified Domain Name (FQDN), binary separate legitimate domains from malware ones and categorise them according to their malware family. To do so, two complementary categories of metrics are introduced and formalised, namely the *Context-Free* and the *Context-Aware* features families; the former is based on Natural Language Processing (NLP) analysis of the domain name while the latter is based on DNS queries inspection.

In brief, the article reported how critical is the comparability inadequacy of the different models proposed in the literature, highlighting the general weakness in terms of data sources, feature characterisation and model optimisations.

II. Literature Review

In [2], a comparative analysis of the state-of-the-art solutions was proposed. Literature items are divided and studied according to their ML approaches, either supervised, unsupervised or semisupervised. To further characterise them, we defined a few taxonomy entries according to their approaches in terms of features usage. Each model can be classified depending on the family of the features used:

Family 1 (Context-Free Feature)**.** *A feature only related to an FQDN and thus independent of contextual information, including, but not limited to, timing, origin or any other environment configuration. An example of this family is the lexical analysis of the domain name.*

Family 2 (Context-Aware Feature)**.** *A feature that is dependent on the specific malware sample execution, realised in a precise environment with a specific configuration and in a particular time frame; for example, features extracted upon DNS-response inspection.*

In summary, the Context-Free feature family represents the complement set of the Context-Aware feature, that is, a feature can either belongs to the Context-Aware or the Context-Free family, but not both. For example, NLP features like the length of the domain name or the ratio between literal and numeric characters belong to the Context-Free family, while other metrics such as the communications patterns or the packet size pertain to the Context-Aware one.

Datasets related to the Context-Free family are generically composed only by raw lists of Algorithmically Generated Domains (AGDs) [3] and eventually legitimate FQDNs [4]. On the other hand, bigger repositories of network traces like [5] are crafted, heavily unbalanced and often include only short burst of malware packets. Nonetheless, the quality of these datasets is notable, and, when correctly used, they can be of great help to any detection model. On the contrary, datasets with Context-Free features consist mainly of lists of FQDNs belonging to a specific malware family. Concerning the Context-Aware family, datasets collecting the required information are subject to a number of limitations, so they are often rare, outdated and generally partial. This is because metrics based on network analysis are difficult to obtain and use due to privacy concerns, making it less feasible to find datasets related to this family of features.

Finally, in literature, there are traces of models that do not use manually engineered feature set, especially when considering Deep Learning (DL)-oriented solutions. Those models belong to a separate category, which has been defined as “Featureless”:

Family 3 (Featureless Model)**.** *A ML model that does not require features to learn the training dataset.*

With regard to the aforementioned aspects, a number of literature researches and solutions have been sorted and analysed, providing comparative tables that report (i) the type of classifier or cluster used; (ii) an indication whenever the authors made a comparison with other works or other methods; (iii) if their proposed framework is capable of real-time detection; (iv) the algorithm proposed; (v) the usage of either Context-Aware or Context-Free features; and (vi) a generic field that considers the overall results.

Remarkably, we noticed that only a few authors have cited any challenge related to 0-day, either as part of their analysis or as potential future work [2].

III. Feature Analysis and Experiments

The literature review resulted in a collection of 40 features related to the analysis of the FQDN with NLP techniques. This feature set is the result of the homogenization process carried out with respect to the existing solutions, having each feature studied, implemented and charted. All the details regarding the list of features, as well as their characteristics, is available in the full version of the reviewed article [2]. For example, the graphical representation of the feature `NLP-L-2LD`, shown in Fig. 1, can be used to depict its relevance in distinguishing malware variants according to their length.

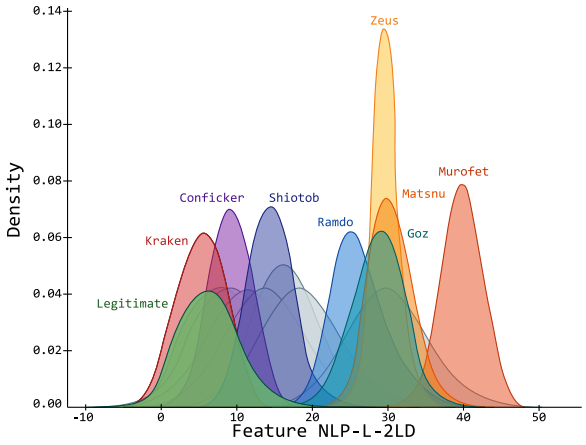


Fig. 1. Distribution histogram of NLP-L-2LD with 8 families highlighted among the 16 analysed.

The first part of the study is committed to describe and analyse these features; specifically, three use cases are built around the transformation applied: (i) the original feature set; (ii) a set of selected features; and (iii) a set of features extracted after applying a Principal Component Analysis (PCA).

To be as representative as possible, we gathered data regarding 16 malware families plus a collection of legitimated domain names and defined two distinct but complementary ML problems, specifically:

Experiment 1 (Binary). The experiment is designed to answer the ML question of separating legitimate FQDNs from malicious AGDs, considering all malware families as a single category.

Experiment 2 (Multiclass). The experiment is designed to classify the malwares samples according to their families.

These two classification tasks have been applied over the three aforementioned use cases and thus solved individually by six amongst the most represented ML-based classifiers in the literature. The experiment and use cases design, assumption and configurations are described in detail in [2]. Finally, Fig. 2 presents the Multiclass experiments' classifiers outcomes with respect to the use case that uses the original feature set. All the results, along with their discussion, are available in the full version of the article [2].

As expected, the evaluation of the classifiers performances over the different datasets pointed out how critical is the unavailability of common structures in models comparisons. The differences pointed out in the experiments and the use cases indicated that the data sources, the configurations and, in general, the conditions are pivotal.

IV. Future Works

The literature review, combined with the feature analysis and the experiments, defined a common ground knowledge regarding the ML approaches for detecting DGA-based botnets. It also proved that Context-Free features are more than capable of pinpointing this class of malwares without harming the user's privacy. As reported in the article, four research challenges should be considered by the cybersecurity community and industry:

1) privacy-oriented datasets must be researched and made publicly available;

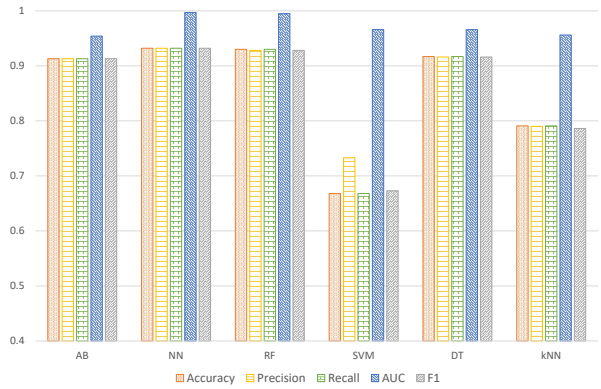


Fig. 2. Multiclass experiment using the original feature set.

2) it is mandatory to establish a series of shared best practices that lead and advise future researches related to ML applications for botnet detection;
3) to enable the research community to focus on the study of new approaches and detection algorithms instead of data gathering and preprocessing, ML-oriented and ready-to-use datasets must be researched and made publicly available; and,
4) having taken into consideration the complex nature of the data, *ad-hoc* nonlinear solutions might be worth investigating.

To this extent, potential research lines might include (i) the study of the Context-Aware feature family to establish whether it may combine and consolidate detection solutions; (ii) the development of a full-fledged, publicly available and labelled dataset of either Context-Aware and Context-Free features that might emerge as common-ground for the evaluation of existing and new ML solutions; (iii) the exploratory analysis of the above-mentioned data with nonlinear techniques to achieve improved classification results; and finally, (iv) testing detection algorithms against 0-day resilience by adding both new malware families and variants.

V. Conclusions

In summary, the conclusions of the reviewed article are experimentally demonstrating that privacy-invasive analyses are not necessary to detect DGA-based botnets. Moreover, as pointed out, the lack of common elements that define and guarantee the reproducibility of the results is still a major shortcoming. As a result, the research may, in the long run, to help establish common ground in terms of data, feature sets, procedures and eventually reactions that ideally enable future researches to focus only on the design and the study of new algorithms and advanced solutions for detection, reaction and mitigation purposes.

Acknowledgment

This study was founded by a predoctoral and a postdoctoral INCIBE grant within the "Ayudas para la Excelencia de los Equipos de Investigación Avanzada en Ciberseguridad" program, with codes INCIBEI-2015-27353 and INCIBEI-2015-27352, respectively.

References

[1] G. Vormayr, T. Zseby, and J. Fabini, "Botnet communication patterns," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2768–2796, 2017.

[2] M. Zago, M. Gil Pérez, and G. Martínez Pérez, "Scalable detection of botnets based on DGA: efficient feature discovery process in machine learning techniques," *Soft Computing*, 2019, doi: 10.1007/s00500-018-03703-8.

[3] A. Abakumov, "Github repository: andrewaeva/DGA," 2016, (Date last accessed 2019-02-01). [Online]. Available: https://github.com/andrewaeva/DGA

[4] Alexa Internet, Inc., "Alexa top sites," 2019, (Date last accessed 2019-02-01). [Online]. Available: https://www.alexa.com/topsites

[5] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Computers and Security*, vol. 45, pp. 100–123, 2014.

A Review of Improving the Security and QoE in Mobile Devices through an Intelligent and Adaptive Continuous Authentication System

José María Jorquera[1], Pedro Miguel Sánchez[1], Lorenzo Fernández[1], Alberto Huertas[2,*], Marcos Arjona[3] and Gregorio Martínez[1]

[1] *University of Murcia*, Murcia, Spain. Email: {josemaria.jorquera, pedromiguel.sanchez, lfmaimo, gregorio}@um.es
[2,*] *Waterford Institute of Technology*, Waterford, Ireland. Email: ahuertas@tssg.org
[3] *ElevenPaths, Telefónica Digital España*, Málaga, Spain. Email: marcos.arjona@11paths.com

***Abstract*—Continuous authentication systems for mobile devices focus on identifying users according to their behaviour patterns when they interact with mobile devices. Despite the benefits of these systems, they also have open challenges such as the authentication accuracy and the adaptability to new users' behaviours. With the goal of improving these challenges, the main contribution of this paper is an intelligent and adaptive continuous authentication system for mobile devices. The proposed system enables the real-time users' authentication by considering statistical information from applications, sensors and Machine Learning techniques based on anomaly detection. Several experiments demonstrated the accuracy, adaptability, resilience, and resources consumption of our solution.**

***Index Terms*—continuous authentication, adaptability, mobile devices, machine learning, anomaly detection**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

Continuous authentication systems for mobile devices aim to identify the owner of the device permanent and periodically but not only at a given moment, as traditional systems do. The fact of having the user permanently authenticated, and not from time to time, contributes to providing a higher level of security and confidence compared to traditional methods. The life-cycle of continuous authentication systems starts by modelling the users' behaviour when they interact with their mobile device for a given period of time. Once the data is acquired, it is pre-processed and stored in a dataset that contains relevant information about the users' behaviour patterns. Once the profile has been generated, the last step consists in the comparison of the current mobile usage with the dataset. This comparison is performed in real time and usually through Machine Learning (ML) techniques. Despite the benefits provided by current continuous authentication systems, there are several open challenges. Among them, we highlight the selection of dimensions and features allowing for modelling the user's behaviour in a precise and effective fashion. The combination of several dimensions and features of mobile devices is one of the critical aspects to obtain a great accuracy during the authentication process. Another challenge is the adaptability of the authentication systems to changes in the user's behaviour. The decision of how and when the user's profile should be updated with new behaviours is critical to reach the desired adaptability. Additionally, forgetting old behaviours is also an important aspect of an adaptable authentication system.

In this context, this paper is a review of [1], whose main contribution is the design and implementation of an intelligent and adaptive system of continuous authentication for mobile devices. The proposed solution relies on modelling and creating users' profiles that contain data and features related to the usage of the applications and sensors of the device. ML-based techniques based on anomaly detection are considered by our solution to measure the level of similarity between the current usage of the device and the well-known usage. Different experiments provided promising results in terms of accuracy, adaptability, resilience and resource consumption of the proposed solution.

II. Proposed Continuous Authentication System

This section describes the four phases making up our intelligent and adaptive continuous authentication system. A diagram with the different steps composing the design process is depicted in Fig. 1.

1) **Phase 0 (red colour): Feature engineering.** This is a preliminary and non interactive stage where we make a first selection of dimensions (sensors and applications usage) and features. This initial set is subsequently refined by using feature selection techniques.
2) **Phase 1 (blue colour): Acquisition of behavioural data and dataset generation.** This phase consists of acquiring data from the mobile device and extracting the relevant features selected in phase 0. By doing it, we are able to capture the user's behaviour and create a dataset. This dataset will be updated in real time with the new user's behaviours.
3) **Phase 2 (orange colour): Computation of the authentication level.** During this stage, a ML algorithm is trained to fit a model from the user's behaviour contained in the dataset. Periodically, the new user's behaviour is sampled and, then, evaluated by the fitted model which returns an authentication level score.
4) **Phase 3 (green colour): Automatic adaptability to new behaviours.** Driven by a use case, it focuses on enabling the system adaptability through the elimination/inclusion of old/new behaviours to the dataset.

After the design phase, we implemented a mobile application for the Android operating system that considers three phases (from 1 to 3) of our system. With the goal of carrying out the acquisition of relevant data from sensors and

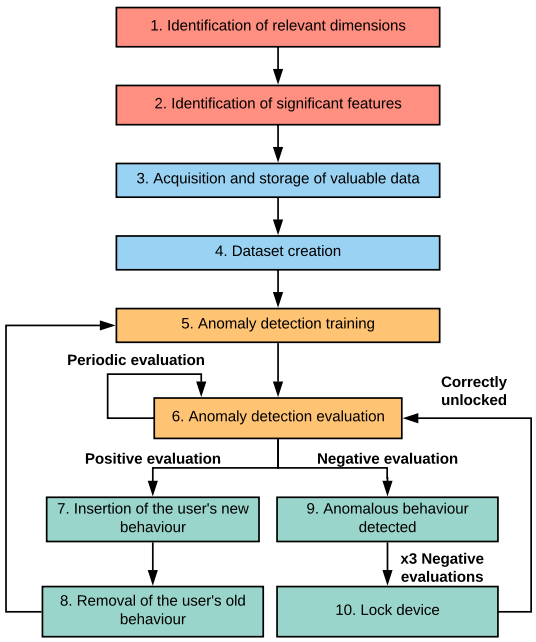


Fig. 1. Phases and processes of the proposed system.

applications (Phase 1), we used and implemented different classes and methods of Android libraries (more information in [1]). After obtaining periodically the features (each 20 seconds for sensors and 60 for application statistics) for 15 days, the dataset is created and Phase 2 starts. In this stage, we use the implementation of Isolation Forest (IF) provided by the Weka library. Specifically, we generate a ML model by training IF with the datasets and, in real time, we evaluate the new feature vectors (sensors and apps) every 60 seconds. Once evaluations for sensors and applications are performed, the two scores are normalized and combined to get a unique and final score. This score will represent the authentication level of the user interacting with the device. Finally, Phase 3 follows a use case specification and checks if the authentication level (AL) is higher than a given threshold, which is defined in advance. If so, the user is positively authenticated by the proposed system and the datasets are updated by including new vectors and removing the old ones. In contrast, if the AL score is lower than the threshold, the user is not authenticated and the datasets are not updated. Furthermore, if the AL is not over the threshold after three consecutive evaluations, the device is locked.

III. Experiments

A pool of experiments measured the accuracy of the system, its adaptability to behavioural changes, its resiliency against attacks, and its resource consumption in terms of energy, storage, and processing time. Due to the room restriction, we focus this section on the adversarial attack experiment (other experiments are explained in [1]). For this experiment our system was trained during two weeks with the owner's behaviour. After that, we prepared the following two adversarial attacks:

1) Trial and error. The device was used for 10 min by five attackers who were not aware of the owner's behaviour.

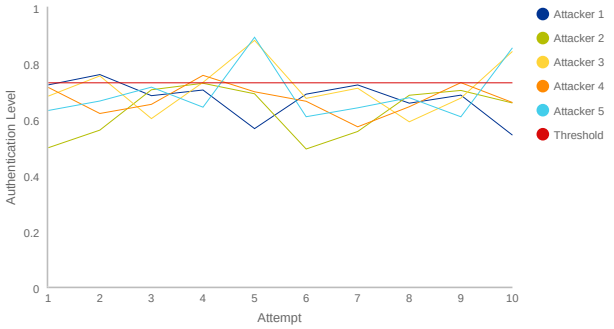


Fig. 2. Shoulder surfing attack scores.

2) Shoulder surfing. The device was used for 10 min by five attackers who had seen the owner's behaviour in terms of position and applications usage for five minutes.

Fig. 2 shows the results of the shoulder surfing attack, being the red line the threshold considered by our solution to classify the attacker as device owner. As can be seen, Fig. 2 demonstrates the resiliency of the proposed solution to this attack. The main reason why our solution is robust to both adversarial attacks is the correct selection of features. On the one hand, regarding the application dimension, there are some features that cannot be learned by attackers when they look at the owner's behaviour. Among these features, we highlight the number of apps opened during the last minute, or the application opening order. On the other hand, regarding the sensor dimensions, although attackers can learn the position in which the owner uses the device, they cannot duplicate the vibrations, inclination and orientation of owner's device in specific moments. In this sense, the inclination and orientation alter the mean values, and the vibrations affect the maximum, minimum and variance of the accelerometer and gyroscope.

IV. Conclusions and Future Work

We have designed, implemented and validated an intelligent and adaptive continuous authentication system for mobile devices that models the users' behaviours by considering data coming from both applications' usage statistics and sensors. The proposed solution is able to adapt itself to changes in the user's behaviours and uses anomaly detection based on semi-supervised ML techniques to perform the authentication process. A pool of experiments show promising results in terms of accuracy, adaptability, resilience, and consumption. As future work, we plan to launch our solution as a final product that meets the real requirements of the current market.

Acknowledgements

This work has been supported by the Irish Research Council, under the government of Ireland post-doc fellowship (grant GOIPD/2018/466)

References

[1] J. M. Jorquera Valero, P. M. Sánchez Snchez, L. Fernández Maimó, A. Huertas Celdrán, M. Arjona Fernández, S. De Los Santos Vílchez, and G. Martínez Pérez: "Improving the Security and QoE in Mobile Devices through an Intelligent and Adaptive Continuous Authentication System," *Sensors, vol. 18, no. 11, pp. 3769, November 2018. DOI 10.3390/s18113769*

Técnica de Autenticación de Imágenes Digitales Basada en la Extracción de Características

Esteban Alejandro Armas Vega, Carlos Quinto Huamán,
Ana Lucila Sandoval Orozco, Luis Javier García Villalba
Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: {esarmas, cquinto}@ucm.es, {asandoval, javiergv}@fdi.ucm.es

***Resumen*—En los últimos años, ha habido un gran crecimiento en el uso de imágenes digitales en la sociedad moderna. Esto, junto con la facilidad del uso de aplicaciones de edición de imágenes, compromete la autenticidad y veracidad de una imagen digital. Estas aplicaciones permiten manipular el contenido de la imagen sin dejar rastros visibles. Además de esto, la facilidad de distribuir información a través de Internet ha hecho que la sociedad acepte todo lo que ve como verdadero sin cuestionar su integridad. Este artículo propone una técnica de autenticación de imágenes digitales que combina el análisis de los patrones de textura locales con la transformada discreta Wavelet y la transformada discreta de coseno para extraer características de cada uno de los bloques de una imagen. Posteriormente, utiliza un SVM para crear un modelo que permita verificar la autenticidad de la imagen. Los experimentos se realizaron con imágenes falsificadas de bases de datos públicas y los resultados obtenidos demuestran la eficacia del método propuesto.**

***Index Terms*—Análisis Forense, Detección de manipulaciones, Empalme, Imágenes Digitales, Patrón Binario Local, Retoque, SVM, Transformada Discreta del Coseno, Transformada Wavelet.**

I. Introducción

En los últimos años el uso de dispositivos móviles ha aumentado considerablemente llegando a ser una herramienta que forma parte de la vida cotidiana de la sociedad actual. En 2017, un informe de Cisco Systems [1] explica que el tráfico de datos móviles se ha multiplicado por 18 en los últimos 5 años y se espera que este tráfico continúe aumentando. Esta información fue confirmada en 2018 por Ericsson en su informe [2] en el que estima que para el año 2023 el tráfico de datos móviles se multiplicará por 7 y casi tres cuartos del tráfico de datos móviles del mundo se utilizará para transferencia de ficheros multimedia y redes sociales. Como consecuencia, se ha facilitado el proceso de compartir datos de forma masiva y rápida. Las imágenes y vídeos digitales son, gracias a las redes sociales y a las aplicaciones de mensajería instantánea, hoy en día uno de los recursos que más tráfico de datos genera.

Por otro lado, la mejora continua de las cámaras incorporadas en los dispositivos móviles junto a la evolución de las herramientas de edición de imágenes han hecho que cada vez sea más sencillo manipular una imagen con excelentes resultados. Para enfrentar este tráfico masivo de imágenes manipuladas el área de análisis forense investiga nuevas técnicas de detección de manipulaciones, para evaluar la integridad de una imagen.

Las imágenes manipuladas llevan existiendo desde hace décadas y están presentes en muchos sectores (política, cine, prensa, rama judicial, etc.). Una de las primeras imágenes manipuladas de la historia [3], es la del fotógrafo Hippolyte Bayard, quien creo una imagen falsa de él cometiendo suicidio. Más tarde se descubrió que la fotografía fue hecha debido a la frustración del autor por haber perdido la oportunidad de convertirse en "el inventor" de la fotografía, en lugar de Louis Daguerre quien fue el que patento el proceso fotográfico antes que Bayard. Antes de las computadoras, se realizaron manipulaciones fotográficas pero es en estos días en que gracias al desarrollo del software, se ha masificado y facilitado el proceso. Por lo tanto, detectar imágenes digitales manipuladas es de gran importancia en muchas áreas y con diferentes objetivos. Una de las áreas en donde la verificación de la legitimidad de una imagen es fundamental es en lo judicial, donde las imágenes o vídeos pueden suponer evidencia de gran valor para la resolución de la demanda. Un ejemplo de esto fue el arresto de un conductor [4] que conducía su automóvil a más de 200 Km/h y la evidencia utilizada por la fiscalía fue el vídeo grabado por un peatón, a través del cual se demostró que el imputado circulaba a dicha velocidad.

Sin embargo, para que una imagen pueda ser usada como prueba válida o evidencia en un juicio, se debe asegurar su integridad y demostrar que no ha sido objeto de manipulación. Para llevar a cabo este tipo de autenticación es necesario hacer uso de técnicas robustas de identificación de manipulaciones que puedan garantizar con gran fiabilidad que la imagen es original. Por lo tanto, la necesidad extrema de encontrar métodos para validar la autenticidad y la integridad de la imagen se volvió vital.

Por todo lo anterior, se deben estudiar y proponer técnicas forenses que permitan hacer frente al gran número de imágenes manipuladas que existen hoy en día.

El resto del trabajo está organizado como sigue: La Sección II detalla características de las manipulaciones comúnmente utilizadas. En la Sección III se describe las principales técnicas de detección de imágenes manipuladas, haciendo énfasis en las técnicas con enfoque pasivo más relevantes de la literatura. Los detalles de la técnica de detección propuesta en este trabajo se presentan en la Sección IV. En la Sección V se analizan los resultados de los experimentos realizados y, finalmente, las conclusiones del trabajo se recogen en la Sección VI.

II. Manipulación de Imágenes

Entre los tipos de manipulación de imágenes, destacan los siguientes: retoque, copiar y pegar, empalme de imágenes y falsificación de huellas digitales [5].

II-A. Retoque

El retoque de imágenes consiste en aplicar diferentes filtros sobre la imagen original para mejorarla según unos objetivos manteniendo siempre unas características similares. Para ello se copian y pegan regiones de la imagen de la misma área. Los retoques que se realizan suelen estar enfocados a perfeccionar la escena [6]. El acabado de las imágenes, el acabado varía dependiendo del contenido de la imagen y de los fines con los que se realiza la manipulación. Esta técnica de manipulación es muy común en los sectores de la publicidad, cine y comunicación [7].

Las portadas y los anuncios de las revistas de moda generalmente utilizan algún tipo de retoque para ocultar las imperfecciones y así aumentar los niveles de belleza en las fotografías. La Figura 1 muestra un ejemplo de retoque fotográfico en el que la apariencia de la modelo se modificó digitalmente. La Figura 1(b) muestra la imagen original sin retoque y la Figura 1(a) muestra el resultado de retocar la imagen para la portada de la revista *Nitro*.

(a) *Imagen Manipulada* (b) *Imagen Original*

Figura 1. Portada manipulada de la revista Nitro[8]

II-B. Copia – Pega

La técnica de copia-pega consiste en copiar una región y pegarla encima de otra región de la misma imagen para ocultar partes de la imagen o duplicar regiones. También pueden llevarse a cabo técnicas de post-procesamiento, como escalar, rotar o aplicar alguna clase de filtro. Estas técnicas hacen más costoso el proceso de detección de la manipulación. Se usa habitualmente para ocultar información relevante de una o más áreas de la imagen [9][10].

Un ejemplo de esta técnica se muestra en la Figura 2(a). La primera imagen 2(a) es una foto histórica que tuvo lugar en Irán en 2008 y representa el lanzamiento exitoso de cuatro misiles, tal como fue publicado por la agencia de noticias de Irán (Sepah News). La foto original que se publicó posteriormente muestra que solo hubo tres misiles lanzados 2(b). La imagen modificada también muestra la aplicación de técnicas de pos-procesamiento en el humo expulsado por el misil para ocultar la manipulación.

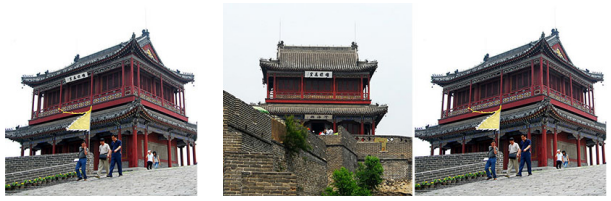
(a) *Imagen Manipulada* (b) *Imagen Original*

Figura 2. Lanzamiento de misil iraní [11]

II-C. Empalme Fotográfico

La técnica de empalme consiste en copiar la región de una determinada imagen y pegarla en otra distinta. Es muy usada en fotomontajes donde se combinan dos imágenes dando la sensación de ser una sola. Detectar el área exacta que se ha falsificado en la imagen, mediante la técnica de empalme, es de gran complejidad en comparación con las anteriores técnicas de manipulación. Esto se debe a que no es posible buscar áreas duplicadas ya que la región manipulada proviene de una imagen diferente [12].

La figura 3 muestra un ejemplo de la técnica de empalme y las dos imágenes originales utilizadas en este proceso. A partir de la primera imagen original Figura 3(b), el signo del templo se recortó y se colocó en la segunda imagen original Figura 3(b) para crear el resultado final Figura 3(a).



(a) *Imagen Manipulada* (b) *Imágenes Originales*

Figura 3. Empalme de imagenes [13]

II-D. Manipulación de la Huella Digital

La manipulación de la huella digital de una imagen se centra en la información que llevan a la identificación de la cámara que la generó [14]. Este tipo de técnica se subdivide en: Anonimización de la imagen, que consiste en eliminar la información del origen de la imagen y, la falsificación de la imagen, que elimina la huella de la cámara que generó la imagen y coloca una huella de otra cámara de un dispositivo diferente. Estas técnicas no implican la alteración de la imagen en sí, sino la modificación de la información asociada (huella digital) que proviene del sensor que capturó la imagen.

Existen varias fuentes de imperfecciones y ruido introducidas durante el proceso de adquisición de imágenes. Esto imperfecciones aparecen principalmente por dos razones; primero hay componentes aleatorios como el ruido de lectura o el ruido de disparo [**?**] [**?**] y segundo debido al ruido del patrón, que es un componente determinista del sensor y permanece aproximadamente igual si varias fotos de la misma escena están tomados. Este patrón es útil para detectar la fuente de origen de una imagen, ya que cada dispositivo tendrá un patrón de ruido específico [15] [14].

III. Trabajos Relacionados

Existen dos enfoques forenses de detección de imágenes manipuladas: Intrusivo o activo y no intrusivo o pasivo [16].

- **Enfoque Activo**: Analiza las marcas de agua o señales que deja un dispositivo al momento de generar una imagen digital. El mayor inconveniente de este tipo de enfoque es que muchas cámaras no tienen la capacidad de incorporar este tipo de marcas o firmas, por lo que su alcance es limitado.
- **Enfoque Pasivo**: Analiza el contenido y las características de la imagen digital. A su vez este enfoque puede clasificarse en: métodos basados en aprendizaje y métodos basados en bloques.

El enfoque pasivo tiene un alcance más amplio que el enfoque activo ya que no necesita información previa sobre las imágenes. A continuación se presentan las propuestas de enfoque pasivo más relevantes.

En [7] se propuso un algoritmo eficiente diseñado específicamente para predecir la presencia de retoques en imágenes de portadas de revistas. El conjunto de datos de 468 fotos (originales y retocadas) se valoraron entre 1 (muy similar) y 5 (muy diferente) dependiendo de la cantidad de alteración fotográfica. Se calcularon las modificaciones geométricas y fotométricas de cada foto original y retocada y, posteriormente, se extrajeron ocho estadísticas de resumen que incorporan el grado de retoque fotográfico para calcular la correlación con la valoración de cada foto. Se utilizó el algoritmo de Máquina de Soporte Vectorial SVM para determinar el grado de modificación de la imagen. La precisión máxima obtenida con los experimentos fue de 98,75 %.

El algoritmo propuesto en [17], utiliza una red neuronal para extraer características y SVM para clasificar las imágenes en una clase sin retoques o retocada. En los experimentos se utilizó el conjunto de datos "*ND-IIITD retouched faces*" de 325 de caras retocadas y se obtuvo un 87 % de acierto.

En [18] se propone la extracción de las características de color, forma y textura de tres regiones faciales predefinidas. Se utilizaron los conjuntos de datos YMU y MIW [19] para entrenar y predecir, respectivamente, un sistema SVM con núcleo RBF para clasificarlas. La precisión que se obtuvo fue de un 93 %. Posteriormente, en [20] se propuso un algoritmo más preciso para la detección de maquillaje en los mismos conjuntos de datos utilizando características de textura y forma. La técnica propuesta extrae un vector de características que captura las características de forma y textura de la cara usada como entrada del algoritmo. Se consiguió aumentar la precisión a un 98.5 % usando un clasificador SVM.

El método de detección de empalme propuesto en [21] modela los cambios de manipulación utilizando características estadísticas extraídas de matrices 2D generadas al aplicar la transformada discreta de coseno de bloques de varios tamaños (MBDCT). En los experimentos se obtuvo un 91.40 % de acierto sobre el conjunto de datos "*Columbia*" usando SVM.

En [22] se representan los cambios de alteración utilizando características extraídas de matrices 2D generadas al aplicar MBDCT y métricas de calidad de imagen (IQM) y utiliza SVM para la clasificación. Se usa el conjunto de datos "*Columbia*" y obtiene una precisión del 87.10 %.

En [23] se utiliza LBP para extraer características de matrices 2D generadas por MBDCT y PCA para reducción de dimensionalidad y SVM para clasificación. Se obtiene una precisión de 89.93 % usando el conjunto de datos "*Columbia*".

En [24] se modelan los cambios de la manipulación utilizando la distribución estacionaria del borde de imagen extraída del componente cromático utilizando una cadena de Markov de estado finito y SVM como clasificador. Se alcanza una precisión de 95.6 % con el conjunto de datos "*CASIA TIDE v2.0*".

En [25] exploraron el efecto de diferentes modelos de color en la detección de falsificación de empalme. En este trabajo, se hace una comparación de los modelos cromáticos frente a los modelos RGB y de luminancia utilizados comúnmente. Se emplean cuatro vectores RLRN con diferentes direcciones extraídas de canales de crominancia correlacionados como características para la detección de empalme en imágenes. Finalmente, se usa SVM como algoritmo clasificador. El conjunto de datos utilizado en los experimentos son "*CASIA TIDE v1.0*" y "*Columbia*" con una precisión de 94.7 %.

En [26] se presenta un trabajo para la detección de falsificaciones aplicada a imágenes de huellas digitales, el método que usa emplea DWT junto a LBP, consigue obtener una precisión del 92 %. Se utilizaron para los experimentos diferentes imágenes de huellas dactilares obtenidas por distintos tipos de escáneres. Todas estas imágenes se obtuvieron del conjunto de datos "*LivDet*" y se utilizó SVM para su clasificación.

En [27] se propone un método que combina el descriptor de textura LBP junto a DCT para detectar cambios producidos por las manipulaciones de empalme y también de copia-pega. En los experimentos se utiliza SVM obteniendo una tasa de acierto entre el 97.50 % y el 97.77 % sobre el conjunto de datos "*CASIA TIDE v2.0*".

En [28] se estudian investigaciones recientes en el campo y proponen la mezcla de dos técnicas (imperfecciones del sensor y transformadas wavelet) para obtener una mejor identificación de fuentes de imágenes generadas con dispositivos móviles. Los resultados muestran que las imperfecciones del sensor y las transformadas wavelet pueden servir conjuntamente como buenas características forenses para ayudar a rastrear la cámara fuente de las imágenes producidas por teléfonos móviles. Además, este modelo también permite determinar con gran precisión la marca y el modelo del dispositivo.

En [29] se preseleccionó un método para la identificación de la fuente de imágenes basado en la extracción de características de ruido de foto respuesta no uniforme (PRNU), se utilizó una máquina SVM para la clasificación. Este trabajo se utilizó únicamente en dispositivos móviles y se consiguió mostrar que este método conseguía buenos resultados cuando se tenía una gran cantidad de cámaras fuente para la clasificación.

Como se puede observar la mayoría de técnicas emplean los patrones de ruido para identificar y extraer el ruido del sensor. Para poder comprobar si se ha llevado a cabo una modificación o eliminación bastaría con comparar la huella digital de la imagen original y de la imagen manipulada.Sin embargo, nuestra propuesta combina el análisis de los patrones de textura locales junto con las características obtenidas de aplicar la transformada discreta de Wavelets junto con la del Coseno a la imagen.

IV. Método Propuesto

En esta sección se presenta un algoritmo que comprueba de forma precisa y eficaz la integridad de una imagen basada en la extracción de características que combina la transformada wavelet con el histograma aplicado a bloques LBP y DCT.

Cada imagen $M \times N$ se convierte al modelo de color $YCbCr$. Este modelo representa los colores en forma de componentes de luminancia Y y de crominancia Cb y Cr.

Como la visión humana percibe el componente de luminancia de una manera más clara que el componente de crominancia, se considera que, la mayoría de las trazas de manipulación, que no pueden detectarse a simple vista, pueden estar en el canal cromático. Este canal describe el contenido de la señal de la imagen, como los bordes. Cualquier inconsistencia en estos bordes causada por la operación de una alteración se enfatiza y, por tanto, se nota. El componente Y se descarta y se hace uso solamente de los dos componentes de crominancia (Cb y Cr).

Por cada componente Cb y Cr de la imagen se aplica DWT. La transformada discreta wavelet analiza una imagen en diferentes escalas y orientaciones. El proceso de empalme a menudo introduce una transición nítida en la matriz bidimensional que representa a la imagen en términos de bordes, líneas y esquinas que se caracterizan por componentes de alta frecuencia en el dominio de transformada de Fourier.

Cada componente Cb y Cr quedará descompuesto en cuatro sub-bandas: el coeficiente de aproximación LL con la información de baja frecuencia y tres coeficientes de dirección (HL, LH y HH) con la información de alta frecuencia en las diferentes direcciones (coeficientes horizontal, vertical y diagonal, respectivamente). Se descartan las sub-bandas HL, LH y HH ya que el componente LL concentra la mayor parte de la energía.

Cada componente de baja frecuencia de la crominancia es dividido en bloques superpuestos $B \times B$ con una ventana deslizante de un pixel. Se obtiene un total de $(M - B + 1) \times (N - B + 1)$ características.

Debido a la capacidad de LBP para capturar las diferencias de textura, es una herramienta muy eficiente para la detección de las falsificaciones en las imágenes digitales. A cada bloque resultante se le extrae el LBP utilizando la Fórmula 1.

$$LBP_{p,r} \sum_{i=1}^{p-1} s(p_i - p_c)2^i \tag{1}$$

donde, p es el número de píxeles vecinos, r es el radio de vecindad y p_c es el valor del píxel central. Si el valor del píxel del vecino es menor que el del centro, se le asigna el dígito binario 0, de lo contrario un 1. Después, los dígitos binarios del vecino se juntan para construir un código binario.

Debido a la capacidad de LBP para capturar las diferencias de textura, es una herramienta muy eficiente para la detección de las falsificaciones en las imágenes digitales. Se extrae por cada bloque el patrón binario local utilizando como valor de vecindad 8 para obtener una mayor precisión. Para reducir el número de características obtenidas con LBP se calcula el histograma, quedando así cada bloque representado por 256 códigos LBP. En la Figura 4 se observa una imagen manipulada y su resultado al aplicar LBP. La imagen ha sido obtenida del dataset CASIA v1.0.

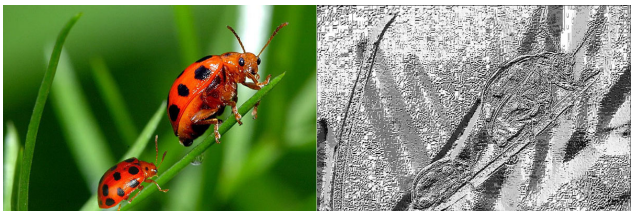


Figura 4. Imagen manipulada y su transformación al aplicar LBP.

De esta manera se observan los cambios de las diferentes texturas con una mayor eficiencia. En este caso las características de textura locales de toda la imagen se pueden describir mediante un histograma normalizado que está formado por 2^P códigos LBP, siendo P el número de vecinos que rodean al píxel central.

Una vez obtenidos los histogramas de todos los bloques de cada componente se aplica la transformada discreta del coseno a cada uno de ellos utilizando la Fórmula 2.

$$f_j = \sum_{k=0}^{n-1} x_k \cos\left[\frac{\pi}{n} j \left(k + \frac{1}{2}\right)\right] \tag{2}$$

La Transformada de Coseno Discreta DCT tiene una gran capacidad de compactar la energía o la mayor parte de información en un número reducido de coeficientes y por ser independiente del número de datos de entrada que recibe garantizando una mayor eficiencia al trabajar con imágenes de grandes dimensiones.

Con esto se representa cada bloque por un conjunto de coeficientes DCT. El resultado de aplicar la transformada discreta del coseno es la obtención de una secuencia finita de puntos como resultado de la suma de varias señales con distintas frecuencias y amplitudes. De cada conjunto de coeficientes se obtiene la desviación estándar como característica.

Finalmente, cada componente de crominancia de la subbanda LL quedará representado como un vector de desviaciones estándar de los conjuntos de coeficientes DCT de todos sus bloques. El vector final de características que se enviará al clasificador SVM será la concatenación de los vectores de ambas crominancias. Para encontrar los mejores pares de parámetros óptimos de clasificación C y γ se utiliza la herramienta *Grid-search* y *Cross-validation* de la implementación LIBSVM [30].

Algorithm 1: Extracción de vector de características

Input: $[Images]$: Imágenes en color
Result: $Vector_{caracteristicas}$: Vector de características extraídas

1. **foreach** $Imagen \in [Imagenes]$ **do**
2. Convertir $Imagen\ RGB$ a $YCbCr$;
3. A cada C_b y C_r aplicar la Transformada Discreta Wavelet DWT;
4. Extraer bloques superpuestos de NxN de cada componente LL_{Cb} y LL_{Cr} resultante de DWT;
5. Extraer las caracteristicas $LBP_{p,r} = \sum_{i=1}^{p-1} s(p_i - p_c)2^i$ de cada bloque NxN;
6. Obtener $Histograma$ de cada bloque NxN;
7. Extraer los coeficientes $DCT = \sum_{k=0}^{n-1} x_k \cos\left[\frac{\pi}{n} j \left(k + \frac{1}{2}\right)\right]$ de cada bloque NxN;
8. Concatenar los coeficientes resultantes de cada bloque para formar un vector de características;
9. **return** $Vector_{Caracteristicas}$;

SVM es una técnica de aprendizaje automático supervisado que se utiliza para resolver problemas de reconocimiento de patrones y análisis de regresión. SVM permite modelos complejos que no están definidos simplemente por los hiperplanos en el espacio de entrada. Para lograr esto, los datos se asignan a un espacio de características de dimensión superior. Una vez, en el espacio de características de dimensión superior, el algoritmo SVM divide los datos aplicando un hiperplano lineal.

SVM ha demostrado ser un método de predicción preciso y confiable [31], [32], [33]. El algoritmo propuesto de este documento, se utiliza una SVM *kernelizada* para clasificar las imágenes entre auténticas y manipuladas (clasificación bi-clase).

V. Experimentos y Resultados

A lo largo de esta sección se mostrarán todos los experimentos que se han realizado para evaluar la efectividad de los algoritmos de identificación de manipulaciones basado en entrenamiento.

El primer conjunto de experimentos se basó en comprobar la variación de la precisión al aplicar el Filtro de Espejo en Cuadratura (QMF) sobre el algoritmo de identificación de manipulaciones basado en DWT. Se realizó una prueba aplicando LBP sobre los coeficientes wavelets de manera directa y otra pasando los coeficientes wavelets por QMF antes de aplicar LBP. Para ello se usaron los conjuntos de datos CASIA v1.0 y CASIA v2.0. En la Tabla I pueden verse los resultados obtenidos.

Tabla I
Variación de las precisiones al aplicar QMF

Conjunto de datos	DWT-LBP-HIST	DWT-QMF-LBP-HIST
CASIA v1.0	97.66 %	98.01 %
CASIA v2.0	98.73 %	99.43 %

Como se puede observar en la tabla, el algoritmo que hace uso de QMF consigue un leve aumento de la precisión en ambos conjuntos de datos.

El segundo grupo de experimentos se buscó cuál de los dos algoritmos de detección de manipulaciones propuestos presentaba mayor precisión y eficiencia. En la Tabla II se muestran los resultados obtenidos.

Tabla II
Precisión obtenida por ambos algoritmos

Conjunto de datos	DWT-QMF-LBP-HIST	LBP–HIST-DCT
CASIA v1.0	96,57 %	53,72 %
CASIA v2.0	99,43 %	94,94 %
IFS-TC	97,56 %	70,22 %

Como se puede observar en la tabla el algoritmo basado en DWT obtiene mejores resultados en los tres conjuntos de datos y es más rápido que el algoritmo basado en DCT. Esto se debe a la división en bloques superpuesta que realiza. Como referencia se midió el tiempo de ejecución de ambos algoritmos para una imagen de dimensiones 1280x854. El algoritmo basado en DCT presentaba un tiempo de ejecución de 89 segundos, siendo más eficiente el algoritmo basado en DWT con un tiempo de ejecución de 16 segundos. Asimismo, se observó que a mayor tamaño de imagen el algoritmo basado en DCT incrementa considerablemente el tiempo de procesado de las imágenes. En cambio en el algoritmo basado en DWT se produce un incremento leve del tiempo de ejecución.

Finalmente, ambos algoritmos consiguen buenos resultados con imágenes comprimidas, como por ejemplo imágenes con formato JPEG. También se obtuvieron buenos resultados para imágenes con formato PNG. Los dos algoritmos no consiguen llegar a una precisión superior al 50 % en imágenes con formato TIFF.

VI. Conclusiones

Las imágenes digitales contienen una gran cantidad de información relevante. Debido a esto, son un elemento muy importante en el ámbito legal y se han convertido en evidencias que aportan gran valor en la resolución de un juicio. Para que estas evidencias lleguen a ser válidas se debe poder garantizar su autenticidad e integridad de forma fiable. Existen numerosas aplicaciones que consiguen editar imágenes con resultados altamente profesionales y detectar si una imagen ha sido modificada mediante alguna técnica de manipulación es una tarea complicada. Para poder garantizar la integridad de una imagen es de mucho interés tener herramientas forenses que puedan detectar estas falsificaciones.

En este trabajo se ha realizado un estudio exhaustivo sobre las técnicas existentes de detección de manipulaciones haciendo énfasis en las técnicas de detección de empalme y copia-pega. Se ha diseñado una técnica para la detección de manipulaciones que mejoran los resultados obtenidos por estas investigaciones. La técnica diseñada está basada en DWT junto a histogramas LBP aplicando QMF para la detección de manipulaciones: Extrae las características wavelets y obtiene el histograma aplicando LBP de la imagen investigada. Finalmente, mejora la precisión con QMF y clasifica la imagen.

Para evaluar la técnica diseñada en este trabajo, se han realizado un conjunto de experimentos con los mismos conjuntos de datos utilizados en el estado del arte para poder comparar los resultados. Los resultados de los experimentos son los siguientes: Primero, el algoritmo consiguió una precisión máxima del 99,43 % en el conjunto de imágenes usado *CASIA v2.0*. A su vez demostró ser eficiente en las pruebas realizadas tras comparar su tiempo de ejecución con los demás algoritmos. Esto se debe a que no realiza una división de la imagen en bloques y trabaja directamente con la imagen original.

Agradecimientos

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326.

RAMSES

Referencias

[1] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, February 2017.

[2] ERICSSON, "Ericsson Mobility Report," ERICSSON, Tech. Rep., 06 2018. [Online]. Available: \url{https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf}

[3] M. Sapir, "The impossible photograph: Hippolyte bayard's self-portrait as a drowned man," *MFS Modern Fiction Studies*, vol. 40, no. 3, pp. 619–629, 1994.

[4] E. Mundo, "Detenido por Circular a 200 Kilómetros por Hora tras Subir un Vídeo a Redes Sociales," http://www.elmundo.es/madrid/2017/08/30/59a68f0a468aeb7a658b4607.html, August 2017.

[5] M. A. Qureshi and M. Deriche, "A Bibliography of Pixel-Based Blind Image Forgery Detection Techniques," *Signal Processing: Image Communication*, vol. 39, pp. 46–74, 2015.

[6] I. T. Young, J. J. Gerbrands, and L. J. Van Vliet, *Fundamentals of image processing*. Delft University of Technology Delft, 1998.

[7] E. Kee and H. Farid, "A Perceptual Metric for Photo Retouching," *National Academy of Sciences*, vol. 108, no. 50, pp. 19 907–19 912, November 2011.

[8] V. Sun, "Photos: 20 More Stars and Celebrities Before and After Photoshop," http://www.vancouversun.com/life/fashion-beauty/Photos+more+stars+celebrities+before+after+Photoshop/7841314/story.html, July 2014.

[9] M. Boutell and J. Luo, "Beyond pixels: Exploiting camera metadata for photo classification," *Pattern Recognition*, vol. 38, no. 6, pp. 935–946, 2005.

[10] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using sift algorithm," in *Proceedings of the IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, Wuhan, China, December 2008, pp. 272–276.

[11] I. Fourandsix Technologies, "Photo Tampering Throughout History," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, December 2017.

[12] X. Zhao, S. Wang, S. Li, J. Li, and Q. Yuan, "Image splicing detection based on noncausal markov model," in *Proceedings of the IEEE International Conference on Image Processing*, Melbourne, VIC, Australia, September 2013, pp. 4462–4466.

[13] J. Dong and W. Wang, "CASIA TIDE v1.0 - v2.0," http://forensics.idealtest.org/.

[14] L. J. García Villalba, A. L. Sandoval Orozco, J. Rosales Corripio, and J. Hernández Castro, "A PRNU-based Counter-forensic Method to Manipulate Smartphone Image Source Identification Techniques," *Future Generation Computer Systems*, vol. 76, pp. 418–427, November 2017.

[15] N. Khanna, A. K. Mikkilineni, G. Chiu, J. P. Allebach, and E. Delp, "Forensic Classification of Imaging Sensor Types," in *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 6505, no. 65050U, February 2007.

[16] B. Mahdian and S. Saic, "A Bibliography on Blind Methods for Identifying Image Forgery," *Signal Processing: Image Communication*, vol. 25, no. 6, pp. 389–399, July 2010.

[17] A. Bharati, R. Singh, M. Vatsa, and K. W. Bowyer, "Detecting Facial Retouching Using Supervised Deep Learning," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 1903–1913, September 2016.

[18] C. Chen, A. Dantcheva, and A. Ross, "Automatic Facial Makeup Detection with Application in Face Recognition," in *Proceedings of the International Conference on Biometrics (ICB)*, Madrid, Spain, June 2013, pp. 1–8.

[19] A. Dantcheva, C. Chen, and A. Ross, "Can Facial Cosmetics Affect the Matching Accuracy of Face Recognition Systems?" in *Proceedings of the IEEE 5th International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Washington DC, USA: IEEE, September 2012, pp. 391–398.

[20] N. Kose, L. Apvrille, and J. L. Dugelay, "Facial Makeup Detection Technique Based on Texture and Shape Analysis," in *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, Ljubljana, Slovenia, May 2015, pp. 1–7.

[21] Y. Q. Shi, C. Chen, and W. Chen, "A Natural Image Model Approach to Splicing Detection," in *Proceedings of the 9th workshop on Multimedia & security*, Dallas, Texas, September 2007, pp. 51–62.

[22] Z. Zhang, J. Kang, and Y. Ren, "An Effective Algorithm of Image Splicing Detection," in *2008 International Conference on Computer Science and Software Engineering*, vol. 1, December 2008, pp. 1035–1039.

[23] Y. Zhang and C. Zhao, "Revealing Image Splicing Forgery Using Local Binary Patterns of DCT Coefficients," in *Communications, Signal Processing, and Systems*, New York, January 2012, pp. 181–189.

[24] W. Wang, J. Dong, and T. Tan, "Image Tampering Detection Based on Stationary Distribution of Markov Chain," in *2010 IEEE International Conference on Image Processing*, Hong Kong, China, September 2010, pp. 2101–2104.

[25] X. Zhao and J. Li, "Detecting Digital Image Splicing in Chroma Spaces," in *Digital Watermarking*, vol. 6526. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 12–22.

[26] Z. Xia, C. Yuan, X. Sun, D. Sun, and R. Lv, "Combining wavelet transform and LBP related features for fingerprint liveness detection," *IAENG International Journal of Computer Science*, vol. 43, no. 3, pp. 290—-298, April 2016.

[27] A. Alahmadi and M. Hussain, "Passive Detection of Image Forgery Using DCT and Local Binary Pattern," *Signal, Image and Video Processing*, vol. 11, no. 1, pp. 81–88, January 2017.

[28] A. L. Sandoval Orozco, D. M. Arenas González, J. Rosales Corripio, L. J. García Villalba, and J. C. Hernandez-Castro, "Source Identification for Mobile Devices, Based on Wavelet Transforms Combined with Sensor Imperfections," *Computing*, vol. 96, no. 9, pp. 829–841, September 2014.

[29] J. Rosales Corripio, D. M. Arenas González, A. L. Sandoval Orozco, L. J. García Villalba, J. C. Hernandez-Castro, and S. J. Gibson, "Source Smartphone Identification Using Sensor Pattern Noise and Wavelet Transform," Madrid, Spain, January 2013, pp. 1–6.

[30] C. C. Chang and C. J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[31] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[32] P. Liu, K.-K. R. Choo, L. Wang, and F. Huang, "Svm or deep learning? a comparative study on remote sensing image classification," *Soft Computing*, vol. 21, no. 23, pp. 7053–7065, 2017.

[33] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 4, pp. 778–784, 2014.

Guía Nacional de Notificación y Gestión de Ciberincidentes, Ventana Única e Indicadores

David Carlos Sánchez Cabello[1,2], Alberto Sánchez Del Monte [2], Ana Lucila Sandoval Orozco[1], Luis Javier García Villalba[1]*
[1]Grupo de Análisis, Seguridad y Sistemas (GASS)
[1]Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
{asandoval, javiergv}@fdi.ucm.es

[2]Centro Nacional de Protección de Infraestructuras y Ciberseguridad
{dcsc, asdm}@interior.es

Resumen- **La Guía Nacional de Notificación y Gestión de Ciberincidentes, aprobada por el Consejo Nacional de Ciberseguridad en enero de 2019, se ha elaborado bajo un espíritu eminentemente práctico por parte de aquellos organismos de la Administración española involucrados en la ciberseguridad. Por ello, ha sido necesario aportar toda aquella experiencia adquirida en la gestión de ciberincidentes a lo largo de los últimos años por CSIRT y autoridades competentes en la materia. El documento se ha diseñado para conformar una metodología de trabajo compartida y establecer unos indicadores y parámetros unificados, tanto para Operadores de Servicios Esenciales (OSE) y Proveedores de Servicios Digitales (PSD) con obligatoriedad de notificación de acuerdo al Real Decreto-ley 12/2018, como para ciudadanos, Administraciones públicas y empresas privadas. Así pues, este Guía define, entre otros, conceptos como el nivel de peligrosidad e impacto asociado a un incidente, una ventanilla única de notificación y una taxonomía homogénea.**

***Index Terms*- Guía Nacional de Notificación y Gestión de Ciberincidentes, Operador de Servicios Esenciales, Proveedor de Servicios Digitales, Operador Crítico, CNPIC, OCC, CSIRT, CCN-CERT, INCIBE-CERT, ESPDEF-CERT, Taxonomía, Nivel de peligrosidad, Nivel de impacto.**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Las redes y sistemas de información ejercen actualmente un papel crucial y determinante en el desarrollo y mantenimiento de las actividades económicas y sociales de los ciudadanos de la Unión Europea. Es por ello que, garantizar y reforzar su fiabilidad se ha convertido en uno de los objetivos prioritarios para el legislador. La creciente dependencia de las Tecnologías de Información y Comunicación (TIC) que ha experimentado la sociedad en los últimos años ha motivado el desarrollo de políticas comunitarias dirigidas hacia la consecución de unos criterios mínimos comunes que garanticen un adecuado desarrollo de capacidades de planificación, cooperación, coordinación y comunicación en lo que respecta a la protección de servicios esenciales y proveedores de servicios digitales de la UE.

La *Directiva (UE) 2016/1148, relativa a las medidas destinadas a garantizar un elevado nivel común de seguridad de redes y sistemas de información de la Unión* (Directiva NIS), ha implantado una senda a través de la que se desarrollan las políticas europeas respecto a la ciberseguridad [1]. La transposición de la anterior Directiva a la legislación española se realizó en 2018 mediante la publicación en el B.O.E. del *Real Decreto-ley 12/2018, de 7 de septiembre, de seguridad de las redes y sistemas de información*. Este Real Decreto-ley define el marco institucional de la ciberseguridad en España, conformado por autoridades públicas competentes y CSIRT, sin dejar de lado la cada vez más relevante colaboración público-privada. Junto a ello, se recogen y detallan en ese texto legislativo las obligaciones específicas respecto a la seguridad lógica para todos aquellos sujetos obligados por esta legislación –Operadores de Servicios Esenciales y Proveedores de Servicios Digitales-, con especial referencia a la comunicación de aquellos incidentes que presenten un carácter transfronterizo [2].

Con la finalidad de ofrecer una respuesta a todos aquellos interrogantes que pudiese generar la notificación de ciberincidentes a la Administración pública, bien sea de carácter obligatorio o potestativo esta comunicación, el Consejo Nacional de Ciberseguridad propuso la elaboración de una guía conjunta de notificación de incidentes. Para ello se conformó un Grupo de Trabajo interministerial liderado por el Centro Nacional par la Protección de Infraestructuras Ciberseguridad (CNPIC) en el que estaban presentes a su vez el Centro Criptológico Nacional (CCN), el Instituto Nacional de Ciberseguridad (INCIBE), y el Mando Conjunto de Ciberdefensa (MCCD). A raíz de los trabajos de ese grupo a lo largo de varios meses, el 9 de enero de 2019 veía la luz el documento tras su aprobación en el Consejo. En esta guía quedan definidos, entre otros conceptos, la ventanilla única a la que dirigirse en función de la naturaleza del operador que notifica el ciberincidentes, una nueva taxonomía o clasificación de eventos, así como una serie de indicadores que definen la peligrosidad e impacto de un ciberincidente.

II. Guía de notificación y gestión de ciberincidentes

La finalidad que se ha buscado alcanzar mediante la implantación de la Guía Nacional de Notificación y Gestión de Ciberincidentes es su consolidación como un documento de referencia a nivel nacional al que cualquier persona, física o jurídica, tenga obligación o no de notificar un incidente de ciberseguridad, pueda recurrir a la hora de enfrentarse a un suceso de este tipo acaecido en su infraestructura tecnológica. Por ello, y como consecuencia de este documento de consenso, todos los organismos nacionales implicados por ley en la notificación y gestión de incidentes emplearán una metodología de trabajo y herramientas comunes a la hora de actuar y enfrentarse a un incidente.

II-A. CSIRT de Referencia, autoridades competentes y Ventanilla Única.

Los CSIRT (*Computer Security Incident Response Team, Equipos de Respuesta ante Incidencias de Seguridad Informática*) ejercen un papel fundamental en la notificación de incidentes, de modo que son aquellos organismos encargados de supervisar incidentes a escala nacional, difundir alertas tempranas, efectuar un análisis dinámico de riesgos e incidentes, participar en la red europea de CSIRT, así como de ejercer su principal y más relevante función; responder a incidentes.

En función de la naturaleza pública o privada de un determinado organismo con obligación de notificación de un incidente, la Guía recoge lo establecido en la normativa legal a este respecto, es decir, que el CCN-CERT será el CSIRT de referencia para todos aquellos sujetos obligados a los que les sea de aplicación la ley 40/2015 (organismos públicos), el INCIBE-CERT será el equipo de respuesta para aquellos sujetos obligados a los que no les sea de aplicación esa ley (organismos privados), y finalmente el ESPDEF-CERT, que como CSIRT del Mando Conjunto de Ciberdefensa colaborará con los CSIRT anteriores, en especial en aquellos incidentes con afectación a la Seguridad Nacional del país. Adicionalmente, el INCIBE-CERT se constituye como el CSIRT de referencia para ciudadanos, entidades afiliadas a RedIris y empresas. Junto con estos organismos, que conforman la puerta de entrada y por tanto de conocimiento de los incidentes en la Administración, se encuentra el Centro Nacional de Protección de Infraestructuras y Ciberseguridad (CNPIC) en calidad de autoridad competente para aquellos Operadores de Servicios Esenciales que a su vez sean Operadores de Infraestructuras Críticas. Las principales funciones de la autoridad competente serán la supervisión de la seguridad, la recepción de las notificaciones a través de los CSIRT, y el ejercicio de la potestad sancionadora. Así mismo, este organismo se consolida como el enlace con las Fuerzas y Cuerpos de Seguridad del Estado en caso de que un determinado incidente pueda constituir un ilícito penal, a través de la Oficina de Coordinación Cibernética (OCC).

La Guía establece un sistema de ventanilla única de notificación (véase Figura 1).

1. De este modo que, mediante la notificación inicial del incidente a través del formulario al efecto recogido en la Guía, remitido mediante email o mediante herramienta de ticketing, se comunica al CSIRT de referencia a través del canal especificado por cada uno de ellos.
2. Se realiza la sincronización unidireccional con el organismo o autoridad competente.
3. En este punto se da inicio a la investigación del incidente.
4. Se podrá requerir una notificación completa del incidente a través del formulario específico del organismo
5. Finalmente, si así se considera, se procederá a comunicar a Ministerio Fiscal y Fuerzas y Cuerpos de Seguridad del Estado los hechos, en el caso de que pueda observarse la posible existencia de un ilícito penal en el incidente.

Figura 1. Ventanilla única de notificación

II-B. TAXONOMÍA.

Uno de los puntos más relevantes del documento es la inclusión de una nueva taxonomía o clasificación de incidentes. Para ello se ha tomado como referencia los trabajos que se desarrollan en la actualidad a nivel europeo desde la Agencia Europea de Redes y Sistemas de Información (ENISA), junto con el TF-CSIRT (*Task Force on Computer Security Incident Response Team, Grupo de Trabajo de Equipos de respuesta ante incidencias de seguridad informática*) en el Grupo de Trabajo creado al efecto [3].

II-C. INDICADORES.

La Guía Nacional de Notificación y Gestión de Ciberincidentes define una serie de indicadores y establece a su vez pautas específicas ligadas a ellos que informan acerca de cómo actuar a partir de la interpretación de los mismos. Estos indicadores son el nivel de peligrosidad y el nivel de impacto asociados a un incidente. En función de la superación de una serie de umbrales ligados a estos niveles será preceptiva la comunicación a la autoridad competente, al entenderse que el incidente presenta efectos perturbadores en la infraestructura tecnológica del sujeto obligado. Estos indicadores se referencian mediante cinco grados o niveles que indican la mayor o menor afectación en cuanto a la peligrosidad o impacto que presenta un determinado suceso, niveles definidos como; crítico, muy alto, alto, medio, y bajo.
Los criterios de determinación del nivel de peligrosidad de los ciberincidentes atienden a una categorización de los ciberincidentes por razón de su naturaleza, y ligado por tanto de forma inequívoca a la taxonomía o clasificación, siendo susceptible de variación si a lo largo del desarrollo de la gestión si se entiende que cambia su tipología.
A modo de ejemplo, se definen como ciberincidentes con nivel de peligrosidad crítico los Daños Informáticos PIC, esto es, aquellas conductas relacionadas con la perturbación de datos o programas informáticos de una infraestructura crítica o relativos a un servicio esencial cuando tengan carácter relevante; los incidentes relacionados con Ciberterrorismo; o aquellos categorizados como Amenazas Persistentes Avanzadas (APT). Posteriormente, una segunda valoración del ciberincidente se basará en las consecuencias que ha generado el incidente en las redes o sistemas afectados, quedando definido de este modo el nivel de impacto. En caso de una posible asignación de varios niveles a un determinado hecho, en todo caso, se asignará siempre el nivel de peligrosidad e impacto más elevado de todos aquellos contemplados. Por otro lado, la Guía establece que será preceptivo, para aquellos sujetos obligados, la comunicación de un incidente categorizado con un nivel de peligrosidad o impacto Crítico, Muy Alto o Alto, siendo el resto de los incidentes de notificación voluntaria.

II-D. NOTIFICACIÓN Y GESTIÓN DE INCIDENTES DE UN OPERADOR CRÍTICO.

Debido a la especial y diferenciado tratamiento que el legislador ha considerado para aquellos sujetos que sean nombrados Operadores Críticos de acuerdo a la ley 8/2011, se ha introducido en la Guía un anexo específico destinado a detallar las singularidades de la notificación de estos sujetos. Bajo las premisas anteriores, y una vez un OSE que tenga condición de Operador Crítico detecte en su infraestructura tecnológica un incidente categorizado con un nivel de peligrosidad o impacto de obligado reporte, deberá comunicar el mismo al CNPIC a través de su CSIRT de referencia, de acuerdo con el sistema de ventanilla única expresada en el punto II-A.
Así pues, y tal y como se indica en la Guía, se notificarán aquellos incidentes de obligado reporte acaecidos en las redes y sistemas de información que soportan los servicios esenciales prestados por las infraestructuras del Operador [3]. Para ello, y tal y como se detalla en la Sección II-A de este texto, será necesario hacer referencia a una serie de datos e informaciones recogidas en un formulario al efecto, contenido en el anexo de la Guía Nacional de Notificación. Junto a ello, estos sujetos deberán cumplir la ventana temporal de reporte marcada en el anexo correspondiente, de modo que mediante su aplicación, la Administración tenga conocimiento oportuno del suceso.

III. CONCLUSIONES

La Guía Nacional de Notificación y Gestión de Ciberincidentes ha sido desarrollada para ser empleada por la comunidad de referencia como una herramienta práctica, eficaz e integral para la notificación de ciberincidentes, dando respuesta a las cuestiones que se puedan llegar a plantear acerca de a quién, cómo y cuándo notificar un incidente de Ciberseguridad. Mediante su publicación se ha buscado la homogeneización de criterios contenidos en los distintos manuales o guías publicadas por autoridades o CSIRT nacionales a lo largo de los últimos años, de modo que constituye un salto cualitativo en la construcción de un marco legislativo potente y robusto en materia de Ciberseguridad a nivel nacional y europeo.

REFERENCIAS

[1] Parlamento Europeo, Directiva 2016/1148 del 6 de julio de 2016
[2] Real Decreto-ley 12/2018, de 7 de septiembre, de seguridad de las redes y sistemas de información
[3] https://github.com/enisaeu/Reference-Security-Incident-Taxonomy-Task-Force
[4] Consejo Nacional de Ciberseguridad, Guía Nacional de Notificación y gestión de Ciberincidentes.

El Efecto de la Transposición de la Directiva NIS en el Sector Estratégico TIC de la ley 8/2011

David Carlos Sánchez Cabello[1,2], Ana Lucila Sandoval Orozco[1], Luis Javier García Villalba[1]*

[1]Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
{asandoval, javiergv}@fdi.ucm.es

[2]Centro Nacional para la Protección de Infraestructuras y ciberseguridad
dcsc@interior.es

Resumen- **Las redes y sistemas de información desempeñan un papel crucial en la sociedad. Su fiabilidad y seguridad son esenciales para las actividades económicas y sociales, y en concreto para el correcto funcionamiento de los mercados tanto nacionales como internaciones. Debido a ese carácter transnacional, un incidente acaecido en esos sistemas informáticos ya sea o no deliberado, y con independencia del lugar en que se produzca, puede afectar a diferentes Estados miembros y a la Unión en su conjunto a través de su mercado interior. Esto permite entender que, la seguridad de las redes y sistemas de información es fundamental para el correcto funcionamiento del mercado. Tanto la directiva NIS como su transposición tienen por objetivo proteger todos estos sectores definiéndolos concretamente en 6. Toda esta nueva regulación ha de encajar con la regulación ya existente como la ley 8/2011 por la que se establecen medidas para la protección de las infraestructuras críticas y el Plan Estratégico Sectorial de las Tecnologías de la Información y la Comunicación (TIC).**

***Index Terms*- Directiva NIS, Operador Servicios Esenciales, Tecnologías de la Información y Comunicación, PNPIC, PES, PSO, PPE, PAO**

Tipo de contribución: *Investigación en desarrollo*

I. INTRODUCCIÓN

Con el espíritu de proteger el mercado interior de la Unión Europea se aprobó la Directiva de la Unión Europea 2016/1148 relativa a las medidas destinadas a garantizar un elevado nivel común de seguridad de las redes y sistemas de información en la Unión (Directiva NIS en adelante).

El objetivo global de la Directiva NIS es lograr un elevado nivel común de seguridad de las redes y sistemas de información en la Unión Europea, a fin de mejorar el funcionamiento del mercado interior.

Para ello, se insta a los Estados Miembros a estar más preparados, adoptar una Estrategia Nacional de Seguridad de las Redes e Información, incrementar la cooperación estratégica y el intercambio de información entre ellos así como a identificar a los Operadores de Servicios Esenciales (en adelante OSE).

Por otro lado, exige que los OSE (definidos estos 6 servicios esenciales en el anexo II de la Directiva como, sector energía, sector transporte, banca y servicios financieros, sector sanitario, sector agua y sector infraestructura digital, tales como los motores de búsqueda y la nube donde almacenan sus datos), adopten medidas para mejorar su capacidad de resistencia y resiliencia ante los ataques cibernéticos, incidiendo en la obligatoriedad de los OSE y de los proveedores de servicios digitales de reportar ciberincidentes a las respectivas autoridades nacionales.

II. INTERRELACIÓN DE LA DIRECTIVA NIS Y LA NORMATIVA PIC

La trasposición de esta Directiva NIS en España se ha realizado mediante la aprobación del Real Decreto-Ley 12/2018 de 7 de septiembre.

Esta directiva define una serie de conceptos importantes en paralelo con los establecidos en la normativa PIC, entre ellos se encuentra la definición de lo que entiende esta normativa comunitaria sobre quiénes son los OSE en su artículo 5.2: "aquella entidad que presta un servicio esencial para el mantenimiento de actividades sociales o económicas cruciales, donde la prestación de dicho servicio depende de los sistemas de redes y de información, y en las que un incidente tendría consecuencias perjudiciales significativas para la prestación de dicho servicio"[1].

Esta definición es la piedra angular, donde recae todo el peso normativo de la Directiva NIS y en donde se centra sin duda la transposición que se llevó a cabo en la normativa española.

La primera norma que definió qué eran los servicios esenciales fue la Directiva 2008/114/CE del Consejo, sobre Identificación y Designación de las Infraestructuras Críticas Europeas y la Evaluación de la Necesidad de Mejorar su Protección definiéndolo, que los definió "como todas las actividades destinadas a garantizar la funcionalidad, continuidad e integridad de las infraestructuras críticas con el fin de prevenir, paliar y neutralizar una amenaza, riesgo o vulnerabilidad" [2].

Esta Directiva, 2008/114/CE, solo definía dos sectores especialmente protegidos, energía y transporte, indicando la posibilidad de poder ampliarse al Sector de las Tecnologías de la Información y de las Comunicaciones, dejando la puerta abierta a que cada estado miembro definiese los que considerase, a parte de los mencionados.

Durante el año 2011, España realizó la transposición de esta Directiva, y a consecuencia de la misma nació la normativa PIC, con la publicación de la Ley 8/2011, de 28 de abril, por la que se establecen medidas para la protección de las infraestructuras críticas (Ley PIC) y su Reglamento de

desarrollo, el Real Decreto 704/2011, de 20 de mayo (Reglamento PIC), que fueron aún más ambiciosas que la propia Directiva, ya que desde el momento de su aprobación la ley PIC regula la protección y seguridad de doce sectores estratégicos[3].

Estos doce sectores que se regulan en la normativa PIC, ya cubren la totalidad de los sectores de los servicios esenciales de los que habla la Directiva NIS por lo que no es necesaria su modificación a causa de la transposición.

El objeto principal que se desarrolla en la normativa de Protección de Infraestructuras Criticas (en adelante PIC) es la implantación, adopción y aplicación de una serie de planes de actuación: Plan Nacional Protección de Infraestructuras Criticas (PNPIC), Planes Estratégicos Sectoriales (PES), Planes de Seguridad del Operador (PSO), Planes de protección Específicos (PPE) y Planes de Apoyo Operativo (PAO).

El establecimiento de una serie de estrategias y estructuras adecuadas que permitían dirigir y coordinar las actuaciones de los distintos órganos de las Administraciones Públicas en la protección de aquellas infraestructuras estratégicas donde descasan los servicios esenciales, y primordialmente la protección de aquellas infraestructuras que sean denominadas infraestructuras estratégicas.

Estas estrategias se definen en el PES, existiendo uno por sector, y encontrándose aprobado el correspondiente al sector TIC desde junio de 2017.

La normativa PIC define en su artículo 2 los servicios esenciales como aquellos "servicios necesarios para el mantenimiento de las funciones sociales básicas, la salud, la seguridad, el bienestar social y económico de los ciudadanos o el eficaz funcionamiento de las Instituciones del Estado y de las Administraciones Públicas".

Fig. 1. Relación entre los 12 Servicios Esenciales definidos en la ley PIC y los 6 Servicios Esenciales definidos en la directiva NIS (rojo)

El normal funcionamiento de los servicios esenciales que se prestan a la ciudadanía descansa en una serie de infraestructuras de gestión tanto pública como privada, cuyo funcionamiento es indispensable, conocidas como Infraestructuras Criticas.

Por lo tanto, esta definición cumple en gran medida los criterios de definición «operador de servicios esenciales» de la Directiva NIS, y donde la eficacia de estos servicios esenciales radica en las redes y sistemas de información.

También es de resaltar por un lado que la Directiva NIS en el art. 14 recoge los requisitos de seguridad y de notificación de incidentes para los OSE y por otro que a nivel de la ley PIC estos requisitos ya se encuentran definidos en el Plan de Seguridad del Operador que ha de confeccionar cada operador crítico, donde debe llevar a cabo una política general de seguridad.

Estas notificaciones de incidentes de ciberseguridad a las que obliga la Directiva NIS quedan definidas en la Guía Nacional de Notificación y Gestión de Ciberincidentes del Consejo Nacional de Ciberseguridad aprobada el 9 de enero de 2019 [4] homogenizando la notificación de ciberincidentes tanto para OSE como Operadores Críticos.

A nivel normativo aún queda pendiente la aprobación del Reglamento que articulara la transposición de la directiva NIS, en el que se espera un régimen sancionador para todos aquellos OSE que no cumplan los requerimientos y las obligaciones que impone la Directiva NIS como no notificar algún ciberincidente, o no hacerlo en tiempo y forma, tal como ha dejado definido la guía de notificación, de los que tenga obligación, como son los incidentes considerados con un nivel crítico, muy alto o alto.

III. Conclusiones

Desde la aprobación de la normativa sobre PIC, en Europa mediante la aprobación de la Directiva 2008/114/CE del 8 de diciembre en la que se definían solo dos sectores estratégicos que debían ser especialmente protegidos, España ha realizado una gran labor normativa adelantándose a las sucesivas normas europeas siendo un referente para otros estados miembros con su legislación PIC.

Esto ha permitido que al aprobar la transposición de la Directiva NIS mediante el oportuno Real Decreto Ley no haya sido necesario modificar en todo y ni siquiera en parte la legislación vigente en materia PIC con la que coexiste en 6 de los 12 sectores estratégicos definidos en España desde el año 2011.

Referencias

[1] Parlamento Europeo, Directiva (UE) 2016/1148, de 6 de julio de 2016
[2] Consejo Europeo, Directiva 2008/114/CE, de 8 de diciembre de 2008
[3] Ley 8/2011, de 28 de abril de 2011
[4] Consejo Nacional de Ciberseguridad, Guía Nacional de Notificación y gestión de Ciberincidentes,

CyberHeroes: Aplicación móvil para fomentar el buen uso de la tecnología e Internet en menores

Mario González[1], Gregorio López[1,2], Víctor Villagrá[1]
[1]Universidad Politécnica de Madrid, Avda. Complutense 30, 28040, Madrid
[2]Universidad Pontificia Comillas – ICAI, C/ Alberto Aguilera 25, 28015, Madrid
mario.gonzalez.lopez@alumnos.upm.es, gllopez@comillas.edu, victor.villagra@upm.es

Resumen-**El uso de la tecnología y el acceso a Internet de los menores es un asunto de capital importancia para la sociedad de hoy en día, que atrae cada vez más atención. Este artículo presenta un prototipo de juego de preguntas y respuestas desarrollado para iOS cuyo objetivo es precisamente concienciar y fomentar el buen uso de Internet y de la tecnología en menores. Asimismo, el artículo esboza futuras oportunidades y líneas de investigación a las que puede dar lugar esta primera iniciativa.**

***Index Terms*- app, formación, Internet, juego, menores, ciberseguridad**

Tipo de contribución: *Formación innovación*

I. Motivación

Según el informe de UNICEF "Estado Mundial de la Infancia 2017: Niños en un mundo digital", los niños y adolescentes menores de 18 años representan aproximadamente uno de cada tres usuarios de Internet en todo el mundo [1]. Aunque estos niños y adolescentes pueden considerarse nativos digitales, en ocasiones no son conscientes de los riesgos, amenazas, beneficios y oportunidades que la tecnología e Internet involucran, como les ocurre en muchos otros aspectos de sus vidas. Por lo tanto, este segmento de población puede considerarse como el más expuesto a los riesgos y amenazas asociados al mundo conectado en el que vivimos, pero, al mismo tiempo, también presenta el mayor potencial para sacarle el máximo partido a sus beneficios y oportunidades. Así, el propio informe de UNICEF no sólo llama la atención sobre los riesgos que un mal uso de Internet y la tecnología puede conllevar, sino también del riesgo de exclusión al que se exponen los menores que no tengan acceso, instando tanto a los sectores públicos como privados a que trabajen en ambas direcciones.

La Comisión Europea, en su informe "*European Strategy for a Better Internet for Children*" del año 2012 [2] también identificaba que este segmento de población es especialmente relevante y proponía una estrategia articulada en torno a cuatro pilares fundamentales: (1) estimular la creación de contenido *online* de calidad para menores; (2) aumentar la concienciación de los menores sobre el uso de Internet y la tecnología y dotarlos de herramientas que les permitan sacarle el máximo partido ("empoderarlos"); (3) crear un entorno seguro para los menores en Internet; y (4) luchar contra el abuso y la explotación sexual de menores *online*.

Parece por tanto que, en este ámbito, como ocurre en muchos otros como, por ejemplo, la ciberseguridad en general, las principales soluciones pueden clasificarse en dos grandes grupos: soluciones orientadas a formar y educar, relacionadas con los pilares (1), (2) y (4) de la estrategia europea, y soluciones orientadas a prevenir, detectar y/o actuar automáticamente ante determinadas situaciones consideradas peligrosas, relacionadas con los pilares (3) y (4). Dentro de este último grupo pueden encontrarse aplicaciones de control parental como, por ejemplo, Qustodio, o las versiones adaptadas a menores de algunas aplicaciones muy populares, como, por ejemplo, YouTube Kids. La iniciativa europea "*Better Internet for Kids*" [3] representa un buen ejemplo de las medidas tomadas por la Unión Europea dentro de la primera categoría, que es en la que se centra este trabajo.

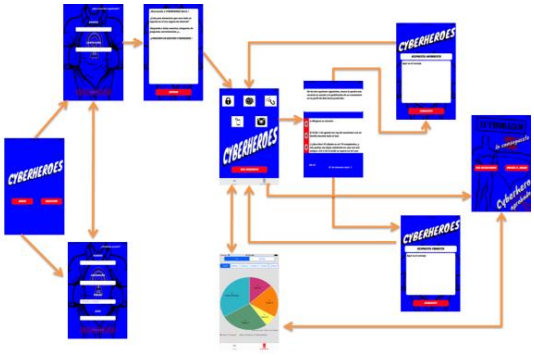
II. Objetivos

Concretamente, este trabajo pretende hacer uso de una herramienta que los niños han utilizado para conocer mejor su entorno desde tiempos ancestrales: el juego. Pero para que surta efecto debe tratarse de un juego adaptado a los nativos digitales a los que va dirigido. Así, en el marco de una iniciativa que involucró varios TFG dentro del programa GITST de la ETSIT de la UPM se decidió desarrollar un juego de preguntas y respuestas relacionadas con la ciberseguridad en Internet y con el uso de la tecnología para tres plataformas: *iOS*, *Android*, y *Facebook*. Este artículo presenta la aplicación desarrollada para *iOS* y esboza futuras oportunidades y líneas de investigación a las que puede dar lugar esta primera iniciativa.

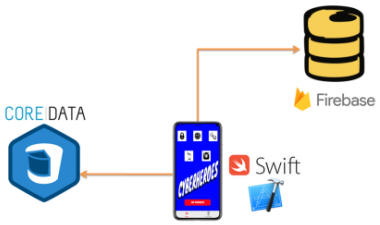
III. Descripción de la aplicación desarrollada

El juego diseñado consta de cinco categorías de preguntas: (1) Credenciales/Privacidad, (2) Ingeniería social/*Phising*, (3) Redes sociales/Aplicaciones de mensajería, (4) Uso del terminal, y (5) Uso de Internet.

La Fig.1.(a) ilustra el flujo de navegación y el funcionamiento del juego. En primer lugar, nos encontramos con una pantalla de inicio, que nos permite registrarnos o iniciar sesión. Si iniciamos sesión correctamente accedemos a una pantalla de bienvenida que sirve de antesala a la pantalla en la que se sortean las categorías. Esta pantalla selecciona aleatoriamente una categoría y permite acceder a la pantalla en la que se muestra la pregunta asociada a la categoría en cuestión. El jugador tiene un tiempo limitado para responder a la pregunta. Una vez respondida, se le lleva a una pantalla en



(a)



(b)

Fig. 1. (a) Flujo de navegación de la aplicación CyberHeroes; (b) Arquitectura y tecnologías utilizadas para el desarrollo de la *app*

la que se le informa de si ha acertado o fallado y, en ambos casos, se le da información relacionada con la pregunta para fomentar el aprendizaje. Si acierta la pregunta, ya no se le volverán a formular preguntas de esa categoría. Cuando se aciertan preguntas de todas las categorías, se da paso a la pantalla de enhorabuena, en la que se permite volver a jugar o ver los resultados. La pantalla de resultados permite ver tanto los resultados obtenidos por uno mismo (p.ej., número medio de intentos necesarios para acabar el juego o para acertar la pregunta de una categoría) como compararse con otros usuarios de manera agregada (p.ej., por edad o por sexo) con el objetivo de fomentar una competitividad "sana".

La Fig.1(b) muestra la arquitectura de la aplicación y las tecnologías que se han utilizado para desarrollarla. Como puede verse, la aplicación ha sido desarrollada usando principalmente *Swift* como lenguaje de programación. El *framework* de *iOS CoreData* ha sido utilizado para diseñar y proporcionar persistencia al modelo de datos local que se utiliza para almacenar la información de registro e inicio de sesión, así como las preguntas y respuestas. *Firebase* se utiliza como base de datos para almacenar los resultados anonimizados y agregados (p.ej, por edad, por sexo). Se eligió *Firebase* por varios motivos: porque no requería de una infraestructura propia al estar alojada en la "nube" de *Google*, porque permitía conectar fácilmente (mediante un ID) los juegos desarrollados para las diferentes plataformas (*iOS*, *Android* y *Facebook*), y por su actual auge en aplicaciones comerciales.

IV. Conclusiones y Trabajos Futuros

Este artículo presenta brevemente un juego de preguntas y respuestas desarrollado para *iOS* cuyo objetivo es formar a los menores en el uso de la tecnología e Internet. Este juego fue presentado en la 7ª edición de la Jornada X1Red+Segura [4] percibiéndose una buena acogida. Sin embargo, se trata de una prueba de concepto que está lejos de poder conseguir el objetivo que se propone. Por ejemplo, actualmente se dispone de una decena de preguntas para cada categoría, lo que limita la jugabilidad. Dichas preguntas deben ser diseñadas por sociólogos y psicólogos, adaptándose a la edad de los jugadores, y la parte visual también debe mejorarse.

El utilizar aplicaciones o videojuegos con el objetivo de concienciar y educar a los menores en el uso de la tecnología no es una idea novedosa. En 2015, la Junta de Andalucía sacó la *app* "Aprende seguridad en la red", que incluía una versión del clásico encuentras las parejas centrado en ciberseguridad (MemoTIC), un juego de preguntas y respuestas tipo "Trivial" (PreguntasTIC) y un juego que retaba al usuario a adivinar si realmente realizaba un uso adecuado de Internet (RetoTIC). Asimismo, en la página web de INCIBE, dentro de is4k (*internet segura for kids*) [5], puede encontrarse *Cyberscouts*, orientado a que toda la familia pueda poner a prueba sus conocimientos sobre ciberseguridad.

Sin embargo, este tipo de juegos y aplicaciones presentan un potencial desde el punto de vista científico y de investigación aún sin explotar. Parece razonable pensar que, si las preguntas las diseñan psicólogos y sociólogos adecuadamente, las respuestas a las mismas podrían utilizarse para "medir" el uso que los menores hacen de la tecnología y de Internet de manera anónima y agregada, representando una alternativa poco o nada invasiva y amigable a las encuestas en papel que se utilizan actualmente con este propósito en proyectos de referencia como *EU Kids Online* [6]. Yendo un paso más allá, el uso de juegos adecuadamente diseñados podría permitir realizar un modelado de usuarios que permitiera a su vez detectar perfiles en riesgo frente a determinadas amenazas o más vulnerables a sufrir abusos *online* (de manera similar a lo que se propone el novedoso y ambicioso proyecto europeo *IBSEN* para modelar el comportamiento humano en general [7]) y tomar las medidas oportunas para protegerlos. No obstante, para llegar a ese punto hay que superar antes un obstáculo para nada baladí: conseguir que ese tipo de aplicaciones las utilicen de manera entusiasta los usuarios objetivo. En cualquier caso, el bienestar *online* de nuestros hijos parece que bien vale el esfuerzo.

Agradecimientos

Los autores quieren dar las gracias al resto de estudiantes que desarrollaron su TFG sobre este tema (Gabriel Díez, Sandra Romero y Javier Vázquez), así como a Ángel Pablo Avilés por su inestimable ayuda y apoyo incondicional durante la realización de estos TFG.

Referencias

[1] UNICEF, "Estado Mundial de la Infancia 2017: Niños en un mundo digital", 2017.
[2] EC. "*European Strategy for a Better Internet for Children*", 2012.
[3] Better Internet for kids: https://www.betterinternetforkids.eu/
[4] X1Red+Segura 2019: https://www.x1redmassegura.com/
[5] is4k: https://www.is4k.es/
[6] *EU Kids Online*: http://www.lse.ac.uk/media-and-communications/research/research-projects/eu-kids-online
[7] IBSEN: https://ibsen-h2020.eu/

A Generic Solution for Authenticated Group Key Establishment From Key Encapsulation – a Compiler for Post-Quantum Primitives

Edoardo Persichetti
Florida Atlantic University
Department of Mathematical Sciences
epersichetti@fau.edu

Rainer Steinwandt
Florida Atlantic University
Department of Mathematical Sciences
rsteinwa@fau.edu

Adriana Suárez Corona
Universidad de León
Research Institute of Applied Sciences to Cybersecurity (RIASC)
asuac@unileon.es

***Abstract*—We present a generic group key exchange construction that builds on a key encapsulation mechanism and a signature scheme. When applied to existing post-quantum proposals, the compiler provides a three-round solution for authenticated group key establishment that is quantum resistant and requires only one signature per user.**

***Index Terms*—Authenticated Group Key Establishment, Key Encapsulation Mechanism, Post-Quantum Cryptography**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Group key establishment protocols allow a group of participants to agree on a common secret key that can be used afterwards with symmetric key cryptographic tools. Although it is a fairly well understood protocol task, there are different security models trying to capture different properties. In particular, if a key establishment protocol is secure against active adversaries, it is called Authenticated Group Key Establisment (AGKE).

A standard technique to derive an AGKE solution is to apply some form of protocol compiler or generic framework to a passively secure solution. If a public-key infrastructure is available, signatures provide an adequate mechanism. This results commonly in protocols with a substantial number of signatures being computed, transmitted, and verified [1], [2], [3]. Aiming at post-quantum solutions, the cost of integrating signatures in a protocol can differ quite a bit from the familiar setting. Specifically, for hash-based designs, arguably one of the most popular approaches for post-quantum signing, the size of signatures remains an issue. For instance, proposed instances of the SPHINCS$^+$ design have signature lengths between $8,080$ and $49,216$ bytes [4]. Taking this into account, a compiler by Tang and Mitchell [5] appears more attractive from today's post-quantum perspective: Assuming the *a priori* availability of a unique session identifier that is distributed among the protocol participants, Tang and Mitchell present a compiler where each participant computes and transmits only one signature (and performs signature verifications).

A popular AGKE building block in the "pre-quantum" scenario is the traditional Diffie-Hellman two-party key exchange. This one-round protocol for two parties enables elegant two-round solutions for group key establishment—the Burmester-Desmedt protocol [6] offering a prominent design. Using a compiler (or a tailored design approach), deriving an AGKE solution with two or three rounds has by now become standard practice. Regrettably, owing to Shor's algorithm for solving discrete logarithms [7], in the post-quantum setting, the Diffie-Hellman protocol is no longer a viable building block for AGKE.

Our contribution. With the current state of the the art, some form of key encapsulation mechanism (KEM) is a natural starting point for implementing key establishment solutions in a post-quantum setting. Below we show how to derive a three-round AGKE from IND-CPA secure KEMs, where no participant signs more than one message.

II. Cryptographic tools and security

As main technical tools, we will make use of `IND-CPA` key encapsulation mechanism (KEM) to send ephemeral keys in a confidential manner. Moreover, to obtain an authenticated group key establishment, we will use a `UF-CMA` secure digital signature scheme.

We briefly recall KEM and refer to [8] for definitions regarding signature schemes and their corresponding security notions and to [9] for the security model considered for AGKE, which captures forward secrecy, as well as Man-In-The-Middle attacks.

Definition II.1 (Key Encapsulation Mechanism)**.** *A* key encapsulation mechanism (KEM) *is a triple of polynomial time algorithms* $(\texttt{KeyGen}, \texttt{Encaps}, \texttt{Decaps})$ *along with a finite keyspace* $\mathcal{K}$ *as follows:*

- `KeyGen` *is probabilistic. Given the security parameter* ℓ*, it generates a pair of public and secret keys* (pk, sk).
- `Encaps` *is probabilistic. Given a public key* pk*, it generates a pair* (K, C) *where* $K \in \mathcal{K}$ *is a symmetric key and* C *is an encapsulation of this key under the public key* pk.
- `Decaps` *is deterministic. Given a secret key* sk *and an encapsulation* C*, this algorithm outputs the symmetric key* K *or a special error symbol* $\perp$.

We require that for all key pairs (pk, sk) *generated by* `KeyGen` *if* $(K, C) = \texttt{Encaps}(pk)$*, then* $\texttt{Decaps}(sk, C) = K$ *is satisfied.*

Informally, a KEM is `IND-CPA` secure, if any ppt adversary, has negligible advantage in distinguishing whether an encapsulation C under a public key pk corresponds to the encapsulated symmetric key or a random one.

III. Construction

We present in figure 1 a generic construction, making use of a key encapsulation mechanism $\mathcal{KEM}$ and a signature scheme $\mathcal{S}$.

Set up:

We assume a pair of keys $(vk_i, sigk_i)$ for the signature scheme $\mathcal{S}$ is generated for each user U_i, which gets the secret key $sigk_i$ while vk_i is publicized.

Round 1:

- Each U_i generates an ephemeral pair of keys for $\mathcal{KEM}$:
$$(pk_i, sk_i) \leftarrow \texttt{KeyGen}(\ell)$$
- Each U_i sends to his right neighbor (pk_i, U_{i+1})

Round 2:

- Each U_i, using the ephemeral public key he has received from his left neighbor, computes an encapsulation for that key
$$(C_{i-1}, K_{i-1}) \leftarrow \texttt{KeyEncaps}(pk_{i-1});$$
- Each U_i sends to his left neighbor the pair (C_{i-1}, U_{i-1})

Round 3:

- Each U_i, using its ephemeral secret key, decapsulates the encapsulation he has received
$$K_i \leftarrow \texttt{KeyDecaps}(sk_i, C_i);$$
- Uses the encapsulated key he sent before to compute $X_i = K_{i-1} \oplus K_i$
- Uses $sigk_i$ to compute a signature σ_i of $pk_0, \ldots, pk_{n-1}, C_0, \ldots, C_{n-1}, X_i, pid_i$
- Each U_i broadcasts (X_i, σ_i)

Key Computation:

Each U_i checks all the signatures, equality of pid's, $X_0 \oplus \cdots \oplus X_{n-1} = 0$;
If something fails, aborts. Otherwise, each U_i:

- Computes $K_0 = K_i \oplus \bigoplus_{h=1}^{i} X_h$
- sets the session key $K := K_0$
- sets the session identifier

$sid := pk_0, ..., pk_{n-1}, C_0, ..., C_{n-1}, X_0, ..., X_{n-1}, pid_i$

Fig. 1. A Compiler for AGKE from KEM

Theorem III.1. *Let $\mathcal{S}$ be an* UF-CMA *signature scheme and $\mathcal{KEM}$ be an* IND-CPA *key encapsulation mechanism. Then, the protocol from Figure 1 is correct, authenticated and secure according to the security model in [9].*

To instantiate the AGKE construction we suggest several options both for KEMs and signature schemes. Some possible choices for quantum resistant KEM include [10], [11], [12], [13]. Moreover, we can propose [14], [4], as examples of post-quantum signature schemes. The particular choice can be made taking into account their efficiency or the assumption their security relies on.

Acknowledgements

Adriana Suarez Corona is partially supported by research project MTM2017-83506-C2-2-P, funded by the Spanish Ministerio de Economia y Competitividad.

References

[1] J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," *J. Cryptology*, vol. 20, no. 1, pp. 85–113, 2007.

[2] E. Bresson, M. Manulis, and J. Schwenk, "On Security Models and Compilers for Group Key Exchange Protocols," in *Advances in Information and Computer Security, Second International Workshop on Security, IWSEC*, 2007, pp. 292–307.

[3] J. Bohli, "A Framework for Robust Group Key Agreement," in *Computational Science and Its Applications - ICCSA 2006*, 2006, pp. 355–364.

[4] D. J. Bernstein *et al.*, "SPHINCS$^+$. Submission to the NIST post-quantum project," November 2017.

[5] Q. Tang and C. J. Mitchell, "Efficient compilers for authenticated group key exchange," in *Computational Intelligence and Security, International Conference, CIS 2005*, 2005, pp. 192–197.

[6] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system (extended abstract)," in *Advances in Cryptology - EUROCRYPT '94*, 1994, pp. 275–286.

[7] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.

[8] E. Kiltz, V. Lyubashevsky, and C. Schaffner, "A concrete treatment of fiat-shamir signatures in the quantum random-oracle model," in *Advances in Cryptology - EUROCRYPT 2018*, 2018, pp. 552–586.

[9] J. Bohli, M. I. G. Vasco, and R. Steinwandt, "Secure group key establishment revisited," *Int. J. Inf. Sec.*, vol. 6, no. 4, pp. 243–254, 2007.

[10] P. S. L. M. Barreto, S. Gueron, T. Güneysu, R. Misoczki, E. Persichetti, N. Sendrier, and J. Tillich, "CAKE: code-based algorithm for key encapsulation," in *Cryptography and Coding - 16th IMA International Conference, IMACC*, ser. Lecture Notes in Computer Science, M. O'Neill, Ed., vol. 10655. Springer, 2017, pp. 207–226.

[11] G. Banegas, P. Barreto, B. O. Boidje, P. Cayrel, G. N. Dione, K. Gaj, C. T. Gueye, R. Haeussler, J. B. Klamti, O. Ndiaye, D. T. Nguyen, E. Persichetti, and J. E. Ricardini, "DAGS: key encapsulation using dyadic GS codes," *J. Mathematical Cryptology*, vol. 12, no. 4, pp. 221–239, 2018. [Online]. Available: https://doi.org/10.1515/jmc-2018-0027

[12] J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the ring! practical, quantum-secure key exchange from LWE," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 1006–1018.

[13] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS - kyber: A cca-secure module-lattice-based KEM," in *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*. IEEE, 2018, pp. 353–367.

[14] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018, no. 1, pp. 238–268, 2018.

Seguridad y Privacidad en el Internet de las Cosas

Alejandra Guadalupe Silva Trujillo*, Jesús Gerardo Heredia Guerrero*, Pedro David Arjona Villicaña*
Ana Paola Juárez Jalomo*, Ana Lucila Sandoval Orozco†

* Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí
Av. Manuel Nava No. 8, Zona Universitaria, C.P. 78290, San Luis Potosí, S.L.P., México
Email: asilva@uaslp.mx, gerardo.hered@gmail.com, david.arjona@uaslp.mx, anajrzj@gmail.com

†Department of Electrical Engineering, Faculty of Technology, University of Brasilia (UnB)
Campus Universitario Darcy Ribeiro,Brasilia CEP 70910-900, Brazil
Email: asandoval@redes.unb.br

***Resumen*—Las tendencias de la Industria 4.0 están fuertemente relacionadas con el Internet de las cosas (IoT). Varias áreas, como la industrial, biomédica, educativa y de entretenimiento, exigen cada vez más el uso de sistemas integrados para ofrecer una mejor experiencia de usuario a través de la conectividad y el uso efectivo de las tecnologías. Estos dispositivos generan, procesan e intercambian una gran cantidad de información, de la cual gran parte se considera información confidencial. Los ataques a los dispositivos IoT son críticos porque pueden ocasionar daños físicos e incluso amenazar la vida de un ser humano. El objetivo de este documento es proporcionar una introducción al funcionamiento de los sistemas IoT, para conocer las opciones que se han desarrollado para la protección de información sensible y los desafíos que permanecen latentes, que es donde se necesita más investigación.**

***Index Terms*—Internet de las cosas (IoT), Industria 4.0, Seguridad, Privacidad.**

Tipo de contribución: *Investigación original*

I. Introducción

A lo largo de la historia han existido tres grandes hitos en la tecnología. La considerada primer revolución industrial nació con la incursión de las máquinas de vapor en fábricas, que automatizaron algunos de los trabajos de nuestros antepasados. Después de ello con la electricidad, las líneas de ensamblaje dieron lugar a la producción en masa. Luego, la tercera era industrial surgió con la llegada de las computadoras y la incorporación de máquinas y robots en las líneas de ensamblaje. Actualmente, en la llamada Industria 4.0, los sistemas de cómputo programados se encuentran trabajando en conjunto con algoritmos de aprendizaje automático a fin de resolver múltiples tareas [1]. Estos continuamente mejoran su capacidad de controlar tales líneas de ensamblaje con poca interacción de un humano.

La Industria 4.0 ha sido promovida gracias a cuatro elementos muy importantes: i) aumento en la capacidad de cómputo, almacenamiento y conectividad; ii) mejor capacidad de análisis e inteligencia de negocios; iii) nuevas formas de interacción humano-computadora; iv) mejores métodos para transferencia de instrucciones digitales al mundo real, como la robótica y las impresoras 3D [2]. La industria 4.0 se puede resumir como un proceso de fabricación integrado, adaptado, optimizado, orientado al servicio e interoperable, que se correlaciona con algoritmos, *big data* y la más alta tecnología. De aquí se desprende la integración del IoT como una herramienta de la Industria 4.0. Y es que tales tendencias prometen modelos de negocio innovadores y una mejor experiencia al usuario a través de una fuerte conectividad y el uso efectivo de dispositivos embebidos de nueva generación. Estos sistemas generan, procesan e intercambian una inmensa cantidad de datos, muchos de ellos críticos y sensibles. Lo que puede ser considerado como una mina de oro para la ciberdelincuencia.

Los ciberataques en dispositivos IoT se consideran de alto riesgo dado que pueden causar daños físicos y poner en peligro la vida de una persona. Al ser dispositivos que se prevé de alta demanda en la población, los fabricantes buscan la optimización de sus componentes para ofrecer bajos costos y se centran en otorgar la funcionalidad mínima, dejando de lado los requisitos básicos de seguridad. Aunado a ello y tomando en cuenta que muchos de los fabricantes son pequeñas compañías, en caso de existir un ataque, es poco probable que se cuente con actualizaciones de software o la colocación de parches de seguridad para mitigar o prevenir algún daño.

A principios del 2018, en comparación con 2017, se produjo un aumento del 29 % en los ataques distribuidos de denegación de servicio (DDoS). Tal aumento ha sido impulsado por las redes de *bots* IoT [3]. Con IoT, surgen nuevos conjuntos de datos potencialmente confidenciales que están disponibles para hacer perfiles de las cosas o de los dispositivos. En este sentido, las preguntas a responder son: ¿qué dicen los datos recolectados sobre el usuario?, ¿quién puede ver y usar esta información? Ante ello, han aparecido nuevas dimensiones de complejidad en la formulación de conceptos de privacidad, tales como en la ingeniería de políticas de privacidad y la administración de la privacidad de la información. Si bien, múltiples investigaciones han demostrado que la información que los usuarios pueden revelar en los sitios de redes sociales puede tener graves consecuencias. Aun así, los usuarios tienen dificultades para comprender correctamente los posibles efectos a largo plazo de su comportamiento [4]. Esto se traduce a la espera de un escenario donde surjan problemas aún más graves para una aplicación a gran escala de los conceptos de IoT.

El objetivo de este trabajo es analizar las capas de operación de los sistemas IoT, así como también ofrecer un panorama de las soluciones encontradas en la literatura que garantizan la seguridad y privacidad de los datos recolectados. Y finalmente,

identificar los retos aún latentes que ofrezcan una experiencia más segura a los usuarios.

El resto del trabajo se estructura como sigue: En la sección II se analizan los elementos que conforman la arquitectura del IoT. La sección III abarca las vulnerabilidades tanto en seguridad como en privacidad en dispositivos IoT. Posteriormente, en la sección IV, se describen algunos ataques en los dispositivos y capas del IoT, además de algunas medidas de prevención. Por último, en la sección V se exponen las conclusiones.

II. Arquitectura IoT

IoT cuenta con diversos métodos de clasificación, el más utilizado para establecer un marco lógico dentro del IoT son las capas que lo componen. Este método es útil en cuanto a la categorización e identificación del sistema del IoT. Muchos de los trabajos encontrados en la literatura coinciden en que la arquitectura del IoT, está basada en tres capas: percepción, transmisión y aplicación [5] [6]. Por otro lado, algunos otros trabajos toman en cuenta una cuarta capa en la arquitectura del IoT [7]. La cuarta capa es la de cómputo, define varios de los protocolos de comunicación usados para la conectividad de los dispositivos. La clasificación por capas que hemos considerado en este trabajo se puede apreciar en la Fig. 1.

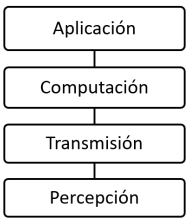


Figura 1. Arquitectura en capas del IoT.

II-1. Capa de percepción: La capa de percepción es la más baja dentro de la arquitectura convencional del IoT. Obtiene datos del entorno, de la persona, del proceso o del mismo sistema sobre el que actúa. Esto es posible brindando los sentidos de visión, el tacto, el olfato, el oído y el pensamiento a objetos inanimados [8]. Algunos sensores de uso común en el IoT se usan para detectar temperatura, peso, movimiento, aceleración y ubicación. Algunas tecnologías emergentes están siendo estandarizadas tomando en cuenta dichas consideraciones [9]. En la Fig.2, se observa los diferentes elementos recopilados de información en el entorno.

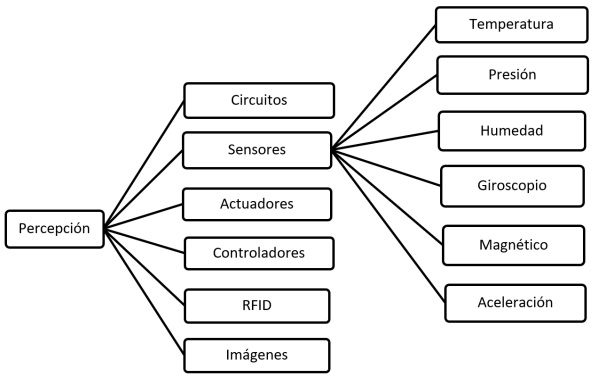


Figura 2. Esquema de la capa de percepción.

Uno de los mayores desafíos es la gestión de los sensores, debido a la complejidad que puedan adoptar los sistemas embebidos. Según la aplicación que tengan o el ambiente en el que se encuentren, se podría recopilar un gran volumen de información, lo que afecta directamente a los dispositivos debido a los recursos limitados que poseen.

II-2. Capa de transmisión: El siguiente paso después de la percepción, es transmitir la información recopilada a las capas superiores. La transmisión está limitada por factores como la potencia, el alcance y la capacidad de almacenamiento [8]. En la Fig. 3, se puede observar la forma en la que se dividen los métodos de transmisión de datos.

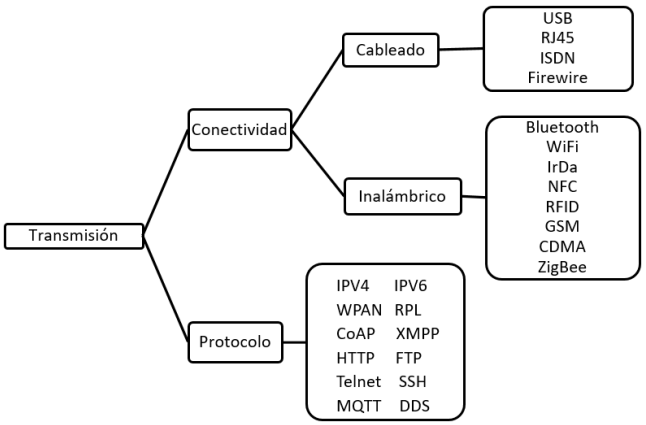


Figura 3. Esquema de la capa de transmisión.

II-3. Capa de cómputo: Se refiere a los medios para recibir y procesar datos, en esta capa se toman y entregan decisiones a la capa de aplicación. La capa de cómputo consiste en hardware, software, algoritmos, computación en la nube, análisis de *big data* y seguridad [10]. También se incluyen los algoritmos necesarios para crear o transformar los datos transmitidos y recopilados. Existen tecnologías de hardware relacionadas con la capa de cómputo que incluyen *SmartThings, arduino, intel Galileo, raspberry pi, beaglebone, cubieboard, smart phone*, entre otras, las cuales fueron desarrolladas para ejecutar aplicaciones de IoT. En la Fig. 4, se puede observar los medios por los cuales se realiza el procesamiento de datos.

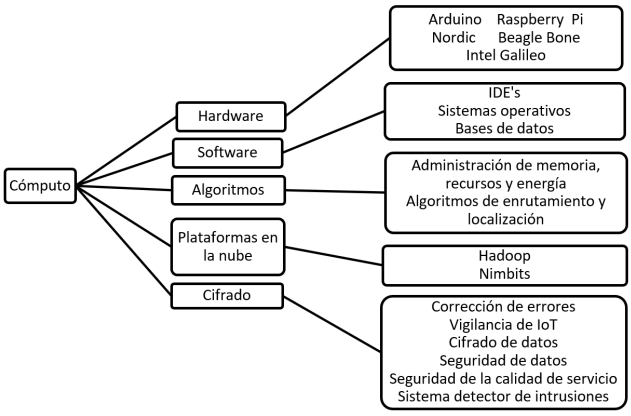


Figura 4. Esquema de la capa de cómputo.

II-4. Capa de aplicación: En esta capa a partir de la información obtenida de las capas inferiores, surgen las aplicaciones para el usuario. Se pueden encontrar múltiples aplicaciones como puede ser, de gestión de recursos en entornos domésticos hasta actividades de tipo industrial. Cualquier aplicación que haga uso de dispositivos conectados corresponde a esta capa [5]. En la Fig.5 se muestran los diferentes tipos de aplicaciones disponibles para el usuario.

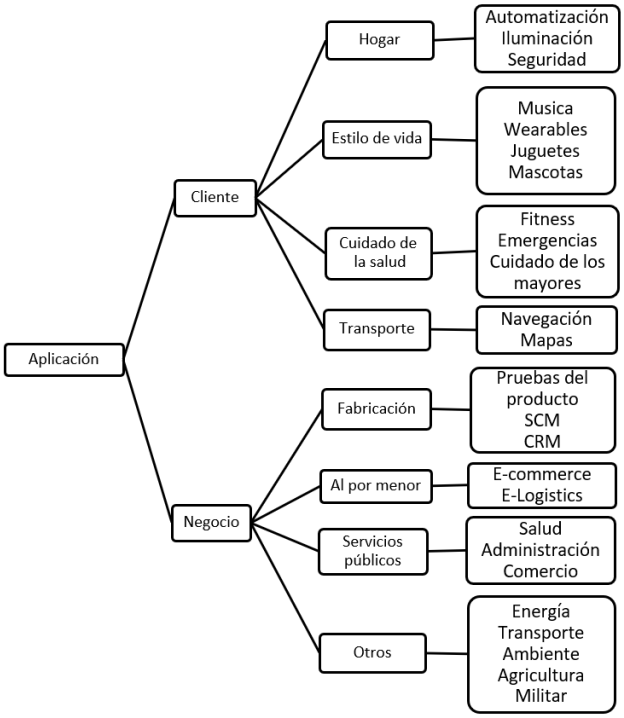


Figura 5. Esquema de la capa de aplicación.

III. Vulnerabilidades en Dispositivos IoT

Es importante definir el término privacidad en una red inteligente [11]. En general, la privacidad en una red inteligente consiste en mantener la información útil de los usuarios fuera del alcance de terceros para evitar la divulgación del perfil de vida del usuario.

En este contexto, los medidores inteligentes son un elemento clave para conocer las actividades de los usuarios al ser un instrumento de comunicación bidireccional entre el usuario y las empresas de servicios públicos [12]. Esto permite el intercambio de datos relacionados con la fijación de precios, así como la información de su consumo. Debido a su naturaleza automatizada y detallada, los datos emitidos desde un medidor inteligente pueden revelar mucho sobre las actividades y el comportamiento del usuario cuando se exponen a técnicas de minería de datos. La información extraída de medidores inteligentes podría ser:

- Detalles del consumo de energía.
- Número de personas dentro o fuera de un edificio.
- Hábitos diarios.
- Patrón de movimiento dentro de un edificio.

Con diversas técnicas de minería de datos se puede conocer las actividades que se desarrollan en un hogar. En la Fig.6 se puede observar el consumo eléctrico extraído de un medidor inteligente. Con esta información es fácil predecir las actividades diarias que una persona realiza en su hogar y con ello realizar un cronograma con el tiempo en el que la persona se encuentra fuera de casa para aprovechar la situación [13].

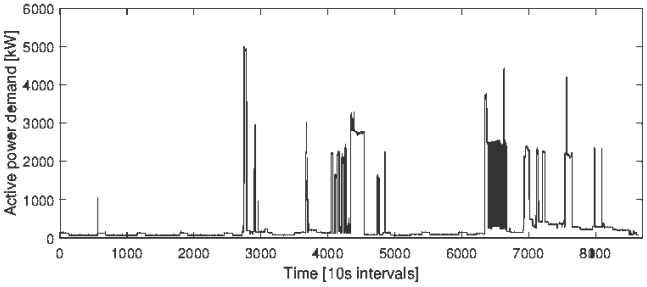


Figura 6. Uso de energía eléctrica extraído de un medidor inteligente.

Se define a la privacidad como el derecho de una entidad, actuando en su propio nombre, para determinar el grado en que interactúa con su entorno, así como el grado en que la entidad está dispuesta a hacerlo [14]. La privacidad debe estar protegida en los dispositivos, en el almacenamiento durante la comunicación y en el procesamiento, lo que ayudaría a evitar la divulgación de la información confidencial [15]. La privacidad de los usuarios y su protección de datos se han identificado como uno de los desafíos más importantes que deben abordarse en dispositivos IoT. A continuación describimos las vulnerabilidades desde sus diferentes enfoques.

III-A. Vulnerabilidades en el Dispositivo

La información confidencial se puede filtrar en caso de manipulación no autorizada o manejo del hardware y software. Por ejemplo, una persona no autorizada al uso del dispositivo puede agregar configuraciones para que un dispositivo envíe información delicada no solo al servidor legítimo, sino también a él mismo o cualquier otro destino. Por lo tanto, para los dispositivos que reúnen datos sensibles, la robustez y la resistencia a la manipulación son especialmente importantes. Para garantizar la seguridad de los dispositivos IoT son necesarias la aplicación de técnicas de validación de integridad de los dispositivos, módulos resistentes a la manipulación y entornos de ejecución confiables.

Con el fin de proporcionar privacidad en los dispositivos, hay muchos problemas que deben abordarse, como podría ser la privacidad de la ubicación del dueño del dispositivo. En este sentido, la ubicación en redes de sensores inalámbricos se puede lograr mediante algoritmos como: *Multi-Routing Random walk* [16], SSRA [17], EEPR [18], entre otros.

III-B. Vulnerabilidades en la Comunicación

Para garantizar la integridad de los datos durante la transmisión de la información, lo más común es aplicar técnicas de cifrado. Sin embargo, debido a que los dispositivos IoT tienen recursos limitados, es complicado aplicarlos a estos dispositivos. Existen iniciativas para el desarrollo de nuevas herramientas dentro de la criptografía, las cuales son algoritmos de cifrado ligero diseñados especialmente para el IoT [19]. Por sí solas, las técnicas de cifrado son insuficientes para

garantizar la integridad de la información. Un correcto enfoque se puede referir a protocolos de comunicación seguros [20].

III-C. Vulnerabilidades en el Almacenamiento

Para proteger la privacidad en el almacenamiento de la información, se pueden considerar los siguientes principios:

- Almacenar la menor cantidad de información necesaria.
- Evitar almacenar información sensible, a no ser que sea estrictamente obligatorio.
- Minimizar el tiempo de almacenamiento de datos.
- Usar un alias para no vincular la información con una identidad real.
- Anonimizar datos.

III-D. Vulnerabilidades en el Procesamiento

La información personal solo debe ser usada cuando sea necesaria, con la aceptación explícita y el conocimiento del propietario de los datos. Sus datos personales no deben divulgarse ni ser retenidos por terceros. Con base en esto, es necesario mantener protegida la información recolectada a través de técnicas criptográficas [19]. Estos métodos aportan ciertas garantías de protección en caso de uso no autorizado de la información recolectada.

IV. Técnicas de Preservación de Seguridad y Privacidad en Dispositivos IoT

El IoT es una tecnología que, a pesar de ser relativamente nueva, ha causado un gran impacto en nuestro entorno. Cada vez se le otorga mayor importancia, esperando que satisfaga todas las expectativas. La compañía de HP, Aruba, realizó un estudio a 3,100 profesionales de 20 países del mundo, incluidos México y Brasil [22]. El informe detalla que muchos de los dispositivos del IoT que se utilizan hoy en día, están inadecuadamente protegidos, dejando a las organizaciones vulnerables a los ataques. Este es un problema inmediato que afecta a las organizaciones actualmente, puesto que un 84 % de estas organizaciones han experimentado una violación de seguridad relacionada con el IoT.

A pesar de las oportunidades que ofrece el IoT, hay muchos riesgos con los que se debe lidiar. Las amenazas y riesgos relacionados con los dispositivos, sistemas y servicios de IoT están creciendo, y los casos de ataques o vulnerabilidades se presentan con mayor frecuencia, ya que en prácticamente todos estos objetos existen vulnerabilidades. Por lo anterior, tanto empresas como organizaciones y usuarios en general, deben de estar preparados para enfrentar y saber mantener una buena postura ante los desafíos que conlleva esta gran tendencia.

Investigadores de la Universidad de Pakistan recolectaron información sobre los ataques más usados en cada una de las capas del IoT, al igual que los desafíos más comunes que se enfrentan en cada una de ellas [23], algunos de estos desafíos se incluyen en la Tabla I.

La protección contra los ataques cibernéticos es uno de los principales retos a considerar en el diseño de los sistemas de IoT. Uno de los primeros ataques exitosos contra sistemas de control industrial fue el gusano Slammer, que infectó dos sistemas de monitoreo críticos de una central nuclear en los EE. UU. en el 2003 [24]. En el mismo año, un virus informático infectó el sistema de control de señal y despacho de una importante red de transporte en los EE. UU. que condujo al paro completo de trenes de pasajeros y de carga [25]. En los años siguientes ocurrieron muchos incidentes afectando significativamente el funcionamiento de organizaciones y sus activos [26].

También, es importante plantear como reto, mantener la disponibilidad de los sistemas, ya que, algún retraso indeseado puede significar grandes pérdidas en la productividad y eficiencia de cualquier organización o empresa.

Otro objetivo fundamental es prevenir cualquier falla del sistema que pueda resultar en daños físicos o que atenten contra la vida de un ser humano. Para lograr este objetivo, se debe preservar la integridad de los sistemas de IoT. Esto incluye la protección contra el sabotaje, que puede provocar una pérdida inadvertida de la calidad del producto. Por lo anterior, es recomendable que se tengan evidencias para demostrar a terceros que la información brindada es integra y de confianza. Mantener la integridad y la confidencialidad de cualquier elemento que utilice el IoT es un aspecto complicado, por lo tanto, siempre se debe de tener un plan estratégico que mantenga un equilibrio entre la seguridad y la disponibilidad. Si llega a ocurrir un ataque, los sistemas afectados deben ser temporalmente deshabilitados y posteriormente restaurados después del ataque. Sin embargo, este concepto es desechado ya que la principal tarea del IoT es la disponibilidad en cualquier momento.

Tabla I
Ataques en las capas de IoT

Capas del IoT	**Vulnerabilidades**
Aplicación	Ataques de código malicioso, software indefenso, ataques de *phishing*.
Cómputo	Seguridad de la aplicación, seguridad de la infraestructura primaria, seguridad de datos en la computación en la nube, amenaza a los recursos compartidos, ataque a máquinas virtuales, relación de terceros.
Transmisión	Ataque de pozo, ataque de hombre en el medio, acceso autorizado RFID, ataque a la puerta de enlace, falsificación de RFID.
Percepción	Inserción de un nodo falso, inserción de código malicioso, ataque de inactividad, interruptor de nodo de red de sensor inalámbrico, ruido en datos.

IV-A. Medidas de Prevención

Existen mecanismos de protección de la seguridad y privacidad en dispositivos IoT. Estas medidas pueden resumirse como se muestra en la Tabla II las cuales están organizadas de acuerdo al nivel de la arquitectura del IoT [23]. Dentro de los mecanismos de protección de la privacidad en dispositivos IoT podemos encontrar esquemas de administración de privacidad que permite al usuario estimar el riesgo de compartir datos privados en medidores inteligentes [27]. También es posible encontrar soluciones basadas en la capa de transmisión de IoT, ya que se necesita autenticar ambos lados para saber que se está comunicando con la parte deseada [28]. Por otro lado, se han desarrollado esquemas donde las transmisiones de datos están protegidas por un protocolo de cifrado simétrico SHS [29].

Tabla II
MEDIDAS DE PREVENCIÓN EN LA ARQUITECTURA IOT

Capas del IoT	Medidas de prevención
Aplicación	Validación de usuario, políticas especiales y permisos, uso de antivirus, *anti-adware* y *antispyware*, uso de firewalls, técnicas de evaluación de riesgos.
Cómputo	Cifrado para asegurar la información clasificada, dispersión de redundancia en la fragmentación de datos, bloqueo hiper seguro, aplicaciones de firewall web.
Transmisión	Confidencialidad de la información, integridad de los datos, enrutamiento seguro.
Percepción	Autenticación de dispositivos, diseño físico seguro de dispositivos finales, arranque seguro, integridad de los datos, anonimato.

En la actualidad las organizaciones enfrentan múltiples ataques contra la privacidad de la información, como en el caso de China en el 2017, el cual fue el país que sufrió más ataques de DDoS posicionándose en el número uno con un 63.30 %, seguido de EE. UU. [30]. Para dar solución a esta problemática lo más común es mejorar la infraestructura de las redes, asegurando la visibilidad completa del tráfico que entra y sale de sus dominios. Es importante observar la cantidad de peticiones de acceso, con el propósito de definir un plan de defensa y conocer el tráfico 'normal' [31]. Esto ayuda a identificar todas aquellas peticiones que representen una amenaza y así actuar de una manera más rápida. Si la organización o empresa cuenta con un plan estructurado, significa que cuenta con la capacidad suficiente para resolver incidentes y dispone de habilidades de mitigación.

A pesar de que el IoT es una tecnología revolucionaria, los dispositivos que se conectan a él son como cualquier otra computadora. Por lo que son igual o más vulnerables [32]. Se ha demostrado que la combinación de intentos de ataque más usados en los últimos años es "telecomadmin/admintelecom" [33]. Por ello, es necesario cambiar las contraseñas con frecuencia y asegurarse de que sean lo suficientemente robustas y fáciles de recordar. Además de comprobar regularmente los parches de seguridad y deshabilitar los dispositivos siempre y cuando no estén en uso. Algunos investigadores recomiendan instalar un *firewall* en la casa o la empresa, para restringir el acceso a usuarios no autorizados. Otro método de protección es el uso de certificaciones, es decir, permitir que los usuarios con certificados de seguridad controlen los dispositivos y a la vez se bloqueen automáticamente los demás perfiles no autorizados.

Los dispositivos conectados al IoT tienen vulnerabilidades que pueden ser explotadas fácilmente Entre dichas vulnerabilidades se encuentran las contraseñas débiles y la falta de cifrado entre las comunicaciones de los dispositivos [28]. Recientemente se han detectado ataques de malware a IoT, como Mirai, el cual es un software que desde que se hizo público su código fuente se han hecho variantes del mismo, haciendo listas de contraseñas y usuarios más largas [33]. Por otro lado, el código ocasiona que los dispositivos cercanos o conectados a la víctima sean infectados igualmente. Los gusanos, *backdoors*, *brickerbot* y *bots* de *spam*, son otro tipo de estrategias usadas para obtener información sensible. Por lo que se recomienda que se ejecuten periódicamente una exploración de puertos en todos los equipos, instalar *firewalls* y mantener todos los dispositivos actualizados. Sin embargo, a pesar de que se cuente con todas las medidas de protección posible, nunca se llega a una seguridad total. Por lo que es necesario contar con estrategias definidas para la reducción del riesgo, tomando acciones por adelantado. Adicionalmente, si es posible, se deben agregar planes de contingencia basados en acciones puestas en marcha sólo si las señales de advertencia se disparan. No se trata de cambiar la probabilidad o el impacto del riesgo, pero sí planificar como controlarlo en caso de que ocurra.

IV-B. Retos aún latentes

IV-B1. Satisfacer las expectativas de los clientes: Gracias a la disponibilidad del Internet como servicio, desde los años 90's a la fecha, se ha incrementado la forma en que las personas realizan sus compras, escuchan música, buscan un domicilio, piden un taxi, entre otras actividades. También ha cambiado el enfoque del cliente de tener productos estandarizados producidos en serie, a pasar a los servicios y productos personalizados. Hoy en día, las expectativas de los clientes son más altas [34].

IV-B2. Actualizaciones de software: Mantener los sensores de IoT correctamente calibrados es esencial para su funcionamiento, como el de cualquier otro tipo de sensor eléctrico. Los sensores de nuevas generaciones están integrados en numerosos dispositivos lo que hará difícil de sincronizar el flujo de datos de todo el hardware sin ayuda de un equipo profesional. También es importante considerar que todos los dispositivos cuenten con un software actualizado, con el fin de prevenir ataques y sean detectados y mitigados por las organizaciones que distribuyen dichos equipos.

IV-B3. Conectividad: En su forma actual IoT utiliza un modelo centralizado cliente-servidor para establecer conectividad en servidores, estaciones de trabajo y sistemas. Por el momento, dicho modelo es eficiente. De acuerdo a algunos reportes, se espera que más de 20 mil millones de dispositivos se conecten al IoT para el 2020 [35]. Es solo cuestión de tiempo para que los usuarios de IoT tengan afectaciones significativas por la gran cantidad peticiones y la capacidad de respuesta limitada a las mismas.

IV-B4. Regulaciones: Puesto que el IoT, la nube e incluso el Internet no están ligados a una ciudad, estado o alguna región específica, ¿quién es el encargado de establecer regulaciones? Existen personas que creen que los gobiernos deberían establecer estas regulaciones, ¿pero cómo hacerlo cuando las leyes son diferentes entre los países? La IEEE, menciona algunas de las propuestas de los defensores de las regulaciones gubernamentales [36]. Entre ellas se destaca la importancia de la privacidad en estas regulaciones.

IV-B5. Inteligencia Artificial: La amenaza de *ransomware* y otro tipo de ataques han crecido 35 veces el último año. Por lo que se puede considerar que los problemas para los proveedores de la nube solo están comenzando [37] [38]. Es muy probable que el siguiente objetivo del *ransomware* sean los proveedores de servicio de la nube. La caída de estos servicios pueden producir un punto de falla único para cientos

de empresas, entidades gubernamentales y organizaciones de atención médica. Se prevé un incremento de *malware* creado completamente por máquinas basado en la detección de vulnerabilidades automatizada y el análisis de datos complejos aprovechado por la inteligencia artificial para crear código nuevo que sea capaz de evadir su detección.

V. Conclusiones

La rápida tasa de adopción del IoT ha llevado a una cantidad sin precedentes de datos recopilados sobre dispositivos que se convirtieron en parte de la plataforma de IoT. Esta inmensa recopilación de datos ha dado lugar a problemas de privacidad y a la pérdida de control sobre la información personal recolectada por los proveedores de servicios. Al observar el funcionamiento del IoT, sus ataques y retos de seguridad y privacidad, es importante establecer que la responsabilidad es tanto del usuario como de los proveedores del dispositivo. Sería deseable que cada dispositivo de IoT debería incluir un software actualizado e incorporar la capacidad de llevar a cabo actualizaciones periódicas a medida que se encuentren nuevas vulnerabilidades. De igual manera, cualquier persona que instale o habilite un dispositivo IoT debería cambiar la configuración predeterminada de usuario/contraseña y permanecer atento a cualquier actividad sospechosa de la red. Sin duda, aún existen muchos retos por vencer y el IoT no va a desaparecer, así como tampoco sus ataques. Con un poco de cuidado durante su configuración y un monitoreo constante en la red, se pueden evitar violaciones de seguridad.

Referencias

[1] Lasi H., Fettke P., Kemper H.G. "Industry 4.0". Springer Fachmedien Wiesbaden:En Business & Information Systems Engineering, Vol. 6, Issue 4, pp. 239–242, August 2014.

[2] Roblek V., Meško M., Krapež A. "A complex view of industry 4.0 ", Sage Open. Los Angeles, CA: SAGE Publications Sage CA, April 2016.

[3] Prachi S.D., Subhash C.S., Sateesh K.P. "A Network-Based Intrusion Detection System". Springer Singapore:Security and Data Storage Aspect in Cloud Computing, Vol 52, pp. 35-48, 2019.

[4] Acquisti A., Grossklags J. "Privacy Attitudes and Privacy Behaviour". Springer:Economics of Information Security, Vol. 12, pp. 165–178, 2004.

[5] Rafiullah K., Sarmad U.K., Rifaqat Z., Shahid K. "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges". IEEE:2012 10th International Conference on Frontiers of Information Technology, 2012.

[6] Mohammed R.A., Djamel T., Imed R. "Architecting the Internet of Things: State of the Art". Springer, Cham: Robots and Sensor Clouds, Vol 36, pp. 55-75, August 2015.

[7] Muhammad Bilal. "A Review of Internet of Things Architecture, Technologies and Analysis Smartphone-based Attacks Against 3D printers". Cornell University (CU), the Simons Foundation, and a global collective of institutional members institutions, June 2017.

[8] Amy J.C. Trappey, Charles V. Trappey, Usharani Hareesh Govindarajan, Allen C.C. Jhuang, John J.H Sun. "Advanced Engineering Informatics". England, UK: Elsevier Science, December 2016.

[9] Víctor B.S. "Internet de las cosas. Horizonte 2050". Instituto Español de Estudios Estratégicos, Julio 2018.

[10] Manuel D., Cristian M., Bartolomé R. "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing". Elsevier:Journal of Network and Computer Applications, Vol. 67, pp. 99-117, May 2016.

[11] Costas E., Georgios K. "Smart Grid Privacy via Anonymization of Smart Metering Data". IEEE:2010 First IEEE International Conference on Smart Grid Communications, November 2010.

[12] Georgios K., Costas E., Stojan Z.D., Tim A.L, Rafael C.. "Privacy for Smart Meters: Towards Undetectable Appliance Load Signatures". IEEE: 2010 First IEEE International Conference on Smart Grid Communications, October 2010.

[13] Michael D., Barry P.H. "Non-Intrusive Load Monitoring using Electricity Smart Meter Data: A Deep Learning Approach". ResearchGate:University College Cork, November 2018.

[14] R. Shirey. Internet security and privacy". BBN Technologies, Network Working Group, May 2000.

[15] Samiksha R.S. "A Study on Privacy and Security concerns in Internet of Things". International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 9, September 2016.

[16] Liming Z., Qiaoyan W,. Hua Z.. "Preserving Sensor Location Privacy in Internet of Things", IEEE:2012 Fourth International Conference on Computational and Information Sciences, pp. 856-859, September 2012.

[17] Sofiane H., Jaime L., Pascal L.. "Smart and Self-Organized Routing Algorithm for Efficient IoT communications in Smart Cities", IET Wireless Sensor Systems, October 2018.

[18] Sang H.P., Seungryong C. and Jung R. L., "Energy-Efficient Probabilistic Routing Algorithm for Internet of Things", Hindawi Publishing Corporation, Journal of Applied Mathematics, Volume 2014, April 2014.

[19] Saurabh S., Pradip K. S., Seo Y. M., Jong H. P. "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions". Springer Berlin Heidelberg:Journal of Ambient Intelligence and Humanized Computing, pp. 1–18, May 2017.

[20] Michalis G., Korina K., Nikos F., Giannis F.M., George C.P. "Towards Secure and Context-Aware InformationLookup for the Internet of Things". IEEE International Conference on Computing, Networking and Communications (ICNC), January 2013.

[21] Liu E., Liu Z., Shao F. "Digital Rights Management and Access Control in Multimedia Social Networks". Springer, Cham:Genetic and Evolutionary Computing, Vol 238, pp. 257-266, 2014.

[22] Aruba HP. "El internet de las cosas IoT". [Online] Available:https://www.arubanetworks.com/es/partners/partners-de-ecosistema/el-internet-de-las-cosas-iot/

[23] Tariq A.R., Ehsan-ul-Haq. "Security Challenges Facing IoT Layers and its Protective Measures". University of Pakistan Lahore, Pakistan, Vol. 179, Issue. 27, March 2018.

[24] Kevin P., SecurityFocus. "Slammer worm crashed Ohio nuke plant network". SecurityFocus, August 2003.

[25] David H.. "Virus Disrupts Train Signals". CBS News, August 2003.

[26] Ahmad-Reza S., Christian W., Michael W. "Security and Privacy Challenges in Industrial Internet of Things". IEEE:2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), July 2015.

[27] Arijit U., Soma B., Arpan P. "IoT-Privacy: To be private or not to be private". IEEE: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 123-124, July 2014.

[28] Gergely A., Lejla B., Lynn B., Veelasha M., Anna K., Antoine G., Iynkaran N. "New Directions in IoT Privacy Using Attribute-Based". Radboud Repository of the Radboud University Nijmegen, pp. 463-464, May 2016.

[29] Tianyi S., Ruinian L., Bo M., Jiguo Y., Xiaoshuang X., Xiuzhen C. "A Privacy Preserving Communication Protocol for IoT Applications in Smart Homes". IEEE:IEEE Internet of Things Journal, Vol. 4, Issue. 6, pp. 1844-1852, December 2017.

[30] Alexander K., Oleg K., Kirill I. "DDoS attacks in Q3 2017". SecureList, November 2017.

[31] "Industries most frequently targeted by denial of service (DDoS) attacks worldwide as of 4th quarter". Statista, 2017.

[32] Ebraheim A., Abdallah T.. "Internet of Things: Features, Challenges, and Vulnerabilities". International Journal of Advanced Computer Science and Information Technology (IJACSIT). Vol. 4, Issue. 1, pp. 1-13, 2015.

[33] Constantinos K., Georgios K., Angelos S., Jeffrey V.. "DDoS in the IoT: Mirai and Other Botnets". IEEE:Browse Journals & Magazines, Vol. 50, Issue 7, pp 80-84, 2017.

[34] Vala A. "New Research Uncovers Big Shifts In Costumer Expectations And Trust". Salesforce, Jun 2018.

[35] Muhammad A., Peter T., Shahid M., Jonathan R. "Context-aware cooperative testbed for energy analysis in beyond 4G networks". Springer:Telecommunication Systems, Vol. 64, Issue 2, pp 225–244, February 2017.

[36] IEEE. "Should the Government Regulate IoT Devices?". IEEE Inovation at work, 2018.

[37] Ibrar Y., Ejaz A., Muhammad H., Abdelmuttlib I., Abdalla A., Mohammed A., Muhammad., Mohsen G. "The rise of ransomware and emerging security challenges in the Internet of Things". Elsevier:Computer Networks, Vol. 129, Part 2, pp. 444-458, December 2017.

[38] Amin A., Ali D., Mauro C., Kim-Kwang R. "Detecting crypto-ransomware in IoT networks based on energy consumption footprint". Springer Berlin Heidelberg:Journal of Ambient Intelligence and Humanized Computing, Vol. 9, Issue 4, pp. 1141–1152, August 2018.

A review of Behavioral Biometric Authentication in Android Unlock Patterns through Machine Learning

Jose Torres*, Marcos Arjona*, Sergio de los Santos*, Efthimios Alepis† and Constantinos Patsakis†
*ElevenPaths. Telefónica Digital España, Málaga, Spain. Email: {jose.torres, marcos.arjona, ssantos}@11paths.com
‡ Department of Informatics, University of Piraeus, Piraeus, Greece. Email: {talepis, kpatsak}@unipi.gr

***Resumen*—Due to the ever-increasing deployment of services for which users need to authenticate, many applications require higher standards of security, such as drawn patterns and fingerprints, used mostly to authenticate users and unlock their smart devices. In this work we propose a biometrics-based machine learning approach that supports user authentication in Android to augment native user authentication mechanisms, making the process more seamless and secure. Our evaluation results show very high rates of success, both for authenticating the legitimate user and for rejecting an adversary who imitates the legitimate user. Finally, we showcase how the proposed solution can be securely deployed in non-rooted Android devices.**

***Index Terms*—Cybersecurity, Machine Learning, Artificial Intelligence, Android**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

This paper focuses on providing a robust solution that utilizes user behavioural biometrics through Machine Learning to improve smartphone locking solutions. This approach can be adapted to a variety of other use cases, such as smartphone app and resource locking and two-factor authentication. The presented evaluation study not only provides proof about the significance of our proposed approach but also gives evidence about the protection levels of specific lock patterns that are frequently used by users. Moreover, contrary to the related work, we detail how the proposed mechanism can be deployed in stock Android devices without the need to root the device.

II. Proposed Solution

This solution implements a specific use case of our online service, also called "*SmartPattern*", which works as an authorization/authentication mechanism of a service that allows protecting any external resource through "Smart Patterns" (using an API, Oauth2 or JWT). At present, the app focuses on securely locking specific apps inside the Android ecosystem. Clearly, the OS itself could adopt the described underlying approach and use it for locking the Android devices more securely.

All resources inside the Android OS are handled through corresponding OS apps or third-party apps. Our approach utilizes the provided by the OS `UsageStatsManager` API that we have exploited to successfully recognize the foreground app, contrary to Android documentation that states that this API is used only for "*Usage data that is aggregated into time intervals: days, weeks, months, and years*". In our approach, we check for newly created lists of `UsageStats` in regular time intervals, taking only the latest `INTERVAL_DAILY` records into consideration. Then, looping through the retrieved list, the most recent record in terms of the timestamp is kept which corresponds to the foreground app's package name.

Then, our app starts working in the background, as a service, silently monitoring each launched app. Whenever the service recognizes a launched app that is selected by the user to be securely locked, our app presents a full-screen Android activity, "hiding" it, and thus protecting the app in question. This "protecting" screen can only be bypassed if the user authenticates himself, through our novel smart pattern mechanism. The pattern mechanism not only checks whether the correct pattern is being drawn, but also whether the user who is drawing the pattern can be authenticated from his behaviour while drawing the pattern, through the underlying machine learning core module.

The secure screen provided by our app cannot be bypassed, since even when it is minimized or closed, the running service will continuously keep on re-launching it, having detected a "protected" app in the foreground. A possible attack on our app could be an attempt to uninstall it. This can also be easily protected, by additionally monitoring the "settings" of the device through our application, denying access to the app uninstallation panel. This work does not focus only to the app being the final "product", but to the underlying approach, which has been made to further secure smartphones and sensitive resources from malicious users. As a result, the proposed solution could be used in an increasing number of even more use cases, such as two-factor authentication with increased levels of security and should be considered as of great importance in terms of its scientific contribution.

III. Machine Learning Core

To work with drawn patterns, we need to initially establish an encoding of the associated data [1]. To encode the information provided by the unlock pattern, we have established an enriched pattern path (pattern + times) as an array of vectors (one by each existing point in the pattern). With this definition, it should be possible to generate a model for each user to predict whether each one of the future patterns introduced in the application actually belongs to the "genuine" user. Therefore, and after checking the shape of the pattern, the next step focuses mainly on exact "timing" extraction of the drawn pattern, generating a feature vector composed by each point's timestamp representation in the drawn pattern.

Using this known and valid pattern features vectors, it is then possible to generate a dataset to build a ML model, able to determine whether an entered pattern belongs to the legitimate user or not. In this case, it seems not straightforward to implement a supervised strategy, mainly because it is not possible to learn incorrect pattern inputs. However, it

is possible to use some One Class Supervised algorithms (such as One Class Support Vector Machine, SVM) or other unsupervised ML approaches like clustering. In our case and for the purposes of this study, we have used both, namely a One Class SVM and also the Clustering approach through an implementation of the K-means algorithm.

IV. Evaluation Experiment

In this section, we describe the settings of the experiment that has been conducted to evaluate the effectiveness of the resulting app. The experiment involved 64 users, as well as 6 supervisors in a total of 70 participants, including 2 phases of evaluation. The first phase involved 54 users and the 6 supervisors, while the second phase involved 10 users and the same supervisors of phase 1.

We should consider different types of user patterns both in terms of complexity that translates into the number of points the users' use and also in terms of complexity that translates in differentiations while drawing the actual pattern, namely quick or slow finger movements and also intended "strategic" pauses of the user in specific parts of the pattern. Our experiment revealed that these criteria are quite significant since both the number of points of the patterns and also the timings involved affect both the effectiveness of the underlying algorithm to reject the false users, but also the system's effectiveness in minimizing false positives of real users. Indeed, as expected, a smart pattern becomes more effective in terms of security as the number of points increases, while users' atomic timings while drawing their patterns is of equal or even greater importance in terms of app safeguarding.

The collected results were merged and analyzed to produce the core of our evaluation experiment. Each user in the first phase made in total 50 attempts to successfully pass each one of the five smart pattern challenges. As for the evaluation results, the 54 evaluation users of the first phase made 2700 attempts in total. After the analysis of the results, from the total of 2700 user attempts, 2530 of them were unsuccessful, meaning that our proposed system had 93.7 % success in denying access to unauthorized users. Respectively, in the second phase of the experiment, 10 users made 10 attempts to unlock the app using their own personal trained model of the smart pattern app. In total, they made 100 attempts, where 86 of them have been successful, which means 86 % in successfully "recognizing" the app's genuine user and consequently accepting his/her access to the app.

For each basic pattern attempt for the attacker, the SmartPattern app blocked the attempts illustrated in percentages in Figure 1. We may easily note that the percentage of the SmartPattern app's security increases, the more "complex" the drawn pattern is. This result might seem expected, however, in our study the complexity level could be translated not only in terms of "more points" but also in terms of "timing pauses", meaning that if the real user had a more "unique" way of drawing the pattern, then this would increase its security level. Another very interesting observation deriving from our study is that malicious users having knowledge only for the "final drawing" and not of the way, in terms of consequent points, that it was drawn did not have much success in "guessing" the correct way of unlocking the pattern. Finally, another significant result of the study is that in the

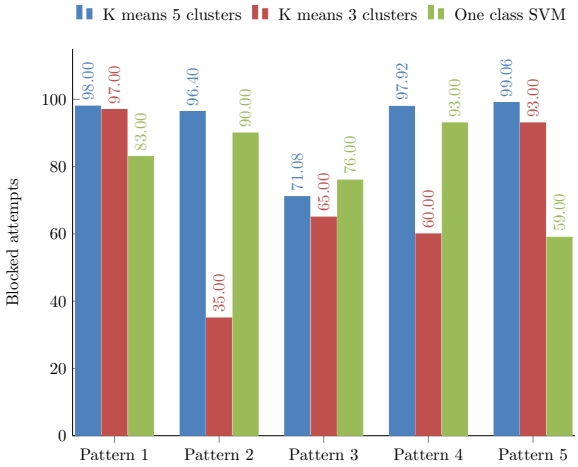


Figura 1. Attempts blocked by SmartPattern when other users try to unlock the mobile phone.

cases of the more complex drawn patterns and consequently their more complex involved timing biometrics, even when the "attackers" had more "help" by watching the pattern being drawn more times in videos, their unlock attempts where still unsuccessful in their vast majority. These results have been quite encouraging, indicating that our research pointed in a good direction, actually providing improvements in the smartphones' unlocking mechanisms.

V. Conclusions

In this paper, a novel locking mechanism for Android smartphones that utilizes Machine Learning and user biometric data has been presented. Our approach can be incorporated in many application domains, ranging from providing more security to mobile devices when locking them, to specifically securing targeted "sensitive" resources and also improving 2FA without the need to root the device. The evaluation results have shown a 93.7 % success rate in denying access to unauthorized users and a 86 % success rate in allowing access to the real users of the app. This little "lower" success percentage of the resulting app can be justified since it was the authors' primary objective, when calibrating the ML backend, to maximize user protection, even if this resulted in lower "user friendliness" level.

Our evaluation experiments have provided us with strong evidence that our approach is very successful in "distinguishing" genuine and malicious users through the way they draw lock screen patterns. The paper's results also provide the scientific literature with valuable evidence about the efficiency of common patterns in securely protecting the smartphones.

Acknowledgments

This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the OPERANDO project (Grant Agreement no. 653704) and the University of Piraeus Research Center

Referencias

[1] J. Torres, S. de los Santos, E. Alepis and C. Patsakis: "Behavioral Biometric Authentication in Android Unlock Patterns through Machine Learning ", at *The International Conference on Information Systems Security and Privacy*, 2019.

Formal verification of the YubiKey and YubiHSM APIs in Maude-NPA

Antonio González-Burgueño
University of Oslo, Norway
antonigo@ifi.uio.no

Damián Aparicio-Sánchez
Universitat Politècnica de València, Spain
daapsnc@dsic.upv.es

Santiago Escobar
Universitat Politècnica de València, Spain
sescobar@dsic.upv.es

Catherine Meadows
Naval Research Laboratory, Washington DC, USA
meadows@itd.nrl.navy.mil

José Meseguer
University of Illinois at Urbana-Champaign, USA
meseguer@illinois.edu

***Abstract*—We have performed in [1] an automated analysis of two devices developed by Yubico: *YubiKey*, designed to authenticate a user to network-based services, and *YubiHSM*, Yubico's hardware security module. Both are analyzed using the Maude-NPA cryptographic protocol analyzer. YubiKey & YubiHSM are cryptographic Application Programmer Interfaces, involving a number of complex features: (i) discrete time in the form of *Lamport clocks*, (ii) a mutable memory for storing previously seen keys or nonces, (iii) event-based properties that require an analysis of sequences of actions, and (iv) reasoning modulo exclusive-or. , we were able to automatically prove security properties of YubiKey and find the known attacks on the YubiHSM.**

***Index Terms*—Encryption, event lists, Exclusive-OR, Lamport clocks, mutable memory, protocol verification, YubiHSM, YubiKey**

I. Introduction

Nowadays there exist several security tokens having the form of a smartcard or an USB device, which are designed for protecting cryptographic values from an intruder, e.g, hosting service, email, e-commerce, online banks, etc. They are also used to ease authentication for the authorized users of a service, e.g., if you are using a service that verifies your Personal Identification Number (PIN), the same service should not be used for checking your flights, reading your emails, etc. By using an Application Programming Interface (API) to separate the service from the authentication system, such problems can be prevented.

Yubico is a leading company on open authentication standards and has developed two core inventions: the *YubiKey*, a small USB designed to authenticate a user against network-based services, and the *YubiHSM*, Yubico's hardware security module (HSM). The increasing success has led to its use by governments, universities and companies like Google, Facebook, Dropbox,CERN, Bank of America etc., including more than 30,000 customers [2].

II. Devices

The YubiKey allows for the secure authentication of a user against network-based services by considering different methods: one-time password (OTP), public key encryption, public key authentication, and the Universal 2nd Factor (U2F) protocol [3]. An important feature of YubiKey is that it is independent of the operating system and does not require any installation, because it works with the USB system drivers. We will focus on the YubiKey OTP mode, a mode that uses a button physically located on the YubiKey. When this button is pressed, it emits a string that can be verified only once against a server in order to receive the permission to access a service. The authentication protocol of YubiKey involves three roles: (i) the user, (ii) the service, and (iii) the verification server. The user can have access to the service if it provides its own valid OTP generated by the YubiKey.

YubiHSM is intended to operate in conjunction with a host application. It supports several modes of operation, but the key concept is a symmetric scheme where one device at one location can generate a secure data element in a secure environment. The YubiHSM is designed to protect the YubiKey AES keys when an authentication server is compromised by encrypting the AES keys using a master key stored inside the YubiHSM.

In [4], [5], Künnemann and Steel reported two kinds of attacks on version 0.9.8 beta of YubiHSM API: (a) if the intruder has access to the server running YubiKey, where AES keys are generated, then it is able to obtain plaintext in the clear; (b) even if the intruder has no access to the server running YubiKey, it can use previous nonces to obtain AES keys. However, they were only able to find the first attack in Tamarin [6] due to the limited support for exclusive-or in Tamarin at that time.

The first attack involved the YubiHSM API command *AES_ECB_BLOCK_ENCRYPT*. This command takes a handle to an AES key and the nonce and applies the raw block cipher. In order to perform this attack the intruder compromises the server to learn an *AEAD* and the key-handle used to produce it. Then, using the block encrypt command an intruder is able to decrypt an *AEAD* by recreating the blocks of the key-stream: inputting $counter_i$ (the nonce) to the YubiHSM Block Encrypt API command. The intruder exclusive-ors the result with the *AEAD* truncated by the length of the *MAC* and obtains the plaintext. Note that the verification of this attack in [4] using Tamarin involved additional user-defined lemmas (see [1]).

The second attack involved the YubiHSM command *AEAD-GENERATE*. Thist tattack takes a nonce, a handle to an AES key and some data and outputs an *AEAD*. An intruder can produce an *AEAD* for the same handle *kh* and a value *nonce* that was previously used to generated another *AEAD*.

An intruder can recover the keystream directly by using the *AEAD-GENERATE* command to encrypt a string of zeros and then discarding the *MAC*. The result will be the exclusive-or of the keystream with a string of zeros, which is equal to the keystream itself. This attack is worse than the first one, because this command cannot be avoided (see [5]).

However, there has not been any completely automated analysis of these two attacks to date because both YubiKey and YubiHSM involve a number of complex features: (1) discrete time in the form of *Lamport clocks*, (2) a mutable memory for storing previously seen keys or nonces, (3) event-based properties that require an analysis of sequences of actions, and (4) reasoning modulo exclusive-or. Maude-NPA [7] has provided support for exclusive-or for years but has not provided support for the other three features, which can be supported by using constraints on natural numbers, protocol composition and reasoning modulo associativity.

III. Results

Cryptographic Application Programmer Interfaces (Crypto APIs) have been subjected to intruder manipulation to disclose relevant information. We automatically prove the properties (a,b,c) below of YubiKey and (d,e) below on YubiHSM:

(a) Absence of replay attacks in YubiKey, i.e., there are no two distinct logins that accept the same counter.
(b) Correspondence between pressing the button on a YubiKey and a successful login. In other words, a successful login must have been preceded by a button pressed for the same counter.
(c) Counter values of YubiKey are different over time, where a successful login invalidates previous *OTP*s.
(d) If the intruder has access to the server running YubiKey, it can use previous YubiHSM nonces to obtain AES keys.
(e) If the intruder has no access to the server running YubiKey, it can use previous YubiHSM nonces to decrypt a previously generated AEAD.

This paper is the third in a series using Maude-NPA to analyze cryptographic APIs. We find this problem area one of particular interest for two reasons. First, these APIs often use exclusive-or and this gives us the opportunity to explore how well Maude-NPA can be applied to protocols that use exclusive-or. Secondly, cryptographic APIs offer a number of other challenging features and this allows us to explore how Maude-NPA can handle them.

IV. Conclusions

The main contributions of this paper are to both prove properties of YubiKey generation 2 and find the known attacks on version 0.9.8 of YubiHSM in a completely automated way beyond the capabilities of previous work in the literature. This allowed us to perform the analysis of these APIs in a fully-unbounded session model making no abstraction or approximation of fresh values, and with no extra assumptions. These APIs involve several challenges: (1) handling of Lamport clocks, (2) modeling of mutable memory, (3) handling of constraints on the ordering of events, and (4) support for symbolic reasoning modulo exclusive or.

The main goal of this work has been to investigate whether Maude-NPA could complement and extend the formal modeling and analysis results about YubiKey and YubiHSM obtained in [5]. This is a non-obvious question: on the one hand, Maude-NPA has provided support for exclusive-or for years, so it is well-suited for meeting Challenge (4). But, on the other hand, previous applications of Maude-NPA have not addressed Challenges (1)-(3). The main upshot of the results we present can be summarized as follows: Challenge (2) can by met by expressing mutable memory in terms of synchronization messages, a notion used in Maude-NPA to specify protocol compositions [8]. Challenge (3) can by met by the recently added unification modulo associativity, allowing an easy treatment of lists. Finally Challenge (1) can be met by a slight extension of Maude-NPA's current support for equality and disequality constraints [9], namely, by adding also support for constraints in Presburger Arithmetic. We show how challenges (1)-(4) can all be met by Maude-NPA, and how these results in automated formal analyses of YubiKey and YubiHSM that substantially extend previous analyses. Finally, a key point of our analysis is that very few tools. are well equipped to simultaneously handle all of the challenges.

Acknowledgements: Partially supported by the EU (FEDER) and the Spanish MCIU under grant RTI2018-094403-B-C32, by the Spanish Generalitat Valenciana under grant PROMETEO/2019/098, and by the US Air Force Office of Scientific Research under award FA9550-17-1-0286.

References

[1] A. González-Burgueno, D. Aparicio-Sánchez, S. Escobar, C. Meadows, and J. Meseguer, "Formal verification of the YubiKey and YubiHSM APIs in Maude-NPA," in *LPAR-22. 22nd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, ser. EPiC Series in Computing, G. Barthe, G. Sutcliffe, and M. Veanes, Eds., vol. 57. EasyChair, 2018, pp. 400–417. [Online]. Available: https://easychair.org/publications/paper/qkkq

[2] "Yubico customer list." Available on: http://www.yubico.com/references. [Online]. Available: https://www.yubico.com

[3] "Specifications Overview, FIDO Alliance," Available on: https://fidoalliance.org/specifications/overview/, Dec. 2015. [Online]. Available: https://fidoalliance.org/specifications/overview/

[4] R. Künnemann and G. Steel, "YubiSecure? formal security analysis results for the Yubikey and YubiHSM," in *Revised Selected Papers of the 8th Workshop on Security and Trust Management (STM'12)*, ser. Lecture Notes in Computer Science, A. Jøsang, P. Samarati, and M. Petrocchi, Eds., vol. 7783. Pisa, Italy: Springer, Sep. 2012, pp. 257–272. [Online]. Available: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/KS-stm12.pdf

[5] R. Künnemann, "Foundations for analyzing security APIs in the symbolic and computational model. Available on: https://tel.archives-ouvertes.fr/tel-00942459/file/Kunnemann2014.pdf," Theses, École normale supérieure de Cachan - ENS Cachan, Jan. 2014. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00942459

[6] "The Tamarin-Prover Manual (September 25, 2018)," Available on: https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf. [Online]. Available: https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf

[7] "Maude-NPA Manual v3.1.1," Available on: http://maude.cs.illinois.edu/w/images/d/d5/Maude-NPA_manual_v3.pdf. [Online]. Available: http://maude.cs.illinois.edu/w/images/9/90/Maude-NPA_manual_v3_1.pdf

[8] S. Santiago, S. Escobar, C. A. Meadows, and J. Meseguer, "Effective sequential protocol composition in Maude-NPA," *CoRR*, vol. abs/1603.00087, 2016. [Online]. Available: http://arxiv.org/abs/1603.00087

[9] S. Escobar, C. A. Meadows, J. Meseguer, and S. Santiago, "Symbolic protocol analysis with disequality constraints modulo equational theories," in *Programming Languages with Applications to Biology and Security - Essays Dedicated to Pierpaolo Degano on the Occasion of His 65th Birthday*, ser. Lecture Notes in Computer Science, C. Bodei, G. L. Ferrari, and C. Priami, Eds., vol. 9465. Springer, 2015, pp. 238–261. [Online]. Available: https://doi.org/10.1007/978-3-319-25527-9_16

A review of Message Anonymity on Predictable Opportunistic Networks

D. Chen
U. Autonoma de Barcelona.
Cybercat
depeng.chen@deic.uab.cat

G. Navarro-Arribas
U. Autonoma de Barcelona.
Cybercat
joan.borrell@uab.cat

C. Perez-Sola
U. Oberta de Catalunya.
Cybercat
cperez@uoc.edu

J. Borrell
U. Autonoma de Barcelona.
Cybercat
joan.borrell@uab.cat

***Abstract*—We review the use of simple onion routing for message anonymity in deterministic opportunistic networks. We provide stochastic onion routing algorithms and anonymity measures for such scenarios.**

***Index Terms*—Anonymity, Opportunistic networks, onion routing.**

Tipo de contribución: *Investigación ya publicada*

I. Introduction

Opportunistic Networks (OppNets) are networks where communication happens opportunistically between nodes, end-to-end connectivity is not guaranteed and disruptions and delays are expected [1]. They are also denoted as Delay and Disruption Tolerant Networks (DTN) [2]. Among OppNets there are those where the contacts between the nodes follow an specific pattern [3], so the behaviour of the network can be predicted to some extend. Those are denoted as networks are denoted as Predictable OppNets (POppNets).

We in the specific POppNet that raises from a network build on public transportation systems. E.g. all public buses in a city carry a simple network node allowing them to opportunistically exchange messages. Given the routes and timetables of the buses one can predict when interactions between nodes will occur during the day, and thus these networks can be used as a low cost urban networks. Routing can be more efficiently solved in POppNets than in generic OppNets due to their predictability, and we believe that some security services can also be improved. More precisely, anonymous routing is a difficult and complex problem in OppNets. Current solutions to provide anonymous routing in OppNets require complex cryptographic solutions, and complex set ups, given that typical onion routing [4] cannot be directly applied. We will show that the predictability in these networks can be exploited to actually use a simplified onion routing approach to provide message anonymous routing in POppNets.

The aim is to support applications where one end needs to send an anonymous short message in one direction. We assume that we can directly use the nodes public keys to perform a simple onion routing since establishing a session key incurs in more penalty than gain. Although these might seem strong assumptions, they are commonly used in OppNet environments, applications, and protocols such as Bundle protocol [2]. The expected delivery time is also relatively large, which allows for more randomness in the path selection.

This is a summary of a paper from the same authors submitted to the Journal of Ambient Intelligence and Humanized Computing (currently undergoing a minor revision). Here we attempt to summary the most relevant aspects of the publication. Most technical details are omitted due to lack of space and in favor of clarity.

II. Paths for onion routing in POppNets

We model a POppNet as an undirected dynamic graph $G(V, E)$. An node is a network node, and an edge represents the fact that two nodes can communicate in both directions. $e = (u, v, t, \lambda)$, is a *temporal edge* between nodes u and v, starting at time t, with a duration of λ. $P = \langle (v_1, t_1), (v_2, t_2), \ldots, (v_l, t_l) \rangle$ denotes a *path* in the dynamic graph $G(V, E)$, where (v_i, t_i) represents the node in the path and the time that the message arrives to such node. The *length* of the path is the number of node, and the *duration* is the time taken by the message to arrive to destination.

We provide two different approaches to determine a path in the POppNet:

- Random path finder (R): time-based random walk search bounded by maximum path length, and execution time.
- Forward-backward path finder (FB): similar time-based search starting both from the source and destination node and performing a random intersection (following a typical meet-in-the-middle strategy).

III. Measuring anonymity

We introduce the following anonymity measures.

- Anonymity Degree $\mathcal{A}(s, t_s)$ for a given starting node $s \in V$ and time t_s. A normalized entropy-based measure of a the anonymity set. Determined by the possible destination nodes taking into consideration time constraints, and estimated path length and duration. This measure improves the typical use of the anonymity set by considering the probability that a node in such set has of being the destination node based on the paths that can reach it from the starting node.
- Approximated Anonymity Degree $\mathcal{A}'(s, t_s)$. An approximation of the previous measure. Note that computing $\mathcal{A}$ requires finding all simple paths from s at t_s, which has exponential memory requirements making it unfeasible for high density networks. If S is the anonymity set, and S' its approximation, we have that $S' \subseteq S$, although it does not imply that $\mathcal{A}' \leq \mathcal{A}$.
- Path-degree measure $\mathcal{D}(P)$ for a path P: metric based on the time based degree of all the nodes of a given path. In some sense these degrees give an estimation of the path uncertainty in each node. If an attacker knows a partial path, or can identify a given node in the path, the degree of the node and following unknown nodes

can be seen as the difficulty in guessing the rest of the path. Moreover, the degree of a node in the path can be seen as the probability that an attacker has in guessing the next node of the path knowing only the current node and time.

IV. Evaluation

We have used the CRAWDAD *rice/ad_hoc_city* dataset [5], a POppNet based on the Seattle public bus transportation. A high density network with close to 1200 nodes and more that $500,000$ dynamic edges over 24 hours. In order to analyze lower density networks we have obtained networks $N2$, $N4$, $N8$, $N16$, $N32$, by randomly selecting $1/2$, $1/4$, $1/8$, $1/16$, $1/32$ number of edges of the original network denoted as $N1$. Given that they correspond to the same network, with the same number of nodes and the same overall behavior it makes the comparison between them to be focused exclusively on the network density. We have randomly selected 200 cases with different starting and destination nodes, and starting time, divided in two sets: those starting at *rush* hour and those starting at time close to *zero* (midnight) and thus non-rush hour. For each case if the experiment includes stochastic results we perform 100 executions and take the average

Performance experiments show how good are the stochastic algorithms in terms of their efficiency, while privacy experiments attempt to evaluate their security. Efficiency is determined by comparing our R and FB routing approaches to a time constrained Dijkstra algorithm [6] denoted as X. This algorithm is deterministic and thus, not desirable for anonymous communications, but can be considered in most cases the best performance achievable.

Network	FB	R	X
N1	5.0	6.8	3.6
N2	5.0	6.8	3.9
N4	5.1	7.1	3.9
N8	5.1	7.2	4.0
N16	5.2	6.8	4.1
N32	5.5	7.1	4.7

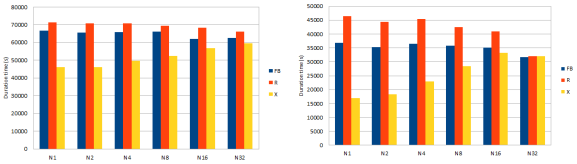
(a) Zero time

Network	FB	R	X
N1	5.0	6.8	3.5
N2	5.0	6.7	3.7
N4	5.1	7.1	3.7
N8	5.1	7.3	4.4
N16	5.2	7.1	4.5
N32	5.5	7.2	5.0

(b) Rush time.

TABLE I: Average path length

Table I shows the average path length for each execution of FB and R and the shortest path, X. FB obtains paths closest to the required minimum due to its meet-in-the-middle strategy. Paths from R are larger but within a reasonable range.



(a) Zero time (b) Rush time

Fig. 1: Duration time

Shortest duration times (Figure 1) are obtained in rush time experiments, and in general FB paths have a shorter duration than R. In any case we consider the penalty in duration time, as compared to X, to be quite acceptable for our scenarios.

Regarding anonymity, we show the anonymity degree in Table II, it approximation in Table III, and path degree measure IV.

Net	$\|S\|$	$\mathcal{A}$
N32	891.21	0.800
N16	1040.5	0.827
N8	1066.69	0.829

(a) Zero hours

Net	$\|S\|$	$\mathcal{A}$
N32	837.48	0.841
N16	1042.81	0.873
N8	1141.31	0.894

(b) Rush hours

TABLE II: Exact anonymity set size ($|S|$) and degree ($\mathcal{A}$) for N32, N16, N8.

Net	$\|S'\|$	$\mathcal{A}'$
N32	792.88	0.826
N16	970.88	0.889
N8	1027.2	0.908
N4	1061.1	0.921
N2	1067.69	0.923
N1	1074.93	0.925

(a) Zero hours

Net	$\|S'\|$	$\mathcal{A}'$
N32	565.77	0.737
N16	852.38	0.843
N8	966.36	0.891
N4	1020.53	0.909
N2	1027.16	0.909
N1	1042.02	0.913

(b) Rush hours

TABLE III: Approximated anonymity set size ($|S'|$) and degree ($\mathcal{A}'$).

Network	Zero	Rush
N1	$1.26E-20$	$1.22E-18$
N2	$4.65E-19$	$9.60E-18$
N4	$2.96E-16$	$1.78E-15$
N8	$2.89E-15$	$2.82E-14$
N16	$4.94E-11$	$3.26E-09$
N32	$6.10E-09$	$5.00E-14$

TABLE IV: Path-degree measure

V. Conclusions

We have analyzed the use of simple onion routing in POppNets to achieve message anonymity. We evaluated path establishment and anonymity in this network with variable network density. In overall we have shown that the approach is feasible in networks and its performance and anonymity is directly impacted by the network density. Our proposal is especially suitable in high density networks, allowing anonymous routing with relatively simple mechanisms as compared to other OppNet solutions.

Acknowledgments

Work partially supported by Spanish MINECO TIN2017-87211-R Project. D. Chen acknowledges the support from the China Scholarship Council, Grant No. 201606140138.

References

[1] V. F. Mota, F. D. Cunha, D. F. Macedo, J. M. Nogueira, and A. A. Loureiro, "Protocols, mobility models and tools in opportunistic networks: A survey," *Computer Communications*, vol. 48, pp. 5–19, 2014.

[2] K. Scott and S. Burleigh, "Bundle protocol specification," RFC 5050, IETF, 2007.

[3] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. of SIGCOMM '04*, 2004, p. 145–158.

[4] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in *Proc. of Information Hiding*, 1996, p. 137–150.

[5] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "CRAWDAD dataset rice/ad_hoc_city (v. 2003-09-11)," Downloaded from https://crawdad.org/rice/ad_hoc_city/20030911, Sep. 2003.

[6] B. B. Xuan, A. Ferreira, and A. Jarry, "Computing shortest, fastest, and foremost journeys in dynamic networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 02, pp. 267–285, 2003.

A Review of "Characteristics and Detectability of Windows Auto-Start Extensibility Points in Memory Forensics"

Daniel Uroz, Ricardo J. Rodríguez
Centro Universitario de la Defensa, Academia General Militar, Zaragoza, Spain
duroz@unizar.es, rjrodriguez@unizar.es

***Abstract*—Memory forensics consists in dumping the memory of a computer to a file and analyzing it with the appropriate tools. Many security incidents are caused by malware that targets and persists as long as possible in a Windows system within an organization. The persistence is achieved using *Auto-Start Extensibility Points* (ASEPs), the subset of OS and application extensibility points that allow a program to auto-start without any explicit user invocation. In this paper, we propose a taxonomy of the Windows ASEPs, considering the features that are used or abused by malware to achieve persistence. Many of these ASEPs rely on the Windows Registry. We also introduce the tool `Winesap`, a Volatility plugin that analyzes the registry-based Windows ASEPs in a memory dump.**

***Index Terms*—Memory forensics, malware, Volatility**

Tipo de contribución: *Investigación ya publicada*

I. Extended Abstract

Computer and network forensics is one of the fundamental steps performed during the detection and analysis stage in an incident response process. In the case of a security incident related to a compromised machine within an organization, an investigator can identify unauthorized and anomalous activity on the target computer or server, analyzing both the device drive and the memory. The latter case is named *memory forensics* [1]. In this paper, we focus on memory forensics since there are situations in which access to device drives are difficult (e.g., in cloud computing). Furthermore, the initial triage is faster since the data set in the memory is smaller than in the device drive.

Memory forensics is usually carried out by executing special software that captures and dumps into disk the current state of the system's memory as a snapshot file, also known as a *memory dump*. This file can then be taken offsite and analyzed with dedicated software such as Volatility [2] to search for evidences of the incident.

A memory dump is full of data to analyze. Every element susceptible to analyze is termed as a *memory artifact*. A memory dump contains a snapshot of the running processes, as well as other system information such as logged users, open files, or open network connections at the time of capturing the dump. Furthermore, since the memory used by object resources is not usually zeroed out when freed, the memory dump also contains this kind of data [1]. Hence, any memory artifact can be retrieved from a memory dump either using the appropriate internal OS structures to go through the data content or using a pattern-like search in the full dump.

Nowadays, one of the most common security incidents is the presence of software specially designed with malicious purposes (known as malicious software or *malware*). The life cycle of malware is composed of several stages [3]: first, it enters into the target computer to compromise it (using drive-by downloads, spear phishing, or other social engineering techniques, for instance). Then, the malware makes itself persistent in the system, i.e., it uses a persistence strategy to ensure that it will persist in the system within system reboots. Finally, it carries out its nefarious purposes.

The persistence stage of the malware is mainly motivated by the cybercriminal motto "*the longer the system is infected, the more the revenue*". The techniques used to persist in the system have been named *Auto-Start Extensibility Points* (ASEPs) in the literature [4], [5]. More formally, ASEPs refer to the subset of OS and application extensibility points that allow a program to auto-start without any explicit user invocation. Note that these extensibility points are also used by legitimate programs, such as system services or update agents, among other software.

In this paper, we focus on Windows OS since it is still the most predominant target platform of malware attacks, according to recent statistics [6]. In this regard, we study all the extensibility points in Windows OS that are susceptible to be used or abused by malware so that it can persist in the system. Furthermore, we introduce the tool `Winesap`, which extends the Volatility framework and allows a memory forensic analyst to detect the presence of unknown and rare programs in registry-based ASEPs.

Windows ASEPs can be classified according to the specific OS features used or abused by malicious programs to persist in the system. We have considered four categories: *system persistence mechanisms*, *program loader abuse*, *application abuse*, and *system behavior abuse*. For each category, we have distinguished different extensibility points.

Some extensibility points are only writable by programs with enough privileges. Furthermore, there are extensibility points that allow persistent programs to execute with elevated privileges, while in others the permissions of the signed-in user are inherited. The taxonomy that we propose is independent of the type (disk or memory) of forensics analysis. All the Windows ASEPs considered here are tracked down in a disk forensics analysis. However, some of these are undetected in the memory forensics of the Windows Registry (that is, we cannot tell which is the program being launched by that ASEP). Besides, it may happen that the ASEPs indicated as detectable in memory forensics could eventually be mapped out of the memory due to memory paging issues, and thus

Windows Auto-Start Extensibility Points	*Characteristics*					
	Write permissions	**Execution privileges**	**Tracked down in memory forensics**†	**Freshness of system**	**Execution scope**	**Configuration scope**
System persistence mechanisms						
Run keys (HKLM root key)	yes	user	yes	user session	application	system
Run keys (HKCU root key)	no	user	yes	user session	application	user
Startup folder (%ALLUSERSPROFILE%)	yes	user	no	user session	application	system
Startup folder (%APPDATA%)	no	user	no	user session	application	user
Scheduled tasks	yes	any	no	not needed‡	application	system
Services	yes	system	yes	not needed‡	application	system
Program loader abuse						
Image File Execution Options	yes	user	yes	not needed	application	system
Extension hijacking (HKLM root key)	yes	user	yes	not needed	application	system
Extension hijacking (HKCU root key)	no	user	yes	not needed	application	user
Shortcut manipulation	no	user	no	not needed	application	user
COM hijacking (HKLM root key)	yes	any	yes	not needed	system	system
COM hijacking (HKCU root key)	no	user	yes	not needed	system	user
Shim databases	yes	any	yes	not needed	application	system
Application abuse						
Trojanized system binaries	yes	any	no	not needed	system	system
Office add-ins	yes	user	yes	not needed	application	user
Browser helper objects	yes	user	yes	not needed	application	system
System behavior abuse						
Winlogon	yes	user	yes	user session	application	system
DLL hijacking	yes	any	no	not needed	system	system
AppInit DLLs	yes	any	yes	not needed	system	system
Active setup (HKML root key)	yes	user	yes	user session	application	system
Active setup (HKCU root key)	no	user	yes	user session	application	application

†*If the memory is paging to disk, it would be not possible to track down these ASEPs in memory forensics.*
‡*Depends on the trigger conditions defined to launch the program.*

TABLE I: A taxonomy of Windows ASEPs and a summary of their characteristics.

be undetected. As shown in the seminal work on memory forensics and the Windows Registry by Dolan-Gavitt [7], a large number of applications running in a system may cause unused portions of the registry to be paged out to disk.

Some extensibility points can require *the freshness of the system*, i.e., the system has to be rebooted or the user session has to be signed out and then signed in again to launch the programs set in these extensibility points. We indicate the minimum required freshness of the system for each technique (i.e., if an extensibility point requires only the user session to be restarted, it is obvious that a system reboot would have the same effect). The execution scope of an extensibility point is system-wide if any program in the system can interact with the program defined in that extensibility point. Otherwise, the execution scope is application-wide. Finally, certain extensibility points have different configuration scope, since they can be configured at system-level or at user-level, that is, they affect all the system as a whole or the current (signed-in) user session, respectively.

Table I summarizes the taxonomy of Windows ASEPs presented in this paper and their characteristics. For the sake of space, we deliberately omit here a detailed explanation of the considered ASEPs. A full description of every extensibility point can be found in [8].

Regarding the tool `Winesap`, it is implemented as a Python plugin on top of Volatility. It is specially designed to track down the previously described Windows ASEPs that rely on the Windows Registry. Our tool has been released under the GNU GPLv3 license and is freely available at https://gitlab.unizar.es/rrodrigu/winesap.

The tool works as follows: first, it obtains all possible detectable registry-based Windows ASEPs in a memory dump. It then determines whether the detected extensibility points are being abused by malware. In this regard, the tool performs different checks, depending on the data type contained in the register key value under analysis. An example of the output of the tool is as follows:

```
WARNING:
Suspicious path file
HKLM\Software\Microsoft\Windows NT\CurrentVersion\
    Image File Execution Options\firefox.exe
Debugger: REG_SZ: C:\Users\me\AppData\Roaming\
    Yztrpxpt\cmd.exe
------------------------------
WARNING:
Suspicious path file
HKLM\Software\Wow6432Node\Microsoft\
     Windows NT\CurrentVersion\Windows
AppInit_DLLs: REG_SZ: C:\Users\me\AppData\
     Roaming\Uxkgoeaoqbf\autoplay.dll
```

The full version of this paper was published in [8].

ACKNOWLEDGMENTS

This work was supported in part by the Aragonese Government under *Programa de Proyectos Estratégicos de Grupos de Investigación* (project reference T21-17R), in part by the Centro Universitario de la Defensa-Zaragoza (project reference CUD-2018-09), and in part by the Spanish Ministry of Economy, Industry and Competitiveness project MEDRESE (project number RTI2018-098543-B-I00).

REFERENCES

[1] M. H. Ligh, A. Case, J. Levy, and A. Walter, *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. John Wiley & Sons, Inc., Jul. 2014.
[2] A. Walters and N. Petroni, "Volatools: Integrating Volatile Memory Forensics into the Digital Investigation Process," in *BlackHat DC*, 2007.
[3] A. K. Sood, R. Bansal, and R. J. Enbody, "Cybercrime: Dissecting the State of Underground Enterprise," *IEEE Internet Computing*, vol. 17, no. 1, pp. 60–68, 2013.
[4] Y.-M. Wang, R. Roussev, C. Verbowski, A. Johnson, M.-W. Wu, Y. Huang, and S.-Y. Kuo, "Gatekeeper: Monitoring Auto-Start Extensibility Points (ASEPs) for Spyware Management," in *LISA'04*. USENIX Association, 2004, pp. 33–46.
[5] M. Russinovich and A. Margosis, *Troubleshooting with the Windows Sysinternals Tools*. Microsoft Press, 2016.
[6] AV-TEST Institute, "Security Report 2017/2018," techreport, Jul. 2018.
[7] B. Dolan-Gavitt, "Forensic analysis of the Windows registry in memory," *Digital Investigation*, vol. 5, pp. S26–S32, 2008.
[8] D. Uroz and R. J. Rodríguez, "Characteristics and Detectability of Windows Auto-Start Extensibility Points in Memory Forensics," *Digital Investigation*, vol. 28, pp. S95–S104, Apr. 2019. [Online]. Available: http://webdiis.unizar.es/~ricardo/files/papers/UR-DIIN-19.pdf

Design recommendations for online cybersecurity courses

L. González-Manzano[a*], J. M. de Fuentes[a],
[a]Computer Security Lab (COSEC). Universidad Carlos III de Madrid
[a]{lgmanzan, jfuentes}@inf.uc3m.es

***Abstract*- Nowadays, a significant amount of free online cybersecurity training courses are offered. When preparing further courses, the designer has to decide what to teach and how to do it. In this paper, we provide with a set of recommendations for both issues. Concerning topic selection, 35 free online courses are analysed using NIST's NICE reference framework. Thus, several training gaps are discovered. Concerning the way of preparing the course (or refining it after the first edition), a set of good practices is proposed based on students' performance and commitment in a cybersecurity MOOC with +2,000 initially active students. To foster further research in this area, an open-source framework is released to enable the analysis of students' performance in EdX MOOCs.**

Tipo de contribución: *Investigación publicada*

I. INTRODUCTION

Cybersecurity needs are becoming more and more widespread. According to the 2015 (ISC)2 Global Information Security Workforce Study, a shortfall of 1.8 million cybersecurity jobs will happen by 2022 [1]. Allegedly, the most common reason for this fact is the lack of qualified personnel.

Because of this trend, a plethora of training materials and courses have been proposed. Among them, two types of training actions can be found. On the one hand, virtual security labs are intended to provide with experimental training [2]. On the other hand, regular courses are usually considered to offer a theoretical and practical background on a topic. In the last years, a significant number of platforms are offering cybersecurity courses as part of their catalogue. Due to their success and general adoption, in this paper we focus on these initiatives.

One key aspect is how to determine which knowledge, skills and abilities (KSAs) are needed to become a cybersecurity professional. This specialty is not monolithic -- indeed, a huge amount of profiles can be devised.

When preparing a new cybersecurity course, one important issue is determining which topics are not properly addressed. Paulsen et al. already pointed out early in 2012 that cybersecurity educators have difficulty gaining a holistic view of the available resources and determining exactly what to teach [3]. However, to the best of authors' knowledge, this issue is still open.

Apart from the topics to be covered, cybersecurity trainers must pay attention to how these concepts are addressed. For this purpose, characterizing the target audience is at stake. As it happens in many disciplines, the background of students is relevant to decide the starting point and learning pace. These aspects should also be considered when refining the course after each edition.

This paper addresses both topic choice and course preparation issues. Concerning the first aspect, the current coverage of the NICE framework is studied. With respect to course preparation, we analyse students' performance and commitment of a Massive Online Open Course (MOOC) carried out in edX platform in 2017. This MOOC counted on +10,000 worldwide enrolled students among which +2,000 were initially active.

The structure of the paper is the following: Section II introduces NICE framework and analyses its coverage in existing online courses. Section III presents the analysis of our cybersecurity MOOC. Section IV points out recommendations according to the performed studied. Section V outlines conclusions.

II. NICE FRAMEWORK. COVERAGE

NICE Framework, developed by NIST [4], establishes a taxonomy to describe cybersecurity work roles with the intention to be applied in any sector, public, private or academic. In particular, the framework includes three different components: 7 categories which present cybersecurity functions at high level; 33 specialty areas to distinguish cybersecurity areas; and 52 work roles which define cybersecurity work in detailed according to specific KSAs required by each work role.

In this work, a total of 35 free courses have been identified. These courses are mainly taught in four platforms, namely Coursera, Cybrary.it, edX and Udacity.

Figure 1 shows the coverage of NICE framework categories. Note that each course typically covers more than one category. One important issue is that there is a huge difference between categories in what comes to their coverage. Thus, *Collect and Operate* (CO) and *Investigate* (IV) are covered by just 3 courses, whereas *Security Provision* (SP) or *Oversee and Govern* (OV) are the most offered aspects with 19 and 18 courses, respectively.

Concerning specialty areas (SA) within categories (33 SAs in total), on average 2.94 SAs are addressed in each course. Thus, most courses, 74%, are quite targeted, as they cover up to 3 SAs. Only a small portion covers a bigger amount, 17% of

courses cover between 4 and 7 SAs and the remaining 9% between 7 and 10 SAs. Among them, our proposed MOOC, *Cyber Security Basics: A Hands-on Approach*, is the most general one, addressing 10 SAs within all categories except for OV.

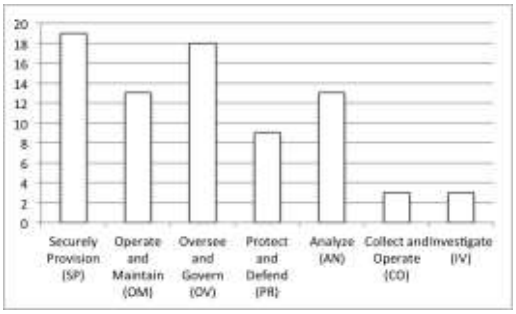


Fig 1. Coverage of cybersecurity domains from training courses

III. Cyber security basics: A hands-on approach MOOC analysis

This MOOC is an initiative to learn cybersecurity from a practical point of view. Theoretical explanations, essential for an appropriate comprehension of all concepts, are supported by examples and tools to guarantee a comprehensive learning process.

This course is composed of 6 lectures: 1. Introduction to cybersecurity; 2. Computer forensics; 3. Assembly programming; 4. Cyberdefense; 5. Malware and advanced persistent threats; and 6. Vulnerabilities and exposures. Each lesson contains a wide range of videos to enhance the students' learning experience. There are 93 videos with a duration of 6.9 hours, together with different activities like forums, homework or self-assignments, to motivate and guide students.

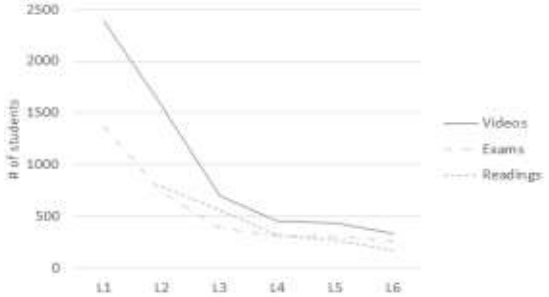


Fig 2. Number of students per lecture for exams, readings and videos.

By mid-March 2017 this course was opened, having active participation of teachers for the initial 8 weeks. After that, the course remained open without teacher intervention. In this study, we consider results obtained until 11 September 2017. To this date, 10,802 students were enrolled. However, only a fraction (2,387 students) were active, meaning that they have watched videos, done exams/readings or written comments. Figure 2 presents the evolution of active students per lecture. As expected, the number of involved students decreases every week which is in line with the fact that the completion rate is between 2% and 10%[1], 216 in this case. However, the number of students from lecture 4 on can be considered constant in terms of doing exams and readings. Note that in this MOOC Lectures 3, 4 and 6 are the most difficult ones and thus, in line with achieved results.

IV. Recommendations.

Before the development of a course:

- Topic choice: issues less covered by NICE should be addressed.
- Level choice: beginners' level in many areas are not considered.
- Teachers commitment: reduces problems along the course.
- Gender adaption: women are underrepresented.
- Regional adaption: students from India and United States are interesting targets.
- Interaction design: audience is important if participation is demanded. It differs between countries.
- Age impact: between 20 and 40 more committed. In line with the level of the course.

After first edition of the course:

- Suitability of content design: according to marks, level and content seem to be appropriate.
- Activities review: activities should be reviewed when many comments appear. It is the case of readings.
- Students videos commitment: videos could be too long or not attractive enough. In this MOOC videos length is right so attractiveness could be an issue to consider.

V. Conclusions and Future work

This paper presents an analysis of 35 cybersecurity online courses concerning NICE framework. Additionally, guidelines on the way to prepare a cybersecurity course are presented. A framework for the analysis of edX courses have been also released to promote the research in this direction[2]. An extended version of this paper is in [5].

Acknowledgement

This work is supported by the MINECO grant TIN2016-79095-C2-2-R and the CAM grant S2013/ICE-3095 and P2018/TCS4566 both co-funded with European FEDER funds.

References

[1] Frost and Sullivan. 2017 global information security workforce study. 2017.
[2] Khaled Salah. Harnessing the cloud for teaching cybersecurity. In Proceedings of the 45th ACM SIGCSE '14, pages 529–534, 2014.
[3] Celia Paulsen, Ernest McDuffie, William Newhouse, and Patricia Toth. Nice: Creating a cybersecurity workforce and aware public. IEEE Security & Privacy, 10(3):76–79, 2012.
[4] National Institute of Standards and Technology (NIST). National initiative for cybersecurity education (nice). cybersecurity workforce framework. 2017
[5] Gonzalez-Manzano, L., & de Fuentes, J. M. (2019). Design recommendations for online cybersecurity courses. Computers & Security, 80, 238-256.

[1] http://www.katyjordan.com/MOOCproject.html , last access March 2019

[2] https://github.com/lgmanzan/Toolset_EDX_dataProcessing

Proceso para la implementación de un ecosistema Big Data seguro

Julio Moreno
Grupo de investigación GSyA
Universidad de Castilla-La Mancha,
Ciudad Real, España
Julio.Moreno@uclm.es

Manuel A. Serrano
Grupo de investigación Alarcos
Universidad de Castilla-La Mancha,
Ciudad Real, España
Manuel.Serrano@uclm.es

Eduardo Fernández-Medina
Grupo de investigación GSyA
Universidad de Castilla-La Mancha,
Ciudad Real, España
Eduardo.FdezMedina@uclm.es

Eduardo B. Fernandez
Dept. of Comp. and Elect. Eng.
Florida Atlantic University,
Boca Raton, Florida, EEUU
fernande@fau.edu

Resumen- **Un entorno Big Data es un potente y complejo ecosistema que ayuda a las compañías a extraer información relevante de los datos, la cual, puede ser utilizada para ayudar en la toma de decisiones y en el desempeño del negocio. En este contexto, y debido a la cantidad, la variedad y la sensibilidad de los datos gestionados por este tipo de sistemas, la privacidad y seguridad se vuelven cruciales. Sin embargo, asegurar un ecosistema Big Data no es trivial y no debe ser tratado desde una perspectiva parcial o aislada, sino que se debe adoptar un enfoque holístico que comience desde el momento en el que se definen los requisitos y políticas, y continúe hasta que se desarrolle e implemente el ecosistema. Por todo ello, en este artículo, presentamos una visión metodológica para resolver este problema mediante la integración de la seguridad y privacidad en el proceso de implementación de un ecosistema Big Data. Nuestra propuesta se basa en los principales estándares y buenas prácticas internacionales. Además, nuestro proceso se encuentra soportado por una Arquitectura de Referencia de Seguridad para Big Data, la cual, establece los principales componentes de este tipo de tecnologías mientras incorpora conceptos de seguridad.**

***Index Terms*- Big Data, Seguridad por diseño, Desarrollo seguro, Patrones de seguridad, Arquitectura de referencia de seguridad**

Tipo de contribución: *Investigación original*

I. Introducción

Cada vez las organizaciones son más conscientes de la importancia de Big Data. Los datos son fundamentales para llevar a cabo sus actividades y ayudar a la alta directiva en la toma de decisiones [1]. El uso de un ecosistema Big Data implica un cambio en cuanto a los sistemas tradicionales en tres propiedades: la cantidad de datos (volumen), el ratio de generación y transmisión de los datos (velocidad) y la heterogeneidad de los tipos de datos estructurados y no estructurados que maneja (variedad). Estas propiedades son conocidas como el las tres Vs de Big Data [2]. Esta es la definición tradicional de Big Data, aunque diferentes autores han ido añadiendo nuevas Vs para adaptarla a la realidad actual, por ejemplo, la veracidad de los datos o el valor obtenido tras realizar los algoritmos de análisis [3]. Así, un ecosistema Big Data puede definirse como un conjunto de componentes que permiten almacenar, procesar, visualizar y entregar informes útiles para aplicaciones especializadas.

Sin embargo, con cada nueva tecnología surgen nuevos problemas, y Big Data no es una excepción. Al usar Big Data no solo aumenta la escala de los problemas tradicionales de privacidad y seguridad, sino que se añaden nuevos desafíos que deben ser afrontados. Estos problemas se derivan del hecho de que Big Data no fue inicialmente concebido como un entorno seguro [4]. Por ello, cuando una compañía decide desarrollar un ecosistema Big Data debe considerar estos problemas que pueden afectar a su implementación. En caso de no ser gestionados de forma apropiada, se pueden generar dificultades que pueden afectar a la organización, por ejemplo, un fallo para cumplir con la legislación que afecta al entorno Big Data puede resultar en la pérdida de reputación de la compañía o incluso en multas. Por ello, es importante que Big Data alcance un nivel de madurez y confianza suficiente, y para ello, es crucial disponer de metodologías, mecanismos y guías que permitan implementar no solo ecosistemas Big Data, sino también su seguridad. A esto se le suma la importancia actual de la seguridad por diseño, la cual, destaca la importancia de incorporar la seguridad en las etapas más tempranas del proceso de diseño y análisis de sistemas y tecnologías de la información [5].

Así, nuestra propuesta se basa en la definición de un proceso que integra aspectos de seguridad en el desarrollo de un ecosistema Big Data, y al mismo tiempo, considera sus características inherentes. Nuestra propuesta se compone de doce fases diferentes que cubren las principales etapas de desarrollo, haciendo hincapié en el análisis y diseño, los cuales, no suelen ser suficientemente considerados en este tipo de escenario. Por otro lado, la implementación del entorno no se descuida, todo lo contrario, durante esta etapa se destina cada fase para implementar cada uno de los componentes típicos de Big Data. Además, es importante destacar que un proceso de este tipo no debe ser solo una descripción de actividades, sino que debe estar soportado por una base conceptual que defina los principales componentes del sistema a desarrollar [6]. En nuestro caso, necesitábamos un metamodelo que cubriera los principales componentes de un ecosistema Big Data y, al mismo tiempo, incorporase aspectos de seguridad. Nuestra solución fue definir una Arquitectura de Referencia de Seguridad (SRA por sus siglas en inglés) para Big Data [7].

Una SRA es normalmente definida como una arquitectura de alto nivel que incorpora un conjunto de elementos que facilitan la definición de requisitos de seguridad y permiten un mejor entendimiento de conceptos como amenazas, vulnerabilidades o políticas de seguridad que pueden ser utilizados para describir un modelo conceptual de Big Data [8]. De esta forma, nuestra SRA está diseñada para ser capaz de usar patrones de diferentes tipos que faciliten la implementación del sistema y ayuden en la incorporación de requisitos no funcionales [9]. En este caso concreto, nos centraremos en los patrones de seguridad para facilitar la implementación de mecanismos de seguridad en Big Data.

El contenido del artículo se organiza de la siguiente forma: primero una sección de antecedentes; a continuación, consideramos necesaria realizar una breve explicación de nuestra SRA para Big Data. La siguiente sección se centra en la contribución principal de este artículo: la definición del proceso para crear Big Data seguros. Finalmente, presentamos las conclusiones y trabajo futuro.

II. Antecedentes

En general, no existen muchas propuestas que traten el problema de seguridad en Big Data desde una perspectiva metodológica. Por ello, para construir un proceso que incorpore seguridad en entornos Big Data, hemos llevado a cabo un estudio de las principales propuestas de metodologías de seguridad. Normalmente, estas aproximaciones se centran en sistemas software en general, por lo que, hay que adaptarlas para utilizarlas a la hora de implementar un ecosistema Big Data seguro.

Un ejemplo de metodología de implementación centrada en la seguridad puede ser Secure Tropos [10], la cual, es una extensión de Tropos que se centra en objetivos de seguridad y en la elicitación de requisitos de seguridad. Esto permite la integración de conceptos de seguridad en todo el proceso de desarrollo. Para ello, Secure Tropos utiliza una versión extendida del lenguaje i* que incluye conceptos como objetivos o tareas. Tropos [11] se basa en dos ideas principales: primero, el uso de conceptos cercanos a la mentalidad humana como, por ejemplo, objetivos o planes que guiarán el proceso de desarrollo. Por otro lado, destaca la importancia del análisis de requisitos en las fases más tempranas del proceso. Esto permite un mejor entendimiento del entorno a implementar.

Entre las principales propuestas metodológicas para incorporar seguridad al desarrollo se pueden destacar SecureUML y UMLSec. SecureUML [12] es un lenguaje de modelado que permite el desarrollo basado en modelos y tiene como principal propósito el aseguramiento de sistemas distribuidos. Esta propuesta se basa en el control de acceso basado en roles. Para ello, define un metamodelo que incorpora conceptos como los roles o los permisos. Por otro lado, UMLSec [13] se centra en el modelado de propiedades de seguridad en la fase de diseño mediante el uso de una extensión del lenguaje UML. Esta extensión define conceptos como estereotipos, valores etiquetados y restricciones que permiten la especificación de requisitos de seguridad.

SERENITY [14] es una metodología basada en patrones especialmente centrada en sistemas de Inteligencia Ambiental (AmI). Esta compuesta por dos partes que cubren el desarrollo y operación para la selección de las soluciones de seguridad. SERENITY propone un enfoque basado en objetivos de seguridad que guía el descubrimiento de requisitos y la selección de patrones. El Proceso Unificado Seguro [15] busca incorporar principios y disciplinas de seguridad en el Proceso Unificado de desarrollo, el cual, se puede considerar como un estándar *de facto* en el desarrollo de aplicaciones software.

Todos los enfoques descritos en esta sección son demasiado generales y deben ser adaptados al específico contexto en el que van a ser aplicados. Además, estas propuestas se centran principalmente en el modelado de sistemas software, mientras que la implementación ecosistema Big Data requiere una doble perspectiva: por un lado, considerar los servicios que provee, y por otro, la infraestructura hardware que soporta estos servicios. Ambas capas deben interactuar entre ellas, por lo que, las decisiones que se tomen para una capa afectarán a la otra y viceversa. Así, para nuestra propuesta de un proceso para la implementación de un Big Data seguro utilizamos conceptos y buenas prácticas expresadas en este tipo de metodologías, además de utilizar la SRA para Big Data como base para soportar nuestro proceso.

III. SRA para Big Data

En esta sección, resumimos nuestra propuesta de SRA [7], la cual, se basa en el esquema y los componentes propuestos por la organización NIST en su arquitectura de referencia para Big Data [16]. Al estar alineada con la propuesta del NIST, se consigue que nuestra SRA sea más fácil de implementar. La SRA destaca la importancia de aplicar conceptos de seguridad en la implementación de entornos Big Data. Para ello, nuestra propuesta tiene un enfoque que parte de los requisitos seguridad que son descritos en el primer componente de nuestra arquitectura: el System Orchestrator (SO). Es importante que estos requisitos de seguridad se encuentren alineados con los objetivos y políticas tanto de la organización como del entorno Big Data. Estos requisitos de seguridad pueden ser satisfechos por medio de diferentes soluciones de seguridad que siguen las políticas de seguridad de la compañía y que tienen el objetivo principal de contrarrestar las amenazas y controlar las vulnerabilidades. A este nivel, las soluciones de seguridad son todavía objetos abstractos que serán implementados en el resto de los componentes de la SRA. Para facilitar su implementación, nuestra SRA permite el uso de patrones. Un patrón de seguridad es una solución abstracta a un problema recurrente que indica cómo defendernos de una amenaza, o conjunto de amenazas, de una forma concisa y reusable [17]. Por todo ello, se puede decir que este componente es el más abstracto de nuestra arquitectura e influirá en la implementación del resto de componentes.

El siguiente componente de nuestra arquitectura es el Big Data Application Provider (BDAP), el cual, tiene el objetivo de satisfacer los requisitos establecidos en el SO. Para ello, el BDAP está compuesto por los diferentes servicios que ofrece Big Data. En general estos servicios son cinco: colección (recopilar los datos que alimentan la analítica), preparación (limpiar o estructurar los datos para mejorar los resultados), análisis (algoritmos para obtener información valiosa de los datos), visualización (representación de los datos) y control de acceso (quién puede acceder a qué datos). No es obligatorio que todos los ecosistemas Big Data provean de todos estos

servicios, hay algunos opcionales como la preparación o visualización de los resultados, que en función del contexto pueden no ser necesarios. Estos servicios son implementados a nivel del hardware en el siguiente componente.

El Big Data Framework Provider (BDFP) soporta las funcionalidades del BDAP. Para ello, normalmente se compone de uno o varios clusters que se componen a su vez de nodos. Además de la infraestructura hardware, este componente provee los servicios de almacenamiento, procesamiento y otros como las comunicaciones o la gestión de recursos. Actualmente, muchas compañías (sobre todo pequeñas y medianas) deciden externalizar esta parte de la arquitectura mediante la contratación de una solución Cloud comercial, sobre la cual, construyen su ecosistema Big Data.

Finalmente, los últimos dos componentes de la SRA son el Data Producer (DP) y el Data Consumer (DC), los cuales, tienen una función similar, pero en extremos opuestos de la arquitectura. Por un lado, el DP se encarga de alimentar con datos al ecosistema Big Data, sirviendo como punto de conexión con las fuentes de datos, estas fuentes de datos pueden ser tanto estructuradas como no estructuradas. Por otro lado, el DC es el componente que consume la información generada por el ecosistema Big Data, sirviendo de punto de conexión con el usuario final de los datos. Este usuario final no tiene por qué ser una persona física, sino que puede ser otro sistema. La Figura 1 muestra la estructura de la SRA con sus componentes y cómo se relacionan entre sí.

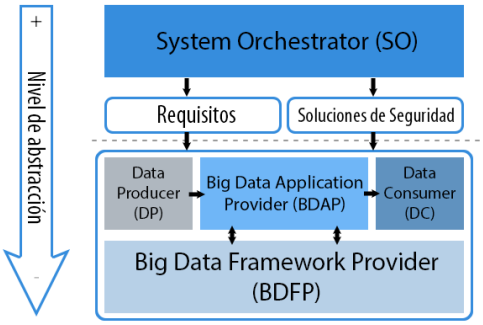


Fig. 1. Esquema general de SRA para Big Data

IV. Proceso para incorporar seguridad en el desarrollo de ecosistemas Big Data

A la hora de llevar a cabo un proyecto de Big Data es importante destacar que existen diferencias con el desarrollo de un proyecto software tradicional. Normalmente, los ecosistemas Big Data suelen ser entornos muy complejos, en los que no solo diferentes tecnologías interactúan, sino que también cobra gran relevancia la capa de hardware que las soporta. Además, el desarrollo de este tipo de sistemas normalmente implica el uso de herramientas software ya creadas, por lo que, la gestión de la configuración gana importancia. Por otro lado, el contexto de las compañías también es relevante y es que este tipo de sistemas se suelen implementar en organizaciones donde el *time to market* y la adaptación al cambio son cruciales para su éxito. Es más, muchas de estas compañías se encuentran inmersas en una revolución cultural interna para adaptarse a las tendencias ágiles, como el movimiento DevOps [18]. Debido a la presión y al entendimiento y uso erróneo de estas metodologías ágiles, el desarrollo de un ecosistema Big Data muchas veces no hace suficiente énfasis en las fases de análisis y diseño, lo que al final conlleva un aumento de la deuda técnica.

Para solucionar este problema, nuestra propuesta de proceso para la creación de Big Data seguros realiza fases de análisis y diseño. Así, nuestro proceso tiene dos conjuntos de fases: por un lado, las ya comentadas fases de análisis y diseño, y por otro, la implementación. Además, estas fases se componen a su vez de diferentes actividades. Las fases iniciales se centran principalmente en la definición de requisitos, soluciones de seguridad y riesgos que pueden afectar al entorno, los resultados de estas fases guiarán la implementación de Big Data que se lleva a cabo en el segundo conjunto de fases. Estos dos conjuntos de fases se encuentran muy relacionados entre sí, ya que, la finalización de las fases de diseño y análisis no implica que estas hayan acabado definitivamente. De hecho, nuestro proceso contempla la posibilidad de que durante la fase de implementación emerjan nuevos requisitos de seguridad, y por tanto, nuevas soluciones de seguridad y riesgos asociados a ellos. Una vez el proceso vuelve atrás, no implica que haya que reiniciarlo por completo, por ejemplo, si durante la implementación del componente de análisis se descubre un nuevo requisito de seguridad sobre cómo garantizar la privacidad de datos sensibles, entonces las tres primeras fases se ejecutarán de nuevo con el fin de elicitar correctamente dicho requisito, analizar y evaluar los riesgos asociados y decidir las soluciones de seguridad para implementarlo. Todos estos cambios no tienen por qué afectar al resto de componentes.

Este proceso ha sido validado mediante la creación de casos de estudio, en los cuales, se implementaban entornos de prueba partiendo de la definición de un objetivo global del Big Data a desarrollar. A partir de este objetivo se elicitaban los requisitos y las soluciones de seguridad que permiten cumplirlos. La selección de los requisitos y soluciones de seguridad se basaba en la opinión de expertos en la materia.

La Figura 2 muestra las diferentes fases del proceso divididas en el conjunto de fases de análisis y diseño e implementación, en las siguientes subsecciones se definirán estas fases. Para mejorar el entendimiento de nuestra propuesta y a modo de ejemplo, varias fases se modelarán utilizando SPEM 2.0 (Software & Systems Process Engineering Meta-Model) [19], la cual, es una especificación usada para definir los diferentes procesos de un sistema y sus componentes. La utilización de este tipo de modelado permite crear una representación homogénea y estandarizada del proceso, que puede ser utilizada y gestionada a través de repositorios electrónicos automatizados. Al usar este tipo de diagramas para modelar nuestra propuesta se obtiene un mejor entendimiento de la misma lo que permite una mayor facilidad en su uso.

A. *Fases de Análisis y diseño*

Este conjunto de fases se centra en descubrir los diferentes requisitos de seguridad que necesita el proyecto de Big Data. Estos requisitos de seguridad llevarán asociado un conjunto de soluciones de seguridad que los satisfaga. Además, también se realiza un análisis de riesgos basado en el contexto, lo cual, puede ocasionar el descubrimiento de nuevos requisitos. Por tanto, hasta que no se considere que la elicitación de las

necesidades de seguridad es suficientemente completa, no se avanzará a la siguientes, sino que se realizarán nuevas iteraciones. A continuación, se definen las diferentes fases.

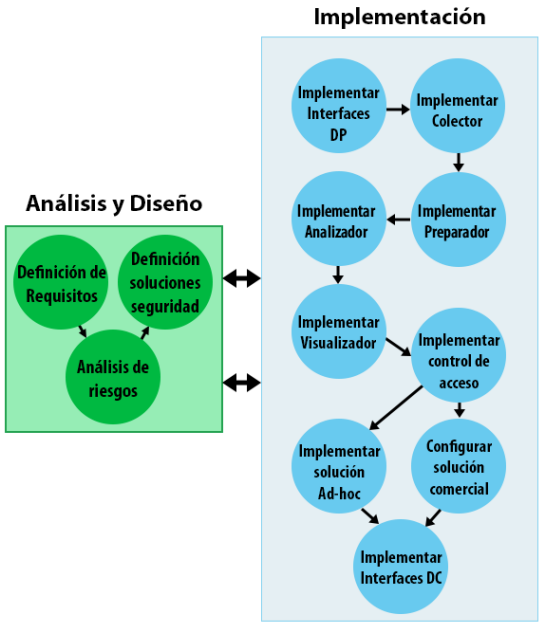


Fig. 2. Proceso para la implementación de Big Data seguro

1. *Fase 1: Definición de requisitos*

El principal objetivo de esta fase es obtener los requisitos del ecosistema Big Data, para ello se compone de cuatro actividades. La primera actividad es la definición de los objetivos de Big Data, es decir, el propósito principal del sistema a desarrollar. Estos objetivos deben estar alineados con las políticas y objetivo de negocio de la compañía. Existen algunos métodos para la obtención y representación de los objetivos, por ejemplo, GORE (Goal-Oriented Requirement Engineering) o i*. Ninguna de estas metodologías es específica para ecosistemas Big Data, pero se pueden adaptar para este propósito [20], [21].

La segunda actividad se centra en la definición de los requisitos de seguridad. Para ello, se consideran los objetivos obtenidos en la fase anterior y el contexto en el que se desarrollará el sistema. El contexto de la compañía es un conjunto de características que pueden cambiar los requisitos del ecosistema Big Data; por ejemplo, los requisitos de seguridad necesarios para un hospital serán más estrictos en temas de seguridad. Este contexto también incluye las regulaciones legales que pueden afectar al sistema. Para modelar los requisitos de seguridad se puede utilizar UMLSec [13] que permite expresarlos haciendo hincapié en dimensiones como la confidencialidad, integridad y disponibilidad del sistema. Finalmente, la tercera y cuarta actividad se dedican a la selección y adquisición de los activos que puedan cumplir con los requisitos definidos en las actividades anteriores. En general, hay seis tipos diferentes de activos que pueden ser identificados en un ecosistema de Big Data: la infraestructura de hardware, los servicios y aplicaciones, los datos y metadatos, los recursos analíticos, las técnicas de seguridad y privacidad, y los individuos y roles. Es importante realizar un estudio riguroso de las diferentes posibilidades para decidir cuál es la opción que mejor se adapta a sus necesidades de Big Data. La selección de los activos influirá en gran medida en la implementación del ecosistema Big Data, por lo que es necesario comprobar la compatibilidad entre los diferentes elementos antes de adquirirlos. En algunos casos, los activos ya forman parte de la empresa, por lo que no es necesario adquirirlos. Una solución ampliamente utilizada para este tipo de problemas son los árboles de toma de decisión que permiten la comparación de ventajas y desventajas entre diferentes posibilidades.

2. *Fase 2: Análisis de riesgos*

El objetivo principal de esta segunda fase es la definición de los riesgos que afectan al entorno Big Data, para ello se definen tres actividades. Cuando nos referimos a los riesgos, nos basamos en la definición tradicional de riesgo: “un evento potencial que tiene cierta probabilidad de ocurrir con potenciales elementos de seguridad involucrados” [22]. Por lo tanto, la primera actividad es la definición de las vulnerabilidades que pueden afectar a los activos. Los activos seleccionados, probablemente, tendrán un conjunto de vulnerabilidades que ya han sido identificadas por la comunidad. Estas vulnerabilidades pueden ser explotadas por las amenazas, cuya definición corresponde a la segunda actividad de esta fase. Además de las amenazas propias de los activos, existen más amenazas que deben considerarse, por ejemplo, ENISA (European Union Agency for Network and Information Security) ha creado una lista de las principales amenazas que se pueden encontrar en Big Data [23]. Además, existen diferentes técnicas que facilitan el descubrimiento de amenazas como árboles de ataque o casos de mal uso.

Una vez identificados todos los riesgos del ecosistema de Big Data, la actividad de evaluación de riesgos se centrará en realizar un análisis cuantitativo y cualitativo de los riesgos. Por lo tanto, sobre la base de ese análisis se obtendrá una lista priorizada de riesgos. Esta lista permitirá a los *stakeholders* decidir cómo hacer frente a los riesgos, por ejemplo, algunos son importantes y deben prevenirse y, por otra parte, hay otros que no son tan significativos y son aceptados por la organización. La decisión de esta clasificación también dependerá del apetito de riesgo de la empresa (el nivel de exposición que están dispuestos a aceptar). No existe un método específico para tratar los riesgos de Big Data, sin embargo, hay muchas propuestas para la evaluación de riesgos de TI en general; por ejemplo, MAGERIT, OCTAVE, CRAMM, o ISO 31000. Como sucedió con los requisitos, el descubrimiento de nuevas vulnerabilidades, amenazas y riesgos es una fase en curso que puede evolucionar durante el proceso de implementación del ecosistema de Big Data.

3. *Fase 3: Definición de soluciones de seguridad*

Esta fase se centra en la definición de soluciones de seguridad que aborden las amenazas y riesgos definidos en la fase anterior. Además, la definición de estas soluciones de seguridad puede llevar a la creación de metadatos de seguridad que ayuden en la implementación. Sin embargo, las soluciones de seguridad definidas en esta fase se encuentran todavía en un nivel muy abstracto, por lo que deben ser implementadas en los niveles inferiores de la arquitectura donde las amenazas pueden afectar realmente a los activos. Encontrar estas soluciones de seguridad es la primera actividad de esta fase.

La segunda actividad es la selección de patrones de seguridad. Como ya se ha dicho, los patrones de seguridad son artefactos que facilitan la implementación de soluciones de seguridad. Existen algunas metodologías propuestas por la comunidad para abordar el problema de la aplicación de patrones de seguridad en la implementación de un sistema de TI [17], [24]. En general, estas metodologías proponen un proceso para cubrir los aspectos de seguridad que es similar a nuestro enfoque, de modo que puedan ser utilizadas conjuntamente. Sin embargo, es posible que no exista un patrón de seguridad que aborde una amenaza o vulnerabilidad específica, en cuyo caso, la solución de seguridad debería crearse desde cero. Otra posibilidad es adaptar los patrones de seguridad de otros campos.

Finalmente, la tercera actividad puede ser considerada como una forma de mejorar los aspectos de seguridad del ecosistema, ya que su objetivo principal es la identificación de amenazas y vulnerabilidades que antes no se consideraban. Para ello, el uso de patrones de mal uso es una práctica interesante desde el punto de vista de la seguridad. Se basa en los objetivos del atacante relacionados con los activos del sistema, por lo que da una nueva perspectiva. Un patrón de mal uso define el uso no autorizado de un activo y cómo se realiza este ataque. También describe las contramedidas que pueden utilizarse para reducir ese riesgo [25]. Que sepamos, no existen patrones específicos de mal uso para los escenarios de Big Data, sin embargo, es posible adaptar los existentes a este tipo de entornos o incluso crearlos. La Figura 3 representa el modelado de este proceso en SPEM, mostrando los artefactos y roles de usuario que participan. Como se puede observar los principales roles que se encargan de realizar esta fase son el CISO (Chief Information Security Officer) y el CTO (Chief Technology Officer), los cuales, reportan los resultados al CSO (Chief Security Officer). Al final de esta fase se describe una primera aproximación a las soluciones de seguridad que deben ser implementadas en el sistema. Sin embargo, y al igual que sucedió con los requisitos y riesgos, esta definición puede actualizarse si durante la fase de implementación se identifican nuevos requisitos de seguridad.

B. *Fases de Implementación*

Una vez definidos los requisitos y soluciones de seguridad abstractas del ecosistema Big Data es el momento de implementarlas. Para ello, se sigue un conjunto de fases que abordan los principales componentes de un entorno Big Data. En caso de descubrir algún nuevo requisito, será necesario volver a las fases anteriores.

1. *Fase 4: Implementar las interfaces del DP*

El objetivo principal de esta fase es la descripción de las fuentes de datos que alimentarán el ecosistema de Big Data, así como las restricciones que deben aplicarse debido a los requisitos de seguridad del ecosistema de Big Data y de las propias fuentes de datos. Por lo tanto, el primer paso es la definición de las diferentes fuentes de datos que se utilizarán para cumplir con los requisitos definidos. Por lo general, un ecosistema de Big Data utiliza diferentes fuentes de datos para realizar su análisis.

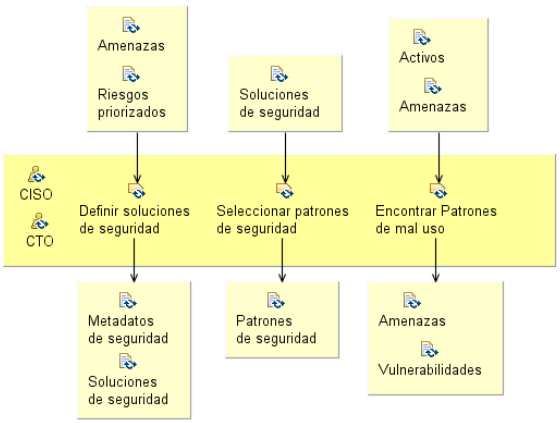


Fig. 3 Fase 3, Definición de soluciones de seguridad

Por ello, una buena práctica es la creación de un metamodelo de las diferentes fuentes de datos que represente cómo se relacionan los datos y qué datos se utilizarán. Este metamodelo se utilizará en la implementación del Colector. Una vez completado, las restricciones de acceso a los datos deben implementarse mediante interfaces. Estas interfaces son una implementación software de los requisitos de seguridad del ecosistema de Big Data y de las políticas que pueden tener las fuentes de datos. Esta fase está muy relacionada con la implementación del Colector, por lo que en algunas ocasiones es posible realizarlas al mismo tiempo. Este enfoque permite un mejor alineamiento entre estos dos componentes.

2. *Fase 5: Implementar el Colector*

Su principal objetivo es la implementación del componente Colector. Sin embargo, la primera actividad será la de definir el Data Lake que lo conforma. Un Data Lake es un repositorio de almacenamiento que contiene una gran cantidad de datos brutos tal y como fueron generados mientras no sea necesario procesarlos. En general, los Data Lake almacenan datos no estructurados, pero pueden combinar diferentes tipos de datos. El Data Lake es una parte muy importante del componente Colector, ya que almacena los datos "en bruto" (*raw data*) recibidos de las fuentes de datos. Por esta razón, es importante que se ajuste a los requisitos y el metamodelo de los datos definido en la fase anterior. De hecho, este "lago de datos" puede gestionarse mejor si utilizamos metadatos para intentar abordar la problemática de tener una gran cantidad de datos desorganizados [26]. A la hora de realizar la definición del data lake, se tendrán que considerar los requisitos de seguridad y patrones de seguridad previamente definidos. En caso de haber requisitos de seguridad que no puedan ser cubiertos por patrones, se deberán implementar *ad-hoc*.

La segunda actividad se centra en la implantación del servicio de recogida que tiene como objetivo la obtención de datos de diferentes fuentes de datos. Dependiendo del tipo de fuente de datos que se necesite, es posible que se necesite utilizar diferentes aplicaciones. Por ejemplo, si los datos se almacenan en una base de datos relacional, se pueden exportar al almacenamiento de Big Data mediante Apache Sqoop. También se incluyen los datos que deben ser analizados en tiempo real, por ejemplo, en un escenario donde un archivo de registro requiere ser procesado. Todos estos datos se almacenarán en el Data Lake. Las soluciones de seguridad también deben ser implementadas en esta actividad. Estas

soluciones son fundamentales en este componente puesto que almacena datos que pueden tener carácter personal, por tanto, el control de acceso, la integridad y la trazabilidad de las acciones realizadas sobre los datos es crítica.

3. *Fase 6: Implementar el Preparador*

Durante esta fase, se implementará el servicio de preparación de los datos. Normalmente, en este tipo de escenarios solo una pequeña parte de los datos es realmente útil para lograr el objetivo, por ello, esta fase es altamente recomendable para analizar los datos correctamente. Además, este servicio está muy relacionado con una de las V's de Big Data: el valor. La identificación de los datos necesarios es la primera actividad de esta fase. Para ello, es importante considerar cuál es el objetivo que se quiere alcanzar con el análisis. Por esta razón, los requisitos (incluidos los de seguridad) son una de las entradas de esta actividad. Como salida se generará un repositorio de datos etiquetados, donde se marcan los datos que se utilizarán en la fase de análisis.

Tras esto es el momento de implementar los diferentes *scripts* de código que transformarán los datos para facilitar su análisis. En este caso, como los scripts de preparación pueden tener acceso a datos personales, es importante controlar su implementación para garantizar que se preserva la seguridad. Por otro lado, algunas de las técnicas que se pueden utilizar para preparar los datos incluyen la detección de valores perdidos y valores atípicos que pueden empeorar el análisis de los datos. Además, hay muchas aplicaciones comerciales que se centran en facilitar la preparación de los datos.

4. *Fase 7: Implementar el Analizador*

Esta fase tiene como objetivo principal la implementación del servicio de análisis. En general, este servicio es el más importante en un ecosistema de Big Data. En primer lugar, se deben considerar los requisitos (seguridad y funcionales), activos y los patrones de seguridad para determinar cómo se producirán los resultados deseados. En otras palabras, describir los algoritmos y la tecnología para implementarlos. Existen diferentes formas de obtener valor de los datos, por lo que los algoritmos a utilizar vendrán determinados por los requisitos sobre cómo analizarlos y cuál es el valor que se quiere obtener, mientras que no se pierde de vista los requisitos de seguridad del entorno. Por ejemplo, se puede utilizar un enfoque basado en técnicas de *machine* o d*eep learning*, sin olvidar la forma más extendida de realizar análisis en Big Data: MapReduce (aunque hoy en día está cayendo lentamente en desuso en favor de otras tecnologías como Apache Spark) [27].

Dado que Big Data utiliza una gran cantidad de datos ambientales y humanos para obtener información valiosa, el principal problema es cómo proteger la privacidad. Muchas veces es difícil encontrar el equilibrio entre la obtención de información útil y la garantía de protección de la privacidad del usuario. Otro problema típico que considerar en los ecosistemas de Big Data es la información inferida de los datos. En Big Data, es posible obtener información sensible a partir de datos que no tenían un nivel especial de sensibilidad. Todos estos escenarios deben ser considerados al implementar los algoritmos de análisis. Finalmente, la última actividad trata sobre la realización de pruebas para comprobar que se cumplen tanto los requisitos funcionales como de seguridad. La Figura 4 muestra el modelado de esta fase con SPEM. En este caso los roles encargados tienen un perfil más cercano a la tecnología, aunque son supervisados por el CISO y el CTO.

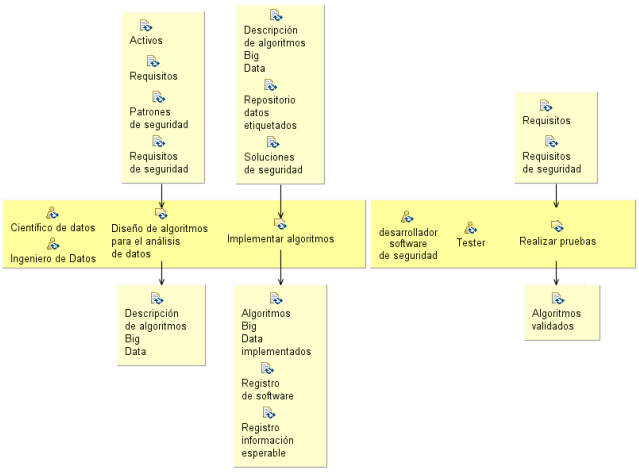


Fig. 4 Fase 7, implementar el analizador

5. *Fase 8: Implementar el Visualizador*

Esta fase tiene el propósito principal de implementar el componente Visualizador. El servicio de visualización proporciona representación de la información obtenida. Este servicio no es obligatorio para todos los casos, por ejemplo, si la información es consumida por otro sistema, el componente de visualización no es necesario. Para decidir qué técnica de visualización es la más apropiada es importante tener un fuerte conocimiento de los actores que utilizarán la información, a fin de satisfacer sus necesidades. En general, las técnicas de visualización se pueden dividir en dos categorías: por un lado, los datos se pueden visualizar en forma de gráfico o gráfico de cualquier tipo; por otro lado, los datos también se pueden representar por medio de un cuadro de mando, en este caso, la información representada está más enfocada a la alta dirección de una organización.

En cuanto a las soluciones de seguridad, en este nivel es importante preocuparse por la información que pueden consultar los *skakeholders*. Esta cuestión se trata en gran medida en la próxima fase, aunque hay algunos detalles a abordar. Por ejemplo, es posible que, debido a la representación de la información, un científico de datos pueda inferir información personal que deba ser protegida. En este caso, se necesita una capa adicional de protección para evitar que suceda.

6. *Fase 9: Implementar el Control de Acceso*

Esta fase finaliza la implementación del componente BDAP. Se centra en la definición e implementación de las reglas de control de acceso y está compuesta por dos actividades. El control de acceso es un servicio que tiene como objetivo principal restringir la lectura de la información. En los entornos de Big Data suele haber distintos *stakeholders* que sólo deben acceder a una parte de la información. De hecho, esta fase depende de cómo se definieron los requisitos de los *stakeholders*. Sobre la base de esos requisitos y de la información obtenida en el servicio de análisis, las normas de control de acceso deben estar bien definidas y, a continuación,

aplicarse. Esta implementación está muy influenciada por la tecnología que se está utilizando, ya que, cada una de ellas tiene una forma diferente de proporcionar control de acceso. Por ejemplo, Apache Spark utiliza Kerberos para realizar el control de acceso. En realidad, estas reglas de control de acceso son la implementación de soluciones de seguridad que fueron definidas en el componente SO. La implementación de este componente también puede verse favorecida por la utilización de modelos de seguridad.

7. *Fase 10. A) Implementar una solución ad-hoc*

Esta fase se centra en la implementación del componente BDFP, aunque hay dos formas principales de abordarlo: mediante la implementación de una solución ad-hoc que cumpla mejor con los requisitos de Big Data o mediante el uso de una solución comercial. El componente BDFP tiene como objetivo implementar la arquitectura de hardware necesaria para realizar los servicios del BDAP. Para lograr este objetivo, esta fase está compuesta por cinco actividades que se centran en diferentes partes del componente BDFP. La primera actividad consiste en desplegar los clústeres y nodos que conforman el ecosistema de Big Data. En un contexto de Big Data, un clúster suele definirse como un grupo de servidores que tienen diferentes funciones para lograr un objetivo general principal: obtener información valiosa de los datos. Estos clústeres pueden componerse de muchos nodos diferentes. Dependiendo de las necesidades del proyecto, debe decidir si va a utilizar una configuración especializada de los nodos (cada uno de los cuales realizará una acción concreta) o si cada nodo tendrá una configuración estándar que se ajuste a los requisitos. Existen algunas ecuaciones que permiten realizar una estimación del tamaño que debe tener el clúster o el número de nodos que se necesitan [28]. Otra forma de utilizar los nodos en un entorno Big Data son los contenedores. Un contenedor es una agregación de diferentes tecnologías que existen en el sistema operativo y que permiten ejecutar una aplicación, normalmente de un único proceso, dentro de un sistema operativo. Esta aplicación se ejecuta como un proceso aislado en espacio de usuario en el sistema operativo del anfitrión, por lo que, disfruta del aislamiento de recursos y de los beneficios de la asignación de recursos de las máquinas virtuales, pero es mucho más portátil y eficiente [29]. Este tipo de tecnologías tiene el propósito de realizar una gestión de alto nivel del hardware subyacente.

La siguiente actividad es la implementación del sistema de almacenamiento. En general, hay tres maneras diferentes de almacenar datos en Big Data dependiendo del formato de los datos y de los requisitos: estructurado, semiestructurado y no estructurado. El almacenamiento estructurado son las bases de datos relacionales tradicionales. Normalmente, utilizan un lenguaje similar al SQL. Por el contrario, las bases de datos no estructuradas, generalmente conocidas como NoSQL, son ampliamente utilizadas en los ecosistemas de Big Data. En este tipo de bases de datos, existen cuatro subtipos diferentes: basados en grafos (normalmente usados para representar datos de redes sociales), columnares (en estas bases de datos cada clave está asociada con uno o más atributos, a diferencia de las relacionales. Son adecuados para aplicaciones analíticas donde se realizan muchas operaciones comunes sobre los datos), documentales (Estas bases de datos almacenan los datos como un formulario de un documento, su principal ventaja es la escalabilidad), y de clave-valor (parecido a las tablas hash donde cada clave está asociada a un conjunto de valores). Además, para esta funcionalidad es necesario disponer de un sistema de ficheros que soporte los ficheros necesarios para proporcionar los diferentes servicios del ecosistema Big Data; normalmente, en este contexto, el HDFS (Hadoop Distributed File System) puede ser considerado como un estándar de facto.

A continuación, se define la capa de procesamiento. En el contexto de Big Data, existen tres tipos diferentes de procesamiento. Una vez más, dependiendo de las necesidades de su proyecto, debería utilizar la configuración que más se ajuste a sus necesidades. El procesamiento por lotes normalmente está relacionado con el paradigma MapReduce, que ejecuta los diferentes trabajos secuencialmente, escribiendo en el disco para almacenar los resultados entre fases. Por otro lado, puede ser necesario procesar los datos en tiempo real mediante un procesamiento en *streaming*. Algunos ejemplos de tecnologías que permiten implementar estos requisitos son Apache Spark y Apache Flink. En un punto intermedio entre estas tecnologías está el procesamiento interactivo, una posibilidad que se está haciendo cada vez más relevante en los entornos Big Data [16]. Estas soluciones permiten realizar consultas sobre los datos mientras se están recibiendo.

Finalmente, las últimas dos actividades se centran en implementar los servicios de apoyo (comunicaciones y gestión de recursos). La funcionalidad de comunicaciones se refiere a cómo se comunican entre sí los diferentes componentes y procesos del ecosistema de Big Data. Por otro lado, la gestión recursos tiene el propósito de controlar y gestionar cómo se utilizan los recursos de cada nodo. Esta funcionalidad es especialmente importante si se utiliza una configuración de nodos en la que cada uno de ellos tiene tecnologías diferentes en funcionamiento.

8. *Fase 10. B) Configurar solución comercial*

Por otro lado, también existe la posibilidad de abstraer la tecnología y componentes del ecosistema de Big Data mediante la contratación de un IaaS (Infraestructura como servicio) virtual. Estos servicios pueden facilitar la implementación del componente BDFP haciéndolo transparente para el usuario. Por lo tanto, esta es una buena opción si no se necesita una solución *ad-hoc* para el sistema o si se trata de un desarrollador sin experiencia en Big Data. Esta fase tiene dos actividades. En primer lugar, hay muchos proveedores diferentes que deben ser considerados. Para elegir la que mejor se adapte a las necesidades del ecosistema de Big Data, no sólo se deben considerar los requisitos, sino también las tecnologías que se han seleccionado para implementar los servicios del BDAP. Además, existen otros criterios a tener en cuenta, por ejemplo, razones económicas o de reputación del proveedor. Existen técnicas generales que pueden ayudar en esta decisión, por ejemplo, los diagramas de árbol de decisión. Una vez seleccionado el proveedor, hay otra actividad a realizar: la configuración del IaaS. Dependiendo del proveedor seleccionado las posibilidades de configuración cambian. Esta actividad debe cubrir todas las características necesarias para soportar el BDAP, incluyendo el tipo de almacenamiento y el motor de *streaming*. Normalmente, este tipo de IaaS incluye un cuadro de mando que facilita la monitorización de todos los

componentes del sistema y permite una configuración de hardware flexible. Esta fase de configuración puede seguir un flujo de actividades similar al descrito en la fase anterior.

8. *Fase 11: Implementar las interfaces del DC*

Esta es la última fase de nuestro proceso. El objetivo principal de esta fase es la descripción de los consumidores de datos que utilizarán los resultados producidos por el ecosistema de Big Data, así como las restricciones que deben aplicarse debido a los requisitos de seguridad del ecosistema de Big Data. Por ello, el primer paso es la definición de los diferentes consumidores de datos y las restricciones de acceso a la información. En general, un ecosistema de Big Data tiene diferentes *stakeholders* accediendo a la información, sin embargo, dependiendo de sus roles, tendrán diferentes limitaciones. Al igual que en el caso del DP, el uso de diferentes diagramas puede ayudar a la implementación de este componente, por ejemplo, los diagramas de secuencia UML. Una vez completado, las restricciones de acceso a la información deben ser implementadas mediante el uso de interfaces. Estas interfaces son las puertas que protegen el acceso a la información generada por el Big Data y pueden considerarse como una implementación de las soluciones de seguridad definidas en el componente SO.

V. Conclusiones y Trabajo Futuro

El desarrollo de un ecosistema seguro de Big Data no es un proyecto trivial. En general, conlleva lidiar con nuevos problemas de seguridad que no se habían considerado anteriormente. Además, un ecosistema de este tipo suele incluir el uso de diferentes tecnologías que interactúan entre sí, lo que complica su aplicación. Por esta razón, en este trabajo presentamos nuestra propuesta de un proceso para incorporar la seguridad al desarrollo de un ecosistema de Big Data. Este proceso cubre las fases típicas de un proceso de desarrollo, desde el análisis hasta la implementación. Además, este proceso fue concebido considerando el escenario actual de las empresas, en el que muchas de ellas están inmersas en un cambio cultural interno para adoptarse a conceptos como las metodologías ágiles. Este proceso está soportado por una SRA que actúa como metamodelo de los diferentes componentes que habitualmente conforman un ecosistema de Big Data permitiendo su abstracción, lo que facilitará el desarrollo de un entorno tan complejo. Como trabajo futuro, nuestra propuesta será validada mediante un caso de estudio en un entorno real lo que nos permitirá refinar nuestra propuesta. Por otro lado, y aunque se han mostrado un par de ejemplos, se realizará una definición formal, completa y detallada de todas las fases y artefactos que conforman nuestra SRA utilizando SPEM.

Agradecimientos

Este trabajo ha sido financiado por el proyecto ECLIPSE (Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional FEDER) y el proyecto GENESIS (Consejería de Educación, Cultura y Deportes de la Dirección General de Universidades, Investigación e Innovación de la JCCM, SBPLY-17-180501-000202).

Referencias

[1] J. Akoka, I. Comyn-Wattiau, and N. Laoufi, 'Research on Big Data – A systematic mapping study', *Computer Standards & Interfaces*, vol. 54, no. Part 2, pp. 105–115, Nov. 2017.
[2] S. Sagiroglu and D. Sinanc, 'Big data: A review', in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, 2013, pp. 42–47.
[3] Z. Sun, K. Strang, and R. Li, 'Big Data with Ten Big Characteristics', presented at the Proceedings of the 2nd International Conference on Big Data Research, 2018, pp. 56–61.
[4] P. P. Sharma and C. P. Navdeti, 'Securing big data hadoop: a review of security issues, threats and solution', *Int. J. Comput. Sci. Inf. Technol*, vol. 5, 2014.
[5] V. Casola, A. De Benedictis, M. Rak, and E. Rios, 'Security-by-design in Clouds: A Security-SLA Driven Methodology to Build Secure Cloud Applications', *Procedia Computer Science*, vol. 97, pp. 53–62, Jan. 2016.
[6] A. V. Uzunov, E. B. Fernandez, and K. Falkner, 'Assessing and improving the quality of security methodologies for distributed systems', *Journal of Software: Evolution and Process*, vol. 30, no. 11, p. e1980, 2018.
[7] J. Moreno, M. A. Serrano, E. Fernandez-Medina, and E. B. Fernandez, 'Towards a security reference architecture for big data', presented at the CEUR Workshop Proceedings, 2018, vol. 2062.
[8] F. Liu *et al.*, 'NIST cloud computing reference architecture', *NIST special publication*, vol. 500, no. 2011, p. 292, 2011.
[9] E. B. Fernandez, N. Yoshioka, H. Washizaki, and M. H. Syed, 'Modeling and Security in Cloud Ecosystems', *Future Internet*, vol. 8, no. 2, p. 13, Apr. 2016.
[10] H. Mouratidis and P. Giorgini, 'Secure Tropos: A security-oriented extension of the Tropos methodology', *International Journal of Software Engineering and Knowledge Engineering*, vol. 17, no. 2, pp. 285–309, 2007.
[11] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, 'Tropos: An agent-oriented software development methodology', *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004.
[12] T. Lodderstedt, D. Basin, and J. Doser, 'SecureUML: A UML-based modeling language for model-driven security', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2460 LNCS, pp. 426–441, 2002.
[13] J. Jürjens, 'UMLsec: Extending UML for Secure Systems Development', in *≪UML≫ 2002 — The Unified Modeling Language*, 2002, pp. 412–425.
[14] D. Serrano, A. Maña, R. Llarena, B. G.-N. Crespo, and K. Li, 'SERENITY Aware System Development Process', *Advances in Information Security*, vol. 45, pp. 165–179, 2009.
[15] C. Steel, R. Nagappan, and R. Lai, 'The alchemy of security design methodology, patterns, and reality checks', *Core Security Patterns: Best Practices and Strategies for J2EE, Web Services, and Identity Management, Prentice Hall*, p. 1088, 2005.
[16] NBD-WG, NIST, 'NIST Big Data Reference Architecture', Jun-2018. [Online]. Available: https://bigdatawg.nist.gov/_uploadfiles/M0639_v1_9796711131.docx. [Accessed: 10-Jan-2019].
[17] E. B. Fernandez, *Security patterns in practice: designing secure architectures using software patterns*. John Wiley & Sons, 2013.
[18] J. Carrasco, F. Durán, and E. Pimentel, 'Trans-cloud: CAMP/TOSCA-based bidimensional cross-cloud', *Computer Standards & Interfaces*, vol. 58, pp. 167–179, May 2018.
[19] OMG, 'Software & Systems Process Engineering Meta-Model Specification v.2.0.' 2008.
[20] L. Liu, 'Security and Privacy Requirements Engineering Revisited in the Big Data Era', in *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, 2016, pp. 55–55.
[21] G. Park, L. Chung, L. Zhao, and S. Supakkul, 'A Goal-Oriented Big Data Analytics Framework for Aligning with Business', in *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*, 2017, pp. 31–40.
[22] ISO 31000, 'ISO 31000:2018 Risk Management', ISO, 2018.
[23] ENISA, 'Big Data Threat Landscape and Good Practice Guide', Jan-2016. [Online]. Available: https://www.enisa.europa.eu/publications/bigdata-threat-landscape/at_download/fullReport. [Accessed: 18-Oct-2017].
[24] A. V. Uzunov, E. B. Fernandez, and K. Falkner, 'A Comprehensive Pattern-Driven Security Methodology for Distributed Systems', in *2014 23rd Australian Software Engineering Conference*, 2014, pp. 142–151.
[25] K. Hashizume, N. Yoshioka, and E. B. Fernandez, 'Misuse Patterns for Cloud Computing', in *Proceedings of the 2Nd Asian Conference on Pattern Languages of Programs*, New York, NY, USA, 2011, pp. 12:1–12:6.
[26] C. Diamantini, P. L. Giudice, L. Musarella, D. Potena, E. Storti, and D. Ursino, 'A new metadata model to uniformly handle heterogeneous data lake sources', *Communications in Computer and Information Science*, vol. 909, pp. 165–177, 2018.
[27] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, 'Big data analytics: a survey', *Journal of Big Data*, vol. 2, no. 1, p. 21, Oct. 2015.
[28] 'Formula to Calculate HDFS nodes storage', *Hadoop Online Tutorials*, 26-Feb-2015. [Online]. Available: http://hadooptutorial.info/formula-to-calculate-hdfs-nodes-storage/. [Accessed: 26-Nov-2018].
[29] D. Steenken, S. Voß, and R. Stahlbock, 'Container terminal operation and operations research - a classification and literature review', *OR Spectrum*, vol. 26, no. 1, pp. 3–49, Jan. 2004.

Mitigación de amenazas a la privacidad en OpenID Connect mediante la introducción de un Privacy Arbiter

Jorge Navas
Universidad Rey Juan Carlos
Madrid
j.navas@alumnos.urjc.es

Marta Beltrán
Universidad Rey Juan Carlos
Madrid
marta.beltran@urjc.es

***Resumen*—Las soluciones de Gestión Federada de Identidad están siendo adoptadas ampliamente en entornos móviles, web y cloud en los últimos años, y lo serán en el futuro en entornos como Internet of Things o Edge Computing. Empresas como Google, Facebook, Amazon, Linkedin, Microsoft o Salesforce, por mencionar algún ejemplo significativo, han apoyado la creación de estándares como OAuth u OpenID Connect convirtiéndose en muchos casos en proveedores de identidad. De esta manera resuelven los problemas de Identificación, Autenticación, Autorización y/o Auditoria (IAAA) de los usuarios finales en un sólo flujo. Sin embargo, las especificaciones de OpenID Connect no se encuentran exentas de amenazas en cuanto a privacidad: el proveedor de identidades almacena información sobre los usuarios y sus atributos y registra las peticiones de acceso que van realizando con sus identidades. Este trabajo en desarrollo propone la introducción de un nuevo agente en los flujos, el árbitro de privacidad, que mitigue o evite estas amenazas.**

***Index Terms*—Gestión de identidades y accesos, OpenID Connect, Privacidad**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La Gestión Federada de Identidad (Federated Identity Management ó FIM) permite a los usuarios acceder a diferentes recursos, aplicaciones y servicios (Relying Parties ó RP) sin necesidad de crear cuentas locales, sino a través de un tercero que realiza la función de proveedor de identidades (Identity Provider ó IdP) dentro de una federación en la que previamente se han establecido relaciones de confianza. Investigaciones como [1] estiman que en la actualidad, alrededor de 60.000 webs y apps soportan diferentes versiones de OpenID Connect, utilizando Google, Facebook y Twitter como principales IdPs. Estas cifras crecen cada día, por lo que es necesario plantearse qué amenazas para la privacidad de los usuarios implican este tipo de soluciones y cómo pueden mitigarse o evitarse.

II. Contexto y punto de partida

La especificación de OpenID Connect [2] propone flujos federados de autenticación y autorización de un usuario. Cuando este usuario solicita realizar un acceso en la RP, ésta prepara una petición de autenticación y la envía al IdP en el que el usuario está registrado. El IdP autentica al usuario final (o simplemente le pide su consentimiento para realizar este nuevo flujo si el usuario ya había iniciado sesión en el IdP previamente). Si el usuario se autentica correctamente en el IdP (o simplemente consiente), el IdP enviará un Código de Autenticación (si el flujo utilizado es del tipo "Authorization Code") a la RP. La RP intercambiará directamente con el IdP este código por dos *tokens*: el ID token y el Access token. Este último puede ser intercambiado por información adicional del usuario final en caso de ser necesaria (por ejemplo, para auto-rellenar un formulario). En caso de que el flujo de autenticación sea del tipo "Implicit Flow", los *tokens* son re-dirigidos a la RP directamente a través del navegador del usuario.

Bastante trabajos previos han analizado las amenazas que OpenID Connect supone para la seguridad, algunos de ellos analizan también potenciales impactos para la privacidad de los usuarios ([3], [4] o [5]). Sin embargo, no se ha realizado hasta el momento un modelo formal de amenazas para la privacidad que sirva como punto de partida para proponer diferentes tipos de soluciones y mitigaciones más allá del tradicional cifrado en diferentes puntos de los flujos de IAAA.

III. Amenazas para la privacidad que implica el uso de OpenID Connect

Nuestro modelo de amenazas para la privacidad ha identificado cinco en relación con la PII (Personally Identifiable Information) de los usuarios de esta especificación [6]:

- **Pérdida de control del usuario final sobre la PII solicitada:** Una vez que el IdP (durante el registro o enrolment) o la RP (en cada acceso) han obtenido información del usuario final, éste no tiene manera de administrar cómo se almacena, utiliza, analiza, etc.
- **Falta de transparencia en el intercambio de PII:** La RP y el IdP tienen la capacidad de compartir PII con terceras partes sin el conocimiento/consentimiento explícito del usuario final.
- **Filtración de PII:** Brechas de seguridad en diferentes puntos del flujo de IAAA pueden producir la revelación no intencionada de información sensible que no se encuentra cifrada a terceras partes y agentes maliciosos.
- **Profiling del usuario:** La RP, y principalmente el IdP tienen la capacidad de registrar las actividades de los usuarios finales y correlacionarlas con otras fuentes de información intentando comprender, modelar y predecir como comportamientos, hábitos, intereses, etc.
- **Rastreo de la localización:** De nuevo la RP y el IdP tienen la capacidad de conocer y registrar la localización

física de los usuarios a través de sus dispositivos móviles o portátiles u ordenadores.

IV. Privacy Arbiter

En la actualidad estamos trabajando en especificar, diseñar e implementar un árbitro de privacidad (Privacy Arbiter o PA) y en incluirlo en los flujos de IAAA realizados siguiendo la especificación de OpenID Connect. Esta nueva entidad (que no reside en el IdP ni el RP, implica introducir un nuevo tipo de agente en las federaciones) permite evitar y/o mitigar, principalmente, las tres primeras amenazas mencionadas en la sección anterior. Combinada con el cifrado de la PII en el IdP, que permite evitar y/o mitigar las dos amenazas restantes, puede dar una solución muy completa a los problemas de privacidad que plantea el uso de OpenID Connect.

Arquitectura: El Privacy Arbiter es un tercero, un nuevo proveedor de servicios que se incorpora a los flujos de IAAA en OpenID Connect cuando los usuarios exigen que se respete su privacidad en mayor medida. Su rol no puede ser doble en ningún caso (no puede ser al mismo tiempo RP ni IdP) y está compuesto, fundamentalmente, por dos módulos: el Privacy Advisor y el Validation Service. Además, el UserInfo Endpoint que tradicionalmente se incluía en el IdP (para proporcionar información adicional del usuario final que complete a la que se incluye en los *claims* del ID token) debe estar ubicado ahora en el Privacy Arbiter, ya que el IdP no tomará ninguna decisión acerca de cómo se comparte PII del usuario final. Para utilizar los servicios de un Privacy Arbiter el usuario final necesitará instalar un *plugin* en su navegador o una *app* (si se trata de accesos realizados desde dispositivos móviles).

Funcionalidades: La principal función del Privacy Advisor es aconsejar al usuario qué información debe compartir con una determinada RP y qué cuenta/pseudónimo/*persona* emplear para realizar una determinada interacción. Para ello, el Privacy Advisor tiene en cuenta las preferencias del usuario (que se recogen durante la fase de enrolment y se pueden actualizar posteriormente) y sus requisitos de privacidad así como sistemas de reputación de las RPs y métricas de riesgo/confianza. Estos consejos se pueden tomar como tal, dejando al usuario la decisión final o se pueden tomar como de obligado cumplimiento para evitar que el usuario tenga que tomar constantemente este tipo de decisiones complejas. El Validation Service/ UserInfo Endpoint se encarga de sus funciones tradicionales (las que tenía en el IdP) pero también permite validar atributos del usuario, como su dirección de email o su número de teléfono, que si se cifran en el IdP este no puede validar (tal y como se especifica en la versión actual de OpenID Connect en la que el cifrado no se contempla). El Privacy Arbiter, por último, permite a un usuario saber en todo momento qué información se ha compartido con qué RP, cómo y cuándo, ya que registra esta información para que sea consultada cuando sea necesario.

Cifrado de PII en el IdP: Como se ha explicado antes, este medida complementa el uso del Privacy Arbiter. Cuando el usuario se registra en el IdP, sólo deja en claro los atributos que son imprescindibles para el funcionamiento del esquema. El resto, como pueden ser número de teléfono, dirección postal o de email, números de tarjetas de crédito, DNI, fotografía, etc. se cifran con su clave privada (el *plugin* o *app*

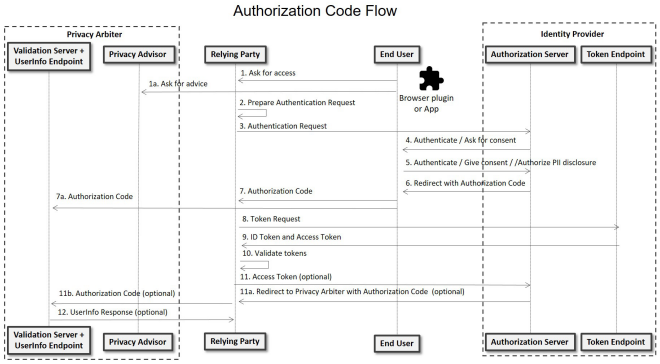


Figura 1. Flujo de OpenID Connect modificado para trabajar con el Privacy Arbiter

antes mencionados se encargan de ello utilizando primitivas software o algún hardware que siga las especificaciones de la FIDO Alliance o del Trusted Computing Group). Dependiendo de lo que aconseje compartir el Privacy Arbiter con cada RP en cada acceso, esta información será descifrada por el usuario y cifrada de nuevo, en este caso con la clave pública de esa RP (que se habrá proporcionado durante su enrolment en el IdP) para que sólo esa RP pueda recuperar la información con la correspondiente clave privada.

Flujo de IAAA modificado: En la figura 1 se observa cómo se realiza el flujo de IAAA con OpenID Connect si se utiliza la figura del Privacy Arbiter (en el caso de un flujo del tipo ”Authorization Code” en el que los tokens no pasan por el navegador del usuario). Los pasos que incorporan una letra (1a, 7a, etc.) son los nuevos, el resto se mantienen de la especificación original de OpenID Connect.

V. Conclusiones

Hasta el momento se ha propuesto un modelo formal de amenazas para la privacidad cuando se utiliza OpenID Connect. Se ha trabajado en proponer un conjunto de soluciones que permitan evitar o mitigar estas amenazas. Este artículo describe nuestros avances en la especificación, diseño e implementación de una de ellas, el Privacy Arbiter. De momento tenemos un primer prototipo, actualmente estamos trabajando en los sistemas de reputación de RPs y en la medida del riesgo/confianza para cada nueva petición de acceso.

Referencias

[1] M. Ghasemisharif, A. Ramesh, S. Checkoway, C. Kanich and J. Polakis: ”Single sign-off, where art thou? an empirical analysis of single sign-on account hijacking and session management on the web”, en *Proceedings of the 27th USENIX Security Symposium*, pp. 1475-1492, 2018.

[2] OIDF (2014). OpenID Connect 1.0. http://openid.net/connect/

[3] J. Werner, C. M. Westphall: ”A model for identity management with privacy in the cloud”, en *Proceedings of the Symposium on Computers and Communication*, pp. 463-468, 2016.

[4] H. Halpin: ”NEXTLEAP: Decentralizing identity with privacy for secure messaging”, en *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017.

[5] R. Weingärtner, C. M. Westphall: ”A design towards personally identifiable information control and awareness in OpenID Connect identity providers”, en *2017 IEEE International Conference on Computer and Information Technology*, pp. 37-46, 2017.

[6] J. Navas, M. Beltrán: ”Understanding and mitigating OpenID Connect threats”, en *Computers & Security*, 84:1-16, 2019.

Extended Abstract: Are You Sure They Are the Same? Identifying Differences Between iOS and Android Implementations

Daniel Domínguez-Álvarez[†‡], Alessandra Gorla[†], Juan Caballero[†], and Roberto Giacobazzi[‡†]
[†]*IMDEA Software Institute* [‡]*University of Verona*
Madrid, Spain *Verona, Italy*

***Abstract*—Most mobile applications are available for multiple platforms, most often Android and iOS since they jointly cover nearly the entire market. While the functionality of the Android and iOS implementations of an application may be expected to be the same, in reality they may differ significantly due to misalignments during the application development process.**

This extended abstract presents an ongoing project whose goal is to identify the differences, in terms of functionality and security offered to the user, of the Android and iOS implementations of a mobile application. Our current approach focuses on differences in the network traffic. Our preliminary results show that some security functionality may be implemented in only one of the two platforms. In a extreme case, one application encrypts its network traffic in Android, but not in iOS. Other applications only implement TLS pinning on Android and may only check it in some parts of the application.

***Index Terms*—Mobile security, privacy, iOS, Android**

Tipo de contribución: *Investigación en desarrollo*

I. Introduction

Most existing research on mobile security and privacy focuses on the Android platform. One reason for this is that Android has the largest share of the mobile market (over 80% in 2018 [1]). Another reason is that the Android ecosystem is more open than those of other platforms like iOS. With Android, researchers can easily download apps from the official store, and there exist a large number of openly available analysis tools that researchers can easily use out of the box and build upon.

Despite its smaller market share, most publishers offer their apps for the iOS platform as well. Due to the development effort required for each platform, there may be separate development teams for different implementations of the same app, e.g., one team for Android and another for iOS. This may introduce a misalignment in the app development process, leading to different features offered by the same app in two different platforms. This problem can be addressed using app development frameworks that allow mobile app developers to produce releases compatible with multiple platforms including Android and iOS [2], [3]. Using these frameworks would entail that the functionality of the app across different platforms would be aligned. However, because of the many disadvantages that these frameworks have (e.g. poor performance of the produced app and poor look and feel of the app interface), most developers prefer to develop native apps.

In this extended abstract we present our ongoing research, which aims to identify the main security and privacy implications due to having separate teams potentially implementing the same mobile app for different platforms. We focus on the Android and iOS platforms because they jointly cover nearly the entire market.

Our approach is based on differential testing. We first collect the latest versions of an app from the official Google Play and Apple App stores. Then, we compare both implementations in order to identify their security-related differences. Currently, we focus on comparing the network traffic generated by both implementations. By monitoring the network traffic we find applications that implement security functionality only in one platform, but not in the other. In a extreme case, an application only encrypts its network traffic in Android, but not in iOS. Other applications implement TLS pinning only in Android and may only check it in some parts of the application.

II. Dataset and Methodology

We have collected a small dataset of 15 apps from 3 categories that require Internet connectivity: News, Entertainment and Shopping. From each of these categories, we selected the top free apps for the Spanish market in the Apple App store. Then, we searched for those apps in the Google Play store and kept only apps that existed and had high ranking in both stores. We downloaded all apps from both stores on March 28th, 2019. Table I summarizes the 15 apps in our dataset. When needed, we have created accounts to use the apps.

Table I
The list of apps used in our preliminary study.

News	Entertainment	Shopping
El Mundo	Netflix	Amazon
El País	HBO	Wallapop
Reddit	Amazon Prime Video	Milanuncios
Twitter	MiTele	Aliexpress
Activo2	Juasapp	Wish

With this dataset we proceed to analyze the traffic that both implementations generate. For intercepting the traffic, we setup a HTTP and HTTPS proxy that our test devices use to access the Internet. In order for the HTTPS proxy to work, we create a self-signed certificate that we install as a trusted certificate in the devices. The proxy generates HTTPS certificates on the fly using our self-signed certificate as a root CA. With this setup, we can intercept the traffic of any app that trusts the device's keystore. If an app implements any kind of public key or certificate check (i.e., TLS pinning), it would be protected against our traffic analysis. While this limits the

applications we can analyze, it also enables us to identify applications that implement TLS pinning in one platform, but not in the other.

We generate traffic by running an app three times in each platform, for a total of 90 runs. In each run, we leave the application idle for the first minute. After the first minute, we interact with the application for up to two additional minutes. During this time we try to trigger network traffic by scrolling through listings, trying to request specific objects from the server and by logging into the app. For consistency, we execute the same actions in different runs of the same app.

III. Preliminary Results

Figure 1 shows the results of our analysis. Each figure plots the number of requests seen per second by the proxy while each application was being monitored. We installed and ran each application individually. To remove the noise caused by the underlying operating system, we filtered the traffic by user-agent and domain name, keeping only requests generated by the app under analysis.

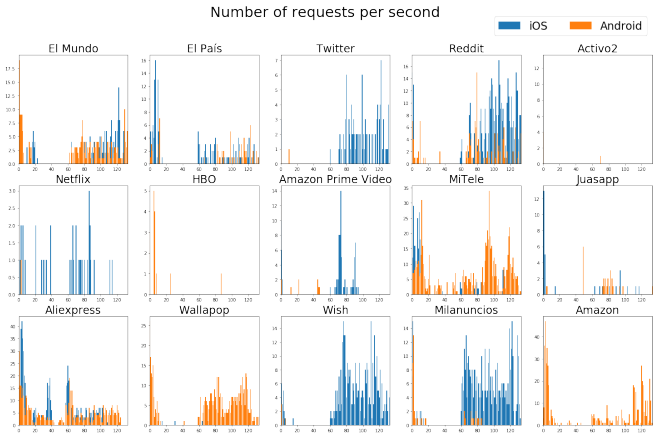


Figure 1. Network traffic generated by each app. On the Y axis we plot the number of requests, and on the X axis time in seconds.

Figure 1 shows that Activo2 and HBO have no traffic in both platforms. The only traffic for HBO is the request of the app to install GooglePlayServices. This indicates that the developers have implemented TLS pinning in both platforms. The checks run as soon as the application starts, preventing the app to work with our proxy. For several applications we observe traffic in one platform, but no traffic in the other one, indicating differences in TLS pinning support across platforms. Some apps lack TLS pinning in iOS, but are protected in Android. Some examples are Twitter, Netflix, Wish and Milanuncios. Other apps like Amazon or Wallapop show the opposite behavior, being vulnerable in Android and protected in iOS.

Some apps showed a very fine grained TLS implementation, probably coupled to a micro-services architecture in the back-end. These apps seem to implement a default TLS policy in non-critical parts of the app, while the more critical parts, such as logins, are secured. The apps that showed this behavior tend to do the same in both platforms. This is to be expected, since developers have put a huge effort in securing each component individually.

An extreme case of missing security functionality in one implementation is the Juasapp app, an application for making prank calls. The app loads information from the server using HTTPS in the Android implementation, but uses (unencrypted) HTTP for the same purpose in the iOS implementation.

IV. Work Plan

We plan to continue our study along several lines. First and foremost we want to have a statistically significant empirical study with several thousands apps. This means automating the execution of each app, which may not be trivial since the same app on two different platforms may differ significantly in the UI and in the functionalities offered. Moreover, existing UI test input generation tools still have significant limitations [4].

We plan to expand our analysis and look for more TLS bugs, similarly to [5], [6], [7]. However we plan to focus on the differences between the iOS and Android platforms.

So far we did not analyze differences in protocols and in the data sent to the server in each platform, but we plan to do it in the near future. There are several challenges that we foresee for this analysis. First of all we would need to properly align the network traffic generated in different executions, and to identify security-relevant data in the network traffic such as usernames, passwords, location coordinates, and phone numbers.

Our longer term plan is to further look into the app implementations with both static and dynamic analyses. This would allow us to have a better understanding of the security checks that are in place and that may be missing in one implemenation. This study is quite challenging, as it requires separate infrastructures for the Android and iOS devices.

Acknowledgments

This work was supported by the italian project ATEN by Fondazione Cariverona, by the Spanish Government through the SCUM grant RTI2018-102043-B-I00 and through the project DEDETIS, and by the Madrid Regional projects N-Greens Software (n. S2013/ICE-2731), BLOQUES and MadridFlightOnChip.

References

[1] "Mobile os market share 2018 — statista," https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/, (Accessed on 04/03/2019).

[2] I. Dalmasso, S. K. Datta, C. Bonnet, and N. Nikaein, "Survey, comparison and evaluation of cross platform mobile application development tools," in *IWCMC 2013*, 2013.

[3] N. Boushehrinejadmoradi, V. Ganapathy, S. Nagarakatte, and L. Iftode, "Testing cross-platform mobile app development frameworks," in *ASE 2015*, 2015, pp. 441–451.

[4] S. R. Choudhary, A. Gorla, and A. Orso, "Automated test input generation for android: Are we there yet?" in *ASE 2015*. IEEE Computer Society, 2015, pp. 429–440.

[5] L. Onwuzurike and E. De Cristofaro, "Danger is my middle name: experimenting with ssl vulnerabilities in android apps," in *WiSec 2015*, 2015, p. 15.

[6] S. Fahl, M. Harbach, T. Muders, L. Baumgärtner, B. Freisleben, and M. Smith, "Why eve and mallory love android: An analysis of android ssl (in) security," in *CCS 2012*, 2012, pp. 50–61.

[7] D. Sounthiraraj, J. Sahs, G. Greenwood, Z. Lin, and L. Khan, "Smv-hunter: Large scale, automated detection of ssl/tls man-in-the-middle vulnerabilities in android apps," in *NDSS 2014*, 2014.

Ciberseguridad en entornos de generación eléctrica en parques renovables. Resumen extendido

Antonio J. Estepa Alonso
AICIA-Universidad de Sevilla
C./de los Descubrimientos s/n
41092 Sevilla
aestepa@us.es

Jesús E. Díaz Verdejo
AICIA-Universidad de Granada
Period. Daniel Saucedo Aranda
18071 Granada
jedv@ugr.es

Estefanía de Osma Ramírez
Isotrol S.L.
C. Isaac Newton. Ed. Bluenet.
41092 Sevilla
edeosma@isotrol.com

Rafael M. Estepa Alonso
AICIA-Universidad de Sevilla
C./de los Descubrimientos s/n
rafaestepa@us.es

Germán Madinabeitia Luque
AICIA-Universidad de Sevilla
C./de los Descubrimientos s/n
german@us.es

Agustín W. Lara Romero
AICIA
C./de los Descubrimientos s/n
aguwala@gmail.com

Resumen- **Este documento presenta un proyecto en curso en el marco de ciberseguridad en entornos industriales de generación eléctrica. Por limitaciones de espacio y por motivos de confidencialidad, tan sólo se describirá el contexto de este proyecto, el alcance esperado y los requisitos que debe cumplir la solución de ciberseguridad. Por último se realiza una breve introducción al diseño inicial de la solución propuesta siguiendo la aproximación de Mínimo Producto Viable. Dicha solución se basa en la definición de Indicadores de Compromiso IoC para la detección anomalías y vulnerabilidades en la planta.**

***Index Terms*- ciberseguridad, SmartGrids, SCADAs, detección anomalías ICS**

Tipo de contribución: *Investigación en desarrollo con Industria*

I. Introducción y Objetivo de proyecto

El ámbito de la ciberseguridad en los sistemas de control industrial (Industrial Control Systems o ICS) resulta extenso y heterogéneo, contando con multitud de publicaciones que recogen las singularidades del mundo ICS frente al mundo IT (p.ej. NIST 800-82, ISA62443, IEC61508) así como las particularidades de cada sector tal y como se analiza en [1]. Este trabajo se centra en sector eléctrico y, en particular, en la monitorización de plantas de generación de energía renovable.

A diario operan miles de plantas de generación de energía renovable susceptibles de sufrir ataques. El uso de TCP/IP en los sistemas *scada* modernos ha provocado que, además de amenazas internas, éstos sean susceptibles de ser atacados con *exploit* tradicionales de sistemas operativos y DDoS. Los sistemas actuales suelen además ofrecer una interfaz web al usuario para el control y operación remota, lo que les puede hacer vulnerables a ataques comunes de aplicaciones web, máxime teniendo en cuenta que la vida útil de los parques renovables es mayor a 10 años.

Para hacer frente a estas amenazas se cuenta con recomendaciones específicas del sector eléctrico (p.e. NERC CIP 02-09, NIST7628, ISO/IEC 27019, NIST.IR7628) y guías de securización de los sistemas scada (p.e. IEC 62351-3-7, IEEE1686). También, desde el ámbito científico se sugieren contramedidas como el uso de técnicas de detección de ataques con herramientas pasivas, a fin de no interferir en los procesos operativos (aunque parece probado que la eficiencia en la detección mejora con el uso adicional de técnicas activas) o la consideración de la semántica del nivel de aplicación en base a eventos de red o características del tráfico.

Isotrol, como empresa de ingeniería y consultoría especializada en sistemas de supervisión y control con más de 30 años de experiencia en el sector energético, detecta la necesidad de abordar este proyecto de cara a afrontar el reto de la ciberseguridad en su sistema Bluence. Bluence es una solución software diseñada por la compañía Isotrol para la gestión de grandes volúmenes de información de plantas de energías renovables. Esta solución se encuentra presente en más de 1.782 plantas y se utiliza en la operación de más de 30 GW de potencia eléctrica en todo el mundo. El presente proyecto abordará la definición de un sistema para mejorar la capacidad de detección de incidentes de ciberseguridad en esta solución, teniendo en cuenta tanto las especificidades propias del sector como los condicionantes propios de la instalación y operación de un producto ajeno al operador en las redes de cada planta.

II. Descripción del escenario a proteger

En la Fig. 1 se muestra una instalación típica que tomaremos como referencia. En ella se puede apreciar el Centro de Control (CC) de un portfolio de plantas de generación renovable. El sistema ofrece una interfaz web al usuario que permite la monitorización y control en tiempo real de dichos parques. Entre otras funciones, esta solución permite visualizar los datos de producción de cada planta, gestionar alertas, análisis e informes, etc. Para hacer posible dicha monitorización y control, es necesario que el CC se conecte con el parque a través del SCADA de planta o a través de sistemas de recolección/envío de datos en tiempo real (RTU). El SCADA o RTU instalado en planta tiene conexión directa con los elementos de parque. Por otro lado, estos sistemas de recolección de datos en parque ofrecen a su vez diferentes funcionalidades que han de ser analizadas desde el punto de vista de la ciberseguridad, como pueden ser la base de datos de almacenamiento temporal (DB), la interfaz (HMI) basada en una aplicación web, un controlador de planta (PPC) o incluso el mismo hardware y sistema operativo sobre el que se soporta la solución. Normalmente, la aplicación web puede ser accedida directamente desde los operadores del Centro de Operaciones (OC) 24x7 y del CC que gestionan las alarmas del sistema o bien desde cualquier navegador de Internet con el control de acceso e intermediación de un *proxy*.

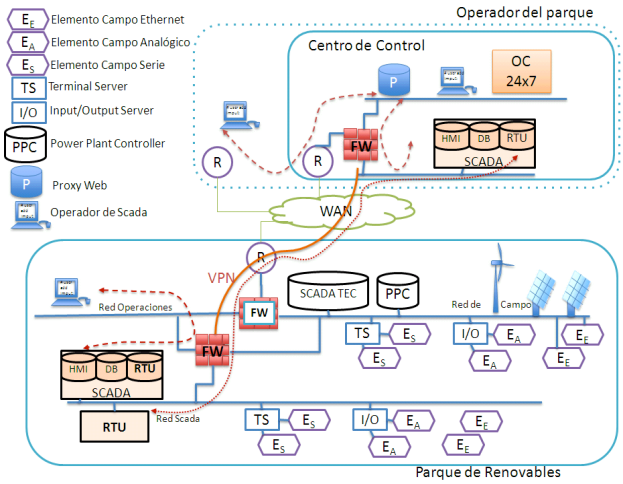


Fig. 1. Esquema de red de los elementos de un parque.

Existe un amplio abanico de protocolos y canales para establecer la comunicación de este tipo de soluciones en los parques de generación renovable, dependiendo de las casuísticas particulares de cada planta. En ocasiones, cuando los elementos de campo ya disponen de su propio sistema SCADA local (*Scada Tec*), la solución SCADA instalada en planta establece un conducto con el protocolo OPC para la comunicación con éste. En otros casos, el SCADA se comunica con los elementos de campo, que suelen estar accesibles directamente en una red Ethernet de campo (E_E), mediante servidores de terminales (E_S) o mediante servidores de entrada/salida que permiten conexión de entradas y salidas analógicas y digitales (E_A), y pueden usar diversos protocolos de comunicación (p.ej. IEC 60870-5-101/104, ModBus). A su vez, el SCADA instalado en la planta suele establecer un conducto de datos IEC-104 con el SCADA en el CC para permitir la gestión remota –además de local– a través del HMI. Entre el CC y el SCADA o RTU instalado, las comunicaciones están cifradas mediante una VPN con IPsec convenientemente securizada.

III. Propuesta Inicial del Sistema de Detección

A. *Alcance y restricciones del sistema*

El alcance del sistema debe cubrir la detección de ataques al SCADA (o RTU), fijando como objetivo principal la generación de alarmas que deben ser integradas con el sistema de monitorización del OC existente. Queda, por tanto, fuera del alcance la protección de los elementos de campo y red de operaciones, así como sus comunicaciones.

Los principales condicionantes de diseño de la solución son: a) no interferir con los sistemas existentes, b) minimizar el uso la red WAN, y c) bajo coste: el nuevo dispositivo deberá estar basado en herramientas de software libre así como minimizar la necesidad de memoria y procesamiento.

B. *Diseño preliminar del sistema*

Se plantea el desarrollo de un sistema SIEM que creará alertas priorizadas según del nivel de riesgo percibido e integradas en los paneles de monitorización del OC.

El sistema procesará dos entradas:

(i) los paquetes recibidos por el FW (reenviados usando *port-mirroring*);

(ii) las peticiones recibidas por el HMI del scada (e.g. log del servidor web de la aplicación). Se plantea seguir la metodología de Mínimo Producto Viable. Para ello se definen 3 fases o refinamientos que pueden dar lugar así a 3 versiones del sistema con niveles de exigencia computacional y desempeño diferenciados en función de los requisitos de cada instalación.

- Fase 1: la entrada (i) es procesada para generar flujos IPFIX que podrán ser enriquecidos con información (local) de geolocalización y reputación. Estos flujos IPFIX son utilizados para: identificación y clasificación de activos, y para generar una matriz de tráfico de la que se extraerán un conjunto de métricas de importancia y exposición de los activos así como indicadores de compromiso (IoC) para detectar comportamientos anómalos (equipos infectados y tipos de ataques).
- Fase 2: se amplía la detección de ataques a través del uso adicional de dos IDS: uno basado en firmas y alimentado por la entrada (i) y otro basado en anomalías alimentado con la entrada (ii).
- Fase 3: se actualizan las vulnerabilidades de los activos a través de técnicas activas de *scanning* y *fingerprinting* que enriquecen la información sobre los activos y así mejoran el cálculo del riesgo asociado a cada alarma.

Las alertas generadas en cualquiera de las fases son puestas a disposición de un módulo de correlación y evaluación de riesgos, que ayudará al SIEM a gestionar y priorizar las alarmas generadas en función del riesgo. Todas las alarmas serán enviadas al SOC mediante la VPN. El SOC deberá a su vez configurar el sistema descrito a fin de reducir los falsos positivos y las alarmas con baja criticidad.

IV. Conclusiones

La protección de sistemas en entornos industriales supone un reto debido a los condicionantes operativos y económicos en cada instalación. Sin embargo es crucial disponer de soluciones que protejan y prevengan de ataques a las infraestructuras críticas como son en este caso las plantas de generación renovable. Este trabajo preliminar aporta las ideas clave que guiarán el diseño de una solución en seguridad flexible y alineada con dichos condicionantes.

Agradecimientos

Este proyecto está siendo desarrollado por Isotrol y AICIA (PI-1814/26/2018) bajo el marco de la financiación del Centro para el Desarrollo Tecnológico Industrial (CDTI), dentro del Programa Estratégico de Consorcios de Investigación Empresarial Nacional (CIEN) 2017.

Referencias

[1] Knowles, W., Prince, D., Hutchison, D., Disso, J. F. P., & Jones, K.: " A survey of cyber security management in industrial control systems", en International journal of critical infrastructure protection n.9 y, pp. 52-80, 2015.

¿Cómo representar un Buffer Overflow? Una revisión literaria sobre sus características

Gonzalo Esteban, Razvan Raducu, Ángel Manuel Guerrero, Camino Fernández
Grupo de Robótica. Universidad de León
Av. Jesuitas, s/n. 24007 León (Spain)
{gestc, rrad, am.guerrero, camino.fernandez}@unileon.es

***Resumen*—Detectar vulnerabilidades como los Buffer Overflows generalmente implica el uso de herramientas de análisis de código fuente o de revisiones manuales por parte de expertos. En este campo, la inclusión de técnicas como el aprendizaje automático pretende mejorar dicho proceso abriendo la puerta a la predicción de vulnerabilidades. A grandes rasgos, tanto las herramientas de análisis de código como las técnicas de aprendizaje automático funcionan representando ciertas características del código fuente. Sin embargo, en la literatura académica no se ha abordado la cuestión de cuántas formas de representar un Buffer Overflow existen. Con el objetivo de cubrir tal carencia, este trabajo presenta una revisión sistemática de la literatura. Fruto de ello ha sido la identificación de 79 características, todas ellas recogidas en 8 representaciones distintas. Estos resultados podrían asentar las bases para futuras investigaciones en el campo de la predicción de Buffer Overflows. En concreto, tanto para crear nuevos conjuntos de características a partir de las ya identificadas en la literatura, como para evaluar o comparar si los conjuntos encontrados son efectivos a la hora de predecir y representar este tipo de vulnerabilidades.**

***Index Terms*—Buffer Overflow, Características, Análisis estático, Aprendizaje automático, Revisión literaria**

Tipo de contribución: *Investigación original (límite 8 páginas)*

I. Introducción

Dada su peligrosidad y facilidad de explotación, los *buffer overflow* (BOF) han sido objeto de interés en el campo de la seguridad informática durante las últimas décadas [1]. Incluso en la actualidad esta vulnerabilidad sigue siendo relevante, ocupando el tercer puesto en el ranking de los 25 errores de software más peligrosos del CWE/SANS [2]. Quizás esto se deba a que, por ahora, la única forma de evitar por completo un BOF es utilizar un lenguaje de programación que gestione de forma adecuada los accesos a memoria [3]; aunque generalmente dicha opción no sea viable. Mientras tanto las soluciones para defenderse frente a este tipo de problemas pasan por analizar el código fuente o los binarios de un programa [4]. No obstante, detectar un BOF es difícil pues depende de la complejidad del ataque que se haya llevado a cabo [5]. Así, utilizar herramientas de análisis automático [6], [5], [7], [3] o auditar manualmente el código fuente con expertos [8], [9], son algunos ejemplos de las alternativas existentes.

Ante esta situación, y parafraseando a Padmanabhuni y Tan [7], las herramientas y soluciones actuales poseen unas limitaciones inherentes a la diversidad y complejidad del problema que pretenden resolver. Este hecho puede afectar a la hora de abordar correctamente la detección de nuevos tipos de BOFs, lo que implica la necesidad de ampliar y mejorar con el tiempo tales productos. De este modo, Padmanabhuni y Tan sugieren que la investigación en este ámbito debería encaminarse hacia unas metodologías complementarias basadas en adquirir conocimiento de forma dinámica y extensible a través de diversas fuentes. Conforme a ello, recientemente se ha incorporado al ámbito de la detección de vulnerabilidades el aprendizaje automático (en inglés *Machine Learning*), técnica que puede llegar a predecir vulnerabilidades a partir de datos derivados del código fuente [10], [11]. Pero a pesar de que ya hay evidencias de su aplicación en la predicción de BOFs [12], [13], lo cierto es que en la literatura no se ha abordado la cuestión de cuántas formas de representar esos datos existen.

El propósito de este trabajo es revisar la literatura presente hasta el momento en el ámbito de la predicción de vulnerabilidades. En concreto se quiere conocer las diferentes formas o modelos de representar un BOF a partir de características extraídas del código fuente. Para ello se lleva a cabo una revisión sistemática de la literatura siguiendo las recomendaciones de Kitchenham, Budgen y Brereton [14]. De este modo, el trabajo plantea las siguientes preguntas de investigación:

- RQ1 ¿Cuáles son las características, extraídas a partir del código fuente, que describen un Buffer Overflow dentro del análisis y predicción de vulnerabilidades?
- RQ1.1 ¿Cuáles son las formas que utilizan los autores de agrupar dichas características con el objetivo de representar un Buffer Overflow?

Para detallar las respuestas, el resto del trabajo sigue la estructura general de la guía PRISMA sobre realizar revisiones sistemáticas [15]. La Sección II describe una breve introducción tanto de los BOFs y el aprendizaje automático como de la relación que hay entre ambos. La Sección III resume el protocolo de revisión planificado para esta revisión y la Sección IV presenta los resultados obtenidos a partir del mismo. En base a las evidencias halladas, la Sección V responde a las preguntas de investigación planteadas e identifica las limitaciones de este trabajo. Finalmente se detallan las conclusiones y aplicaciones de esta revisión en la Sección VI.

II. Contexto

En palabras de Black y Bojanova [3], un BOF sucede al utilizar la referencia de un array (buffer) con el fin de leer o escribir en una dirección de memoria fuera de los límites permitidos (overflow). Es más, los datos a los que se acceden mediante dichas operaciones son totalmente ajenos a los esperados inicialmente por el programa y contienen información para comprometer su funcionamiento. De este modo, escribir implica, en el peor de los casos, escalar privilegios o ejecutar código arbitrario; mientras que leer implica acceder

a datos residuales que podrían contener información sensible (contraseñas o información personal), [6], [4], [7], [3]. Sin embargo, y como se dijo anteriormente, detectar un BOF no es una tarea trivial.

Con respecto a las técnicas de detección automática de vulnerabilidades, estas han sido concebidas con el propósito de facilitar a los programadores la tarea de localizar vulnerabilidades [6]. Principalmente dichas técnicas se clasifican en dos: estáticas y dinámicas [7], [16], [17]. Las técnicas estáticas se aplican directamente sobre el código fuente, es decir, sin ejecutar el programa; pero suelen necesitar la intervención humana por la cantidad de falsos positivos o negativos que encuentran [18]. Por el contrario, las técnicas dinámicas se centran en la ejecución del programa y analizan sus salidas para verificar si estas son las esperadas. Ahora bien, aunque ambas técnicas tienen su ámbito de aplicación, resulta de especial interés para la predicción de vulnerabilidades conocer las formas de representar el código fuente utilizadas en el análisis estático.

A grandes rasgos, el análisis estático utiliza un conjunto conocido de patrones (o reglas) para buscar coincidencias en el código fuente [18]. En concreto, el análisis léxico, la ejecución simbólica o la verificación formal, son algunos ejemplos de dichas técnicas [16]. Todas ellas tienen en común el uso de símbolos para representar diferentes atributos del código fuente: llamadas a funciones, variables utilizadas, estructuras de flujo y de control, etc. Teniendo en cuenta esto, una situación similar sucede en el aprendizaje automático, en donde también se trabaja con representaciones de datos.

Parafraseando a Domingos [19], el aprendizaje automático es un grupo de técnicas que aprenden a clasificar (predecir) datos de forma precisa a partir de un conjunto de datos empíricos. Más aún, Domingos resume dicho aprendizaje como la unión de tres elementos: representación, evaluación y optimización. El primero se refiere a representar los datos en un lenguaje formal que un ordenador pueda entender. El segundo se corresponde con una función que evalúa cuán buena o mala es dicha representación. El último es un procedimiento para buscar cuál es la mejor representación de los datos. Estos tres elementos trabajan sobre lo que se denomina un clasificador. Como define el propio Domingos, un clasificador es un sistema que, a partir de una entrada formada por un vector de valores discretos y/o continuos (características), produce una salida compuesta por un único valor discreto (clase). Cabe señalar que el tipo de representación de los datos es tan importante que influye en el propio rendimiento del aprendizaje automático [20].

Llevando todo lo anterior al ámbito de la predicción de BOFs, buscar un conjunto de características que permita representar adecuadamente este tipo de vulnerabilidades puede ser algo beneficioso para el futuro desarrollo de este campo de investigación.

III. Metodología

Para la realización de este trabajo se ha optado por la aplicación de la metodología propuesta por Kitchenham, Budgen y Brereton [14], que sirve para realizar revisiones sistemáticas en el ámbito de la informática. En particular, el objetivo de esta revisión es obtener una visión general y rápida del estado de la literatura publicada en revistas o congresos. De este modo, la revisión está basada en cualquier tipo de artículo publicado en el que se presenten un conjunto de características para describir un BOF; eso sí, extraídas a partir de código fuente. Ello implica que dichas características pueden corresponder tanto a trabajos de detección (análisis estático) como de predicción de vulnerabilidades (aprendizaje automático). Los resultados a medir se componen de tales características y de las formas existentes de agruparlas a fin de representar, en conjunto, un BOF.

III-A. Proceso de búsqueda

El proceso se planificó en base a un conjunto de artículos de semi-referencia (ver Tabla I). Este conjunto se fundamenta en la propuesta de Zhang, Babar y Tell [21]; aunque se obtuvo de manera informal antes de la revisión. Teniendo en cuenta eso, se planificó una búsqueda automática hasta enero de 2019 en las siguientes bases de datos electrónicas: ACM Digital Library, IEEE Xplore Digital Library, Scopus y Web of Science. En todas las bases de datos se buscaron artículos de revista o congreso escritos en inglés. Además, dichas búsquedas se realizaron por los campos de título, abstract y palabras clave.

Tabla I
Conjunto de artículos de semi-referencia

Referencia	Título
[13]	Buffer Overflow Vulnerability Prediction from x86 Executables Using Static Analysis and Machine Learning
[22]	The Bugs Framework (BF): A Structured Approach to Express Bugs
[12]	Assisting in Auditing of Buffer Overflow Vulnerabilities via Machine Learning

Después se construyó una sola cadena de búsqueda para las dos preguntas (RQ1 y RQ1.1) por estar ambas relacionadas entre sí. La cadena estaba basada en las palabras clave extraídas de la pregunta de investigación RQ1 con la estrategia PICOC [14]:

- Population: buffer overflow;
- Intervention: análisis; predicción;
- Comparison: -
- Outcome: características;
- Context: -

Dichas palabras se tradujeron al inglés y se les añadieron algunos sinónimos y variantes procedentes de los artículos de semi-referencia. El resultado fue la siguiente cadena: *(“buffer overflow”) AND (“source code” OR “static code”) AND (analy* OR audit* OR predict* OR “machine learning”) AND (features OR attributes OR characteristics OR properties)*. Cabe señalar que para validar esta cadena, y por consiguiente el proceso de selección, se fijó como criterio que entre los resultados de la búsqueda automática figurasen al menos 2 de los 3 artículos del conjunto de semi-referencia (66.6 %).

Finalmente se complementó la búsqueda automática planificando una revisión de las referencias de los artículos primarios, es decir, de aquellos artículos revisados tras el proceso de selección que responden a las preguntas de investigación.

III-B. Proceso de selección

Para filtrar los resultados de las búsquedas se planificó un proceso de selección basado en el conjunto de criterios

de inclusión y exclusión de la Tabla II. Se planificó una única fase porque se esperaba que hubiera un número reducido de resultados—dada la relativa juventud del área de investigación. Además, otra razón para esta decisión fue la posibilidad de que la definición de las características fuese algo secundario con respecto a la hipótesis del artículo: leer el título y el abstract podría no ser suficiente para tomar una decisión. De este modo, primero se eliminaron aquellos artículos duplicados y luego se obtuvieron copias del resto de artículos para la revisión completa de su texto.

Tabla II
CONJUNTO DE CRITERIOS DE INCLUSIÓN Y EXCLUSIÓN

Tipo de criterio	Descripción
Inclusión	Contiene características que describen un BOF
Exclusión	No se definen características
	Las características definidas no se obtienen de forma estática
	No se ha conseguido copia del artículo

III-C. Proceso de evaluación de la calidad

Dada la planificación del proceso de selección, se optó por desarrollar un proceso de evaluación de la calidad con el objetivo de ser excluyente. En otras palabras, se asignó una puntuación de calidad a cada artículo y se estableció una puntuación mínima por debajo de la cuál descartar su selección. Para ello se planteó una lista de control con las siguientes preguntas:

1. ¿Se extraen las características a partir del código fuente y de forma estática?
2. ¿Se enumeran las características de forma clara (por ejemplo, mediante una lista, figura o tabla)?
3. ¿Se da una descripción de cada característica?
4. ¿Se indican los posibles valores nominales de cada característica?

Cada pregunta se respondió con un "Sí", "No" o "Parcialmente". Además, para cuantificar la calidad de un artículo, a dichas respuestas se les asignó unas puntuaciones de "1", "0" y "0.5" respectivamente. De este modo se fijó una puntuación máxima para cada artículo de 4 puntos y un umbral de calidad de 3.0 puntos. Así, si un artículo no superaba dicho umbral quedaría excluido de la revisión.

III-D. Procesos de extracción y síntesis de datos

Para responder a las preguntas de investigación se planificó una extracción de datos mediante un formulario electrónico—utilizando la herramienta web Parsifal [1]—, y una síntesis de datos en forma de tablas. Los datos a extraer por cada artículo fueron: referencia bibliográfica, identificador único, nombre de la característica, posibles sinónimos con características de otros artículos, conjunto de valores definidos para la característica, cardinalidad (es decir, número total de características presentadas en el artículo) y finalmente el listado de todas las características propuestas en el propio artículo (que forman la representación del BOF).

IV. RESULTADOS

El resultado de ejecutar los procesos de búsqueda y selección queda resumido en el diagrama PRISMA de la Figura 1.

[1]https://parsif.al

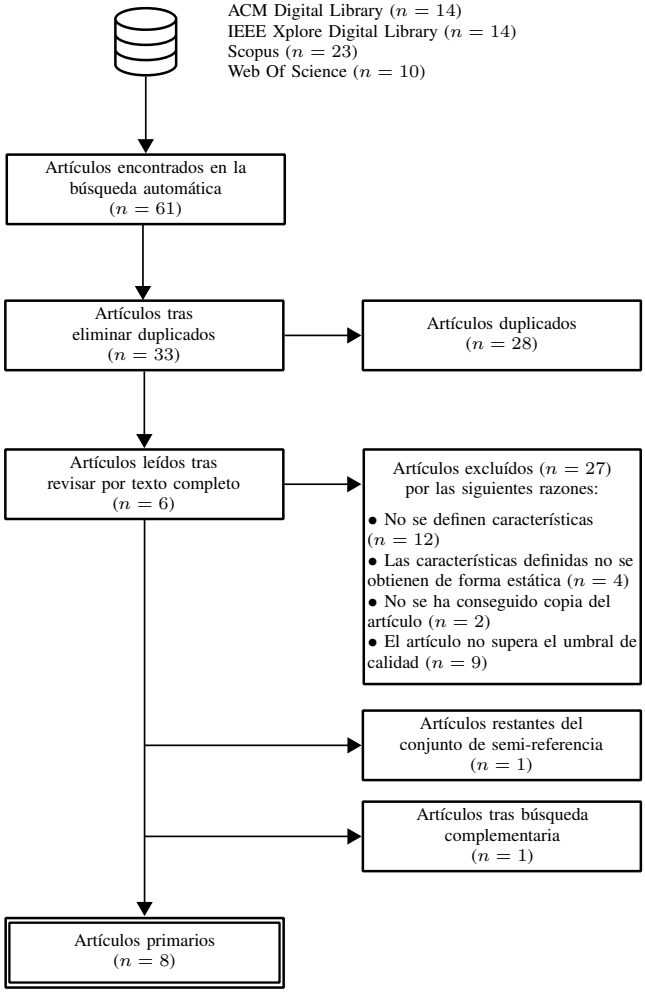


Figura 1. Diagrama PRISMA.

Se encontraron un total de 61 artículos a partir de la búsqueda automática en las fuentes de información establecidas. Además, la búsqueda quedó validada porque entre los resultados figuraban 2 de los 3 artículos pertenecientes al conjunto de semi-referencia. En lo relativo al proceso de selección, primero se eliminaron 28 artículos duplicados y luego se revisaron los 33 restantes. Posteriormente se descartaron 27 artículos: 18 por cumplir algún criterio de exclusión y otros 9 por no alcanzar el umbral de calidad. Por consiguiente, el conjunto de artículos primarios quedó formado por 6 trabajos; al que hubo que añadir el remanente de semi-referencia. A continuación se realizó una búsqueda complementaria sobre las referencias de esos 7 artículos primarios, revisando únicamente su título y abstract. Sin embargo, únicamente se encontró como relevante un trabajo de máster [23]. Al ser este un trabajo de literatura gris, se descartó su selección. A cambio se identificó a través de Google Scholar otro artículo similar publicado por la misma autora [24]. Finalmente se identificaron un total de 8 artículos primarios, que se han enumerado de forma única siguiendo la estructura S1, . . . , S8. Los artículos estan publicados entre 2004 y 2017, concentrándose en su mayoría entre 2014 y 2017 (ver Figura 2).

En cuanto a la evaluación de la calidad, la Tabla III recoge

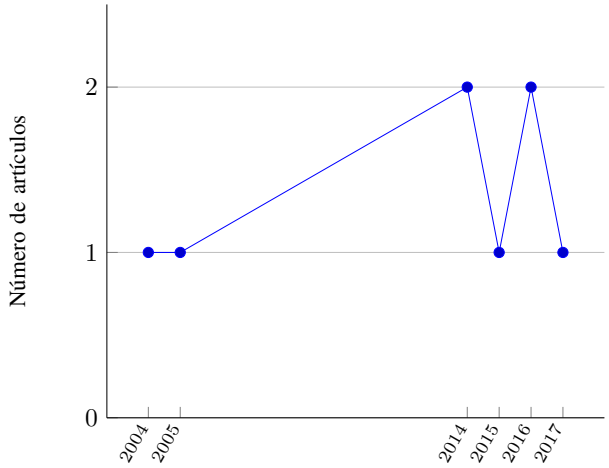


Figura 2. Gráfico de tendencia de los artículos primarios.

los resultados de estos 8 artículos primarios. La puntuación mínima fue de 3.0 (para S1) y la máxima de 4.0 (para S3, S4, S5, S6 y S8). El artículo S1 fue el único en responder con un "No" a la pregunta 3 ("¿Se da una descripción de cada característica?"). Además, los artículos S2 y S7 fueron los únicos en responder con "Parcialmente" a la pregunta 1 ("¿Se extraen las características a partir del código fuente y de forma estática?") porque contienen características que se extraen de forma dinámica.

Tabla III
RESULTADOS DE LA CALIDAD DE LOS ARTÍCULOS PRIMARIOS.

Referencia	ID trabajo	Puntuación
[25]	S1	3.0
[23]	S2	3.5
[26]	S3	4.0
[27]	S4	4.0
[13]	S5	4.0
[28]	S6	4.0
[22]	S7	3.5
[12]	S8	4.0

Por lo que se refiere a la síntesis de los datos, las Tablas IV y V recogen toda la información extraída sobre características para describir un BOF, mientras que la Tabla VI recopila las propuestas de modelos para representar un BOF.

V. DISCUSIÓN

V-A. *Pregunta de investigación RQ1*

La pregunta RQ1 pretende identificar cuántas características distintas existen para describir un BOF. Las evidencias recogidas a partir de los 8 artículos primarios muestran un total de 79 características (ver Tablas IV y V). En realidad, el número de características totales es de 175 pero muchas de ellas, bien se repiten entre los diferentes artículos, o bien utilizan sinónimos. Para determinar si una característica es sinónimo de otra, se tomó como criterio cuán semejantes eran sus definiciones. Así, por ejemplo, la característica "Location" aparece en los artículos S1, S5 y S7 pero también en S2 y S8 como "Memory location". En cuanto a los tipos de valores tomados por las propias características, cada artículo los define de forma independiente. No obstante, se puede observar que las características se agrupan de dos maneras. Por un lado, características como "Bound", "Magnitude" o "Is source a buffer" son de tipo cualitativo nominal porque sus valores oscilan dentro de un posible rango específico. Por ejemplo, en el caso de "Is source a buffer" los valores oscilan en el conjunto $\{false, true\}$. Por otro lado, características como "Loop depth", "NULL check" o "Input count" son de tipo cuantitativo discreto porque sus valores representan un contador en el intervalo $[0, +\infty)$.

Ahora bien, si se analiza en detalle la naturaleza de cada característica se podrían considerar cuatro clases; que, por otra parte, no se han recogido en una tabla por las limitaciones de espacio del propio trabajo. Por un lado están aquellas que representan aspectos presentes en la instrucción que desencadena el BOF; como por ejemplo "Type" (tipo de dato almacenado en el buffer de destino), "Access" (tipo de instrucción para acceder al buffer de destino) o "Number of elements copied within bounds" (flag para indicar si los elementos copiados al buffer de destino están dentro de los límites permitidos). También hay características relativas a la entrada de datos que acaban afectando a la instrucción vulnerable. En ese caso, algunos ejemplos son: "Command line" (número de instrucciones en las que se ha leído la línea de comandos), "Input count" (número de instrucciones de entrada de datos) o "Input dependent predicates" (número de instrucciones de entrada de datos de las que depende la ejecución de la instrucción que desencadena el BOF). Por otro lado existen características referentes a la presencia de mecanismos (instrucciones) que evitan la aparición del BOF. "String length of source buffer" (número de instrucciones en las que se comprueba la longitud de la cadena del buffer de origen), "Validation" (número de instrucciones que hacen referencia al tamaño del buffer de la instrucción vulnerable) o "Direct sanitization" (número de instrucciones que contienen un flujo de control condicional que evita el BOF) son algunos ejemplos de ello. Adicionalmente, hay un tipo de características que hace referencia a la propia existencia de un BOF, como es el caso de "Magnitude" (tamaño del BOF) o "Data size" ("cuántos datos se han escrito fuera de los límites").

Por último, otro aspecto a destacar es la existencia de características compuestas, es decir, de características formadas por otras más pequeñas; aunque los autores las consideren como una sola. Dada su naturaleza, este tipo de características no se han recogido en las Tablas IV y V, pero sí en la Tabla VI (en la que se representan en letra cursiva). En dicha tabla, algunos ejemplos de características compuestas son "Input classification", "Sink characteristics" o "Control dependency characteristics".

V-B. *Pregunta de investigación RQ1.1*

La pregunta RQ1.1 se centra en conocer los distintos modelos existentes de agrupar características para representar un BOF. Sin embargo, en ninguno de los artículos revisados se propone explícitamente una representación de BOF. Por ello se entiende como representación (o modelo) de un BOF, al conjunto de características propuesto en un determinado artículo. Así pues, las evidencias extraídas y resumidas en la Tabla VI muestran 8 representaciones diferentes; algunas de ellas compartiendo ciertas características. Para empezar, el trabajo S1 define 11 características que se utilizan como base

Tabla IV
RESUMEN DE LAS EVIDENCIAS EXTRAÍDAS SOBRE CARACTERÍSTICAS PARA REPRESENTAR BOFS.

#	Característica	Sinónimos	Valores
1	Bound (S1)	Upper/Lower Bound (S2); Boundary (S7)	S1=S2={upper; lower}; S7={below; above}
2	Type (S1)	Data Type (S2)	S1={char; unsigned char}; S2={character; integer; floating point; wide character; pointer; unsigned character; unsigned integer}
3	Location (S1,S5,S7)	Memory location (S2,S8)	S1={stack; vss; heap; data}; S2={stack; heap; data region; BSS; shared memory}; S5={global; local; mixed; heap}; S7={heap; stack}; S8={stack; heap; data segment; bss segment; shared memory}
4	Scope (S1,S2)	-	S1={inter-procedural; same function; global function}; S2={same; inter-procedural; global; inter-file/inter-procedural; inter-file/global}
5	Container (S1,S2,S8)	-	S1={none; union}; S2={no; array; struct; union; array of structs; array of unions}; S8={none; array; struct/union; others}
6	Index or Limit (S1)	Index complexity (S2); Index type (S8)	S1={none; variable; linear expression; contents of a buffer}; S2={constant; variable; linear expression; non-linear expression; function return value; array contents; N/A}; S8={constant, addition, multiplication, nonlinear, function call, array access}
7	Access (S1)	Sink classification (S3,S4,S6); Operation type (S5); Sink type (S8)	S1={C function; pointer; index; double de-reference}; S3=S4=S6={string copy; string concatenation; memory alteration; formatted string output; unformatted string input; formatted string input; array element writes}; S5={copy; concatenation; formatted write; unformatted write; array write}; S8={pointer dereference; array write; dangerous function}
8	Buffer alias (S1)	Alias of buffer address (S2)	S1={alias; no alias; alias of an alias}; S2={no; one; two}
9	Control flow (S1)	Local control flow (S2)	S1={none; if-statement; switch}; S2={none; if; switch; cond; goto/label; setjmp/longjmp; function pointer; recursion}
10	Secondary control flow (S2)	-	S2={none; if; switch; cond; goto/label; setjmp/longjmp; function pointer; recursion}
11	Surrounding loops (S1)	Loop structure (S2)	S1={none; while; for; nested}; S2={none; standard for; standard do-while; non-standard for; non-standard do-while; non-standard while}
12	Input taint (S1)	Taint (S2,S8)	S1={packet; dir function; file; argc/argv}; S2={no; argc/argv; environment variables; file read or stdin; socket; process environment}; S8={false; true}
13	Write/Read (S2)	Access (S7)	S2=S7={write; read}
14	Pointer (S2)	-	S2={no; yes}
15	Address complexity (S2)	Address type (S8)	S2={constant; variable; lineal expression; non-linear expression; function return value; array contents}; S8={constant; addition; multiplication; nonlinear; function call; array access}
16	Length/limit complexity (S2)	Length type (S8)	S2={n/a; none; constant; variable; linear expression; non-linear expression; function return value; array contents}; S8={constant; addition; multiplication; nonlinear; function call; array access}
17	Alias of Buffer index (S2)	-	S2={no;one;two;N/A}
18	Loop complexity (S2)	-	S2={n/a; none; one; two; three}
19	Asynchrony (S2)	-	S2={no; threads; forked process; signal handler}
20	Runtime environment dependence (S2)	-	S2={no; yes}
21	Magnitude (S2, S7)		S2={none; 1 byte; 8 bytes; 4096 bytes}; S7={small; moderate; far}
22	Continuous/Discrete (S2)	Reach (S7)	S2=S7={discrete; continuous}
23	Signed/Unsigned mismatch (S2)	-	S2={no; yes}
24	Data size (S7)	-	S7={little; some; huge}
25	Direct sanitization (S8)	-	S8=[0, +∞)
26	Indirect sanitization (S8)	-	S8=[0, +∞)
27	Interprocedural sanitization (S8)	-	S8=[0, +∞)
28	Loop depth (S8)	-	S8=[0, +∞)
29	Condition depth (S8)	-	S8=[0, +∞)
30	Call depth (S8)	-	S8=[0, +∞)
31	Command line (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
32	Environment variable (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
33	File (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
34	Network (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
35	String length of source buffer (S3, S4)	String length of source (S5); String length of source buffer check (S6)	S3=S4=S5=S6=[0, +∞)
36	Size of source (S3, S4, S5, S6)	-	S3=S4=S5=S6=[0, +∞)
37	NULL check (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
38	EOF check (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
39	Character check (S3, S4, S6)	-	S3=S4=S6=[0, +∞)
40	Character occurrence in string check (S3, S4, S6)	-	S3=S4=S6=[0, +∞)

Tabla V
RESUMEN DE LAS EVIDENCIAS EXTRAÍDAS SOBRE CARACTERÍSTICAS PARA REPRESENTAR BOFS (CONTINUACIÓN).

#	Característica	Sinónimos	Valores
41	String comparison (S3,S4,S6)	-	S3=S4=S6=[0, +∞)
42	Other check (S3,S4,S6)	-	S3=S4=S6=[0, +∞)
43	Size of destination buffer check (S3,S4,S6)	Size of destination (S5)	S3=S4=S6=[0, +∞)
44	Size of destination (buffer-1) check (S3,S4,S6)	-	S3=S4=S6=[0, +∞)
45	Size of destination (buffer-x) check (S3,S4,S6)	-	S3=S4=S6=[0, +∞)
46	String length of destination buffer check (S3,S6)	String length of destination buffer (S4); String length of destination (S5)	S3=S4=S5=S6=[0, +∞)
47	Post size check increment (S3,S6)	-	S3=S6=[0, +∞)
48	Data buffer declaration (S3)	Data buffer declaration statement classification (S4,S6)	S3=S6={static allocation; dynamic source independent allocation; dynamic source dependent allocation}; S4={static allocation; dynamic allocation}
49	Number of elements copied within bounds (S3,S4,S6)	-	S3=S4=S6={not applicable; false; true; unknown}
50	Array write index within bounds (S3,S4,S6)	-	S3=S4=S6={not-applicable; false; true}
51	Format string precision within bounds (S3)	-	S3={not-applicable; false; true}
52	String copy within bounds (S3,S6)	-	S3=S6={not-applicable; false; true}
53	Data dependent on destination buffer size (S3,S4,S6)	-	S3=S4=S6=[0, +∞)
54	Data dependent on destination buffer size variant (S3,S4,S6)	-	S3=S4=S6=[0, +∞)
55	Is character case conversion sink (S3,S6)	-	S3=S6={not-applicable; false; true}
56	Resets in control predicates (S3,S6)	-	S3=S6=[0, +∞)
57	Input count (S5)	-	S5=[0, +∞)
58	Inputs with limiting (S5)	-	S5=[0, +∞)
59	Environment dependent input (S5)	-	S5=[0, +∞)
60	Is source a buffer (S5)	-	S5={false; true}
61	Is source null terminated (S5)	-	S5={false; true; sometimes}
62	Has defensive limiting (S5)	-	S5={false; true; not-applicable}
63	Is destination NULL terminated (S5)	-	S5={false; true; sometimes}
64	Multiple writes destination (S5)	-	S5={false; true; sometimes}
65	Same source and destination size (S5)	-	S5={false; true; sometimes}
66	Declaration (S5)	-	S5={static; dynamic source dependent; dynamic source independent; dynamic constant size; mixed}
67	Source dependent loop termination (S5)	-	S5={false; true; not-applicable}
68	Validation (S5)	-	S5=[0, +∞)
69	Input dependent predicates (S5)	-	S5=[0, +∞)
70	Source buffer predicates (S5)	-	S5=[0, +∞)
71	Destination buffer predicates (S5)	-	S5=[0, +∞)
72	Inter procedural destination buffer access (S5)	-	S5={false; true}
73	Number of destination writes (S5)	-	S5=[0, +∞)
74	Is destination buffer ambiguous (S5)	-	S5={false; true}
75	Is source ambiguous (S5)	-	S5={false; true; not-applicable}
76	Sink dependent ambiguous containers (S5)	-	S5=[0, +∞)
77	Sink dependent non-ambiguous containers (S5)	-	S5=[0, +∞)
78	Predicates referring to ambiguous containers (S5)	-	S5=[0, +∞)
79	Predicates referring to non-ambiguous containers (S5)	-	S5=[0, +∞)

para el modelo presentado en el trabajo S2. Precisamente en S2 se presentan las mismas 22 características incluidas en el trabajo de máster descartado durante la búsqueda complementaria [23]. A su vez, esas 22 características sirven como base para los modelos presentados en S7 y S8; en cuyos casos se resumen y amplían a 6 y a 31 respectivamente. Por otro lado están los trabajos S3-S6, que pertenecen a los mismos autores. En concreto, los trabajos S3, S4 y S6 comparten prácticamente las mismas características (aunque con algunas variaciones) mientras que S5 es diferente (pues la mayoría de las características son distintas).

Llegados a este punto, cabe señalar que los modelos propuestos en los artículos S3, S4, S5, S6 y S8 están enfocados al aprendizaje automático. Cada uno de estos artículos tiene una sección de evaluación en la que se evalúa la efectividad de la representación propuesta a la hora de detectar BOFs. Particularmente comparan qué algoritmo de aprendizaje automático clasifica mejor la representación de un BOF. También se debe mencionar que casi todos los artículos (S3, S5, S6 y S8) analizan el rendimiento a la hora de detectar o predecir este tipo de vulnerabilidades; comparando el algoritmo que mejor clasifica frente a determinadas herramientas de análisis estático de

Tabla VI
RESUMEN DE LAS EVIDENCIAS EXTRAÍDAS SOBRE MODELOS PARA REPRESENTAR BOFS.

Referencia	ID	Cardinalidad	Conjunto de características
[25]	S1	11	<Bound; Type; Location; Scope; Container; Index or Limit; Access; Buffer alias; Control flow; Surrounding loops; Input taint>
[23]	S2	22	<Write/Read; Upper/Lower bound; Data type; Memory location; Scope; Container; Pointer; Index complexity; Address complexity; Length/Limit complexity; Alias of buffer address; Alias of buffer index; Local control flow; Secondary control flow; Loop structure; Loop complexity; Asynchrony; Taint; Runtime environment dependence; Magnitude; Continuous/Discrete; Signed/Unsigned mismatch>
[26]	S3	27	<Sink classification; *Input classification*—Command line; Environment variable; File; Network—; *Input validation*—String length of source buffer; Size of source; NULL check; EOF check; Character check; Character occurrence in string check; String comparison; Other check—; *Buffer size predicate classification*—Size of destination buffer check; Size of destination (buffer-1) check; Size of destination (buffer-x) check; String length of destination buffer check; Post size check increment—; *Sink characteristics classification*—Data buffer declaration; Number of elements copied within bounds; Array write index within bounds; Format string precision within bounds; String copy within bounds; Data dependent on destination buffer size; Data dependent on destination buffer size variant, Is character case conversion sink; Resets in control predicates>
[27]	S4	23	<Sink classification; *Input classification*—Command line; Environment variable; File; Network—; *Input Validation*—String length of source buffer; Size of source; NULL check; EOF check; Character check; Character occurrence in string check; String comparison; Other check—; *Buffer size check predicate classification*—Size of destination buffer check; Size of destination (buffer-1) check; Size of destination (buffer-x) check; String length of destination buffer—; *Data Buffer Declaration Statement Classification*— Static allocation; Dynamic allocation—; *Sink characteristics classification*—Number of elements copied within bounds; Array write index within bounds; Data dependent on destination buffer size; Data dependent on destination buffer size variant>
[13]	S5	29	<*Input and Source characterization*—Input count; Inputs with limiting; Environment dependent input; Is source a buffer; Is source NULL terminated—; *Sink characteristics*—Operation Type; Has defensive limiting—; *Destination buffer characterization*—Is destination NULL terminated; Multiple writes to destination; Location; Same source and destination size; Declaration—; *Control Dependency Characteristics*—String length of source; Size of source; String length of destination; Size of destination; Source dependent loop termination; Validation; Input dependent predicates; Source buffer predicates; Destination buffer predicates—; *Data dependency precision characterization*—Inter procedural destination buffer access; Number of destination writes; Is destination buffer ambiguous; Is source ambiguous; Sink dependent ambiguous containers; Sink dependent non-ambiguous containers; Predicates referring to ambiguous containers; Predicates referring to non-ambiguous containers>
[28]	S6	26	<Sink classification; *Input classification*—Command line; Environment variable; File; Network; *Input validation*—String length of source buffer check; NULL check; EOF check; Character check; Character occurrence in string check; String comparison; Other check—; *Buffer size check predicate classification*—Size of destination buffer check; Size of destination (buffer-1) check; Size of destination (buffer-x) check; String length of destination buffer check—; *Data buffer declaration statement classification*—Static allocation; Dynamic source independent allocation; Dynamic source dependent allocation—; *Sink characteristics classification*—Number of elements copied within bounds; Array write index within bounds; String copy within bounds; Data dependent on destination buffer size; Data dependent on destination buffer size variant; Is character case conversion sink; Resets in control predicates; Post size check increment>
[22]	S7	6	<Access; Boundary; Location; Magnitude; Data size; Reach>
[12]	S8	31	<Sink type; *Memory location*—Stack; Heap; Data segment; BSS segment; Shared memory—; Container; *Index type*—Constant; Addition; Multiplication; Nonlinear; Function call; Array access—; *Address type*—Constant; Addition; Multiplication; Nonlinear; Function call; Array access—; *Length type*—Constant; Addition; Multiplication; Nonlinear; Function call; Array access—; *Sanitization*—Direct sanitization; Indirect sanitization—; Loop depth; Condition depth; Call depth; Taint>

código. En este ámbito, los resultados son prometedores ya que todos los modelos propuestos suelen ser ligeramente más efectivos que las herramientas convencionales. No obstante, no hay que descartar un posible sesgo en esos resultados, pues las evaluaciones no utilizan los mismos conjuntos de datos de partida ni tampoco comparan las mismas herramientas de análisis de código.

V-C. Limitaciones de la revisión

Dado que la revisión se planificó con el objetivo de tener una idea general y rápida de las evidencias actuales, existen ciertas limitaciones que afectan principalmente a su completitud. De este modo, la restricción del idioma (solo artículos en inglés) así como la exclusión de libros y literatura gris, implican la posibilidad de no haber incluido trabajos relevantes. Por otra parte, debido al corto plazo de ejecución y documentación del artículo no se ha podido hacer una validación más exhaustiva de todo el proceso de revisión. Idealmente, los procesos de selección, evaluación de la calidad y extracción de datos se podrían validar realizando un test-retest transcurrido algún tiempo tras la revisión [14]. Otro aspecto a mejorar es la evaluación de la calidad. Sería recomendable añadir más preguntas relativas al potencial sesgo en el proceso de evaluación de los conjuntos de caracterísicas, así como añadir preguntas del estilo a: "¿Se indica cómo extraer técnicamente cada característica?" o "¿Se realiza una evaluación del conjunto de características propuesto?".

En relación con los resultados analizados, sería conveniente añadir más tablas en las que categorizar las caracteríticas extraídas, así como aportar algún gráfico de la frecuencia de utilización.

VI. CONCLUSIONES

Detectar vulnerabilidades en el código fuente es una tarea compleja en la actualidad y los BOFs no son una excepción. Las herramientas de análisis de código han sido tradicionalmente el estándar seguido para intentar alcanzar ese objetivo. Asimismo, hoy día existen técnicas como el aprendizaje automático que pueden complementar este proceso y llegar incluso a predecir BOFs. En esencia, ambas técnicas utilizan

características para representar tales vulnerabilidades. Sin embargo, la temprana edad de esta disciplina implica la ausencia de una revisión de la literatura que indique cuántos tipos de características existen para describir y representar un BOF.

Ante esta situación, el presente trabajo ha intentado cubrir ese hueco. De la literatura revisada se han identificado 175 características que describen a un BOF, de las cuáles 79 son únicas. Tales características son, o bien de tipo cualitativo nominal—valores definidos en un rango específico—, o bien cuantitativo discreto—valores entre 0 y $+\infty$. Además, las características se podrían clasificar según su naturaleza en cuatro grupos: 1) aquellas que representan aspectos de la instrucción del BOF; 2) aquellas relativas a la entrada de datos que acaban afectando a la instrucción del BOF; 3) aquellas referentes a la presencia de instrucciones que eviten la aparición del BOF; y 4) aquellas que hacen referencia a la propia existencia del BOF. Por otra parte, los autores no utilizan una forma de representar un BOF más allá de listar un conjunto determinado de características. Por consiguiente, las 175 características quedan repartidas entre 8 propuestas diferentes—una por cada artículo revisado. De las 8 propuestas, 2 sirven como base para proponer algunos de los otros 6 modelos restantes. De igual modo, 5 conjuntos de características han sido propuestos con el objetivo de ser utilizados en algoritmos de aprendizaje automático. Según los propios autores, estos conjuntos muestran resultados prometedores; aunque cabe notar que deberían validarse primero de forma homogénea.

En definitiva, los resultados presentados en este trabajo pueden ser de especial utilidad para futuros trabajos en el ámbito de la predicción de BOFs con aprendizaje automático. Así, por ejemplo, hay dos aplicaciones directas. Por un lado la lista de características puede servir como base para crear nuevas representaciones de BOFs, y por otro lado los 8 conjuntos identificados pueden servir como punto de partida para comparar su efectividad a la hora de representar o predecir un BOF. En el primer caso bastaría con combinar un número determinado de características de alguna forma razonada. En el segundo caso se podría utilizar un mismo conjunto de datos para evaluar el rendimiento que se obtiene tras aplicar los diferentes algoritmos de aprendizaje automático.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Adenda 4 al Convenio Marco Universidad de León-Instituto Nacional de Ciberseguridad (INCIBE), por la Consejería de Educación de la Junta de Castilla y León a través del Proyecto LE028P17 y por el Ministerio de Ciencia, Innovación y Universidades a través del proyecto RTI2018-100683-B-I00.

Referencias

[1] Y. Younan, "25 Years of Vulnerabilities: 1988-2012," Sourcefire Vulnerability Research Team, Tech. Rep., 2013.

[2] B. Martin, M. Brown, A. Paller, D. Kirby, and S. Christey, "2011 CWE/SANS Top 25 Most Dangerous Software Errors," CWE/SANS, Tech. Rep., 2011.

[3] P. E. Black and I. Bojanova, "Defeating Buffer Overflow: A Trivial but Dangerous Bug," *IT Professional*, vol. 18, no. 6, pp. 58–61, 2016.

[4] J. Deckard, *Buffer Overflow Attacks: Detect, Exploit, Prevent*, 1st ed., Elsevier, Ed. Rockland, MA: Syngress, 2005.

[5] K. S. Lhee and S. J. Chapin, "Buffer overflow and format string overflow vulnerabilities," *Software - Practice and Experience*, vol. 33, no. 5, pp. 423–460, 2003.

[6] C. Cowan, P. Wagle, C. Pu, S. Beattie, and J. Walpole, "Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00.* Hilton Head, SC, USA: IEEE, 2000.

[7] B. M. Padmanabhuni and H. B. K. Tan, "Defending against Buffer-Overflow Vulnerabilities," *Computer*, vol. 44, pp. 53–60, 2011.

[8] A. Bosu, J. C. Carver, M. Hafiz, P. Hilley, and D. Janni, "Identifying the Characteristics of Vulnerable Code Changes: An Empirical Study," in *FSE 2014. Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. Hong Kong, China: ACM New York, 2014, pp. 257–268.

[9] M. Hafiz and M. Fang, "Game of detections: how are security vulnerabilities discovered in the wild?" *Empirical Software Engineering*, vol. 21, no. 5, pp. 1920–1959, 2016.

[10] J. Gong, X. H. Kuang, and Q. Liu, "Survey on Software Vulnerability Analysis Method based on Machine Learning," in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*. Changsha, China: IEEE, 2016, pp. 642–647.

[11] S. M. Ghaffarian and H. R. Shahriari, "Software Vulnerability Analysis and Discovery Using Machine-Learning and Data-Mining Techniques," *ACM Computing Surveys*, vol. 50, no. 4, 2017.

[12] Q. Meng, C. Feng, B. Zhang, and C. Tang, "Assisting in Auditing of Buffer Overflow Vulnerabilities via Machine Learning," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–13, 2017.

[13] B. M. Padmanabhuni and H. B. K. Tan, "Buffer Overflow Vulnerability Prediction from x86 Executables Using Static Analysis and Machine Learning," in *Proceedings - International Computer Software and Applications Conference*, 2015.

[14] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*. CRC press, 2016, vol. 4.

[15] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, P. Group *et al.*, "Preferred reporting items for systematic reviews and meta-analyses: the prisma statement," *PLoS medicine*, vol. 6, no. 7, p. e1000097, 2009.

[16] P. Li and B. Cui, "A Comparative Study on Software Vulnerability Static Analysis Techniques and Tools," in *Proceedings 2010 IEEE International Conference on Information Theory and Information Security, ICITIS 2010*. Beijing, China: IEEE, 2010, pp. 521–524.

[17] B. Liu, L. Shi, Z. Cai, and M. Li, "Software Vulnerability Discovery Techniques: A Survey," in *Proceedings 2012 4th International Conference on Multimedia and Security, MINES 2012*. Nanjing, China: IEEE, 2012, pp. 152–156.

[18] B. Chess and G. McGraw, "Static Analysis for Security," *IEEE Security & Privacy*, vol. 2, no. 6, pp. 76–79, 2004.

[19] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[20] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 78–87, 2013.

[21] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Information and Software Technology*, vol. 53, no. 6, pp. 625–637, 2011.

[22] I. Bojanova, P. E. Black, Y. Yesha, and Y. Wu, "The Bugs Framework (BF): A Structured Approach to Express Bugs," in *Proceedings - 2016 IEEE International Conference on Software Quality, Reliability and Security, QRS 2016*. Vienna, Austria: IEEE, 2016, pp. 175–182.

[23] K. Kratkiewicz and R. Lippmann, "Using a Diagnostic Corpus of C Programs to Evaluate Buffer Overflow Detection by Static Analysis Tools," in *Workshop on the Evaluation of Software Defect Detection Tools*, Chicago, IL, USA, 2005.

[24] K. J. Kratkiewicz, "Evaluating Static Analysis Tools for Detecting Buffer Overflows in C Code," Master's thesis, Harvard University, Boston, CA, USA, 2005.

[25] M. Zitser, R. Lippmann, and T. Leek, "Testing Static Analysis Tools using Exploitable Buffer Overflows from Open Source Code," in *PSIGSOFT '04/FSE-12 Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering*. Newport Beach, CA, USA: ACM, 2004, pp. 97–106.

[26] B. M. Padmanabhuni and H. B. K. Tan, "Predicting Buffer Overflow Vulnerabilities through Mining Light-Weight Static Code Attributes," in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*. IEEE, nov 2014, pp. 317–322.

[27] ——, "Auditing Buffer Overflow Vulnerabilities using Hybrid Static-Dynamic Analysis," in *2014 IEEE 38th Annual Computer Software and Applications Conference*. Vasteras, Sweden: IEEE, 2014, pp. 394–399.

[28] ——, "Auditing buffer overflow vulnerabilities using hybrid static-dynamic analysis," *IET Software*, vol. 10, no. 2, pp. 54–61, 2016.

Boosting child abuse victim identification in Forensic Tools with hashing techniques

Rubel Biswas
Departamento IESA
Universidad de León
Researcher at INCIBE
rbis@unileon.es

Victor González-Castro
Departamento IESA
Universidad de León
Researcher at INCIBE
vgonc@unileon.es

Eduardo Fidalgo Fernández
Departamento IESA
Universidad de León
Researcher at INCIBE
efidf@unileon.es

Deisy Chaves
Departamento IESA
Universidad de León
Researcher at INCIBE
dchas@unileon.es

***Abstract*—In this work, we present a scheme to identify occluded faces using perceptual image hashing. Most of the existing methods for this problem focus on occlusion detection and further removal of the occluded area by training a facial model. In this paper, we propose a new hashing method which does not require prior training. Our model combines frequency coefficients and statistical image information to increase the recognition accuracy of occluded faces. The proposed method aims to improve face recognition accuracy in forensic tools such as victim identification in Child Sexual Abuse (CSA) materials. Experimental results showed that the proposed method outperforms the results obtained with perceptual image hashing for occluded face identification using the LFW database.**

***Index Terms*—Face identification, Face recognition, Perceptual hashing, CLOSIB, pHash, NMF**

Type of contribution: Ongoing research

I. Introduction

Automatic face identification or recognition is widely used in many real-time applications such as forensics, surveillance or criminal identification among others. In recent years, deep learning techniques have achieved a considerable development in this area [1]. Nevertheless, there are still some open issues during face identification. One of the most challenging problems is the occlusion of the face, which can be caused by several reasons, such as self-occlusion (e.g. non-frontal position), accessories (e.g. glasses, masks or hair) or adversarial attacks (i.e. image faces modified by adding small changes to make difficult the identification).

In the literature, we can find works that deal with the automatic Child Sexual Abuse (CSA) material detection [2]. However, after detecting such kind of material, Law Enforcement Agencies (LEA) may face the problem of identifying the victims. Children usually are presented dressed with customs, accessories that cover their faces and also sometimes the CSA material receives eye adversarial attacks before uploading the material to the Web. Therefore, occluded face identification in CSA images remains a challenging task for LEAs.

To address the problem of identifying occluded faces we proposed a face recognition method based on perceptual image hashing [3] to avoid the training of facial models and represent an image content as a fixed-length vector. This research is part of the European project Forensic Against Sexual Exploitation of Children (4NSEEK), and our primary goal is to enhance the Forensic Tools ability to recognize occluded faces in child pornography materials.

II. Related Work

Due to the availability of high-end GPU cards and training data, nowadays, deep learning methods achieve state of the art results in the task of face recognition [4]. Hongjun and Aleix [5] used SVM to find a hyperplane which is parallel to the affine subspace of occluded data. While Min et al. [6] trained a SVM classifier to detect the occluded part and use the non-occluded area to match with a corresponding part of a faces gallery.

On the other hand, many perceptual image hashing methods have been proposed and applied to the field of multimedia security [7]. Researchers have focused on image hashing schemes based on the concepts of deep hashing, Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), and non-negative matrix factorization (NMF), among others.

We have noticed that occluded face recognition requires a trained model with occluded face features. To overcome the need for training a model, we designed an image hashing method to identify a person when the face image is partially occluded which can be applied in CSA cases.

III. Methodology

Fig. 1 represents the proposed perceptual hashing scheme, which is two-fold: (1) computation of the hash code of a non-occluded face image and (2) verification of the cropped occluded face.

In the first stage, Multi-Task Cascade CNN (MTCNN) method [8] is applied to detect a face contained in an image. Then, the face is cropped using the detected bounding box coordinates and re-sized it to 120 $\times$120 pixels. Next, the perceptual hash [3] and CLOSIB descriptors [9], [10] are computed to extract 64 coefficients and 128 statistical features of the face image, respectively. Afterwards, NMF [11] is applied to reduce the 128 CLOSIB features into 64 values. Finally, an element-wise multiplication between the 64 pHash coefficients and the 64 CLOSIB features is carried out to attain the final feature vector, i.e. the hash code, for the face, which is stored. In the second stage, an occluded face is cropped manually from an image and resized into 120 $\times$120 pixels. After that, the image hash codes are obtained using the process previously described. Finally, the similarity score is computed between the occluded face hash code and the ones stored with a correlation coefficient function. If the score is greater than a threshold, T, it is considered that a similar face is found in the internal storage of the module.

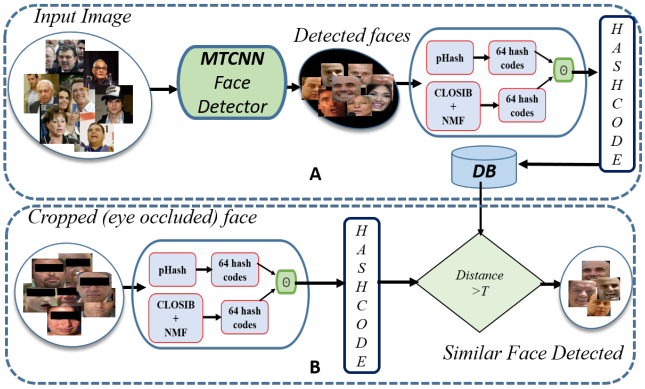


Figure 1. Occluded face verification scheme. A) Non-occluded face hash code computation and storage. B) Occluded face identification by comparing face hash code against previously stored hash codes

IV. Experimental Results and Discussion

During the experimentation, a PC with 128GB RAM and 12GB Titan Xp GPU is used. For testing purposes, we created an occluded face dataset by modifying the images in the LFW dataset. The resulting dataset, *occluded-LFW*, comprises 13,233 images with mostly frontal views of occluded faces that correspond to 5,749 identities.

The occluded-LFW dataset is used to evaluate the pHash algorithm and pHash combined with CLOSIB through NMF. We computed the correlation coefficient scores between occluded face hash codes and stored non-occluded face ones to assess their similarity. Finally, the proposed scheme retrieved as the identity the one with the maximum similarity score greater than a threshold, T. Table I presents the identification accuracy and the average processing time for occluded face recognition. In the experimentation, firstly, the hash codes of 13,233 non-occluded faces are extracted from LFW dataset and saved into the system storage. Secondly, in the occluded face identification stage, the hash code of each occluded face from the *occluded-LFW* dataset is computed and compared with the stored hash codes to identify the person. We have selected the threshold $T = 0.98$, because it reduces the false positive rate.

We have observed that the proposed combination of pHash and CLOSIB with NMF yields an accuracy of 69.89% in this experiment and outperformed pHash which obtained an accuracy of 40.3%. Presumably, because NMF allows reducing the feature vector dimension and helps to combine pHash and CLOSIB successfully to get a richer face representation. Finally, we found that the processing time of the combination of pHash and CLOSIB with NMF, in average 0.017 seconds, is higher than the one observed for pHash, in average 0.002 seconds. However the proposed method is still suitable for real-time applications such as forensics tools and it does not need to train any model.

V. Conclusions and Future Work

In this work, we proposed a perceptual image hashing scheme for occluded face recognition aiming to improve the occluded faces recognition in child pornography materials. The proposed approach is based on the combination of statistical features and frequency coefficients of an image instead than a hard training. However, to achieve accurate results at least a non-occluded face (in a similar scenario) must be stored in the system before performing the occluded face identification. Moreover, we demonstrated that the proposed hashing method obtained the highest accuracy in this task, 69.89%, outperforming pHash, which obtained an accuracy of 40.30%. However, the processing time for the proposed schema is higher than that of pHash, but it presents a better trade-off between accuracy and the processing time.

In future work, we will attempt the design of new hashing techniques and evaluate them on different state-of-the-art datasets with similar characteristics to the problem domain.

Table I
Partial occluded face identification accuracy and average processing time for occluded-LFW dataset

pHash		**pHash and CLOSIB-NMF**	
Accuracy (%)	Avg. time (Sec.)	Accuracy (%)	Avg. time (Sec.)
40.30	0.002	69.89	0.017

Acknowledgement

This research has been funded with support from the European Commission under the 4NSEEK project with Grant Agreement 821966. This publication reflects the views only of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein. This work is also supported by the framework agreement between the University of León and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01. We acknowledge NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao and A. Zisserman: "Template adaptation for face verification and identification", in *Image and Vision Computing*, vol. 79, pp. 35–48, 2018.

[2] A. Gangwar, E. Fidalgo, E. Alegre, V. Gonzáles-Castro, Pornography and Child Sexual Abuse Detection in Image and Video: A Comparative Evaluation, in: 8th International Conference on Imaging for Crime Detection and Prevention (ICDP), 2017, pp. 37–42

[3] C. Zauner: "Implementation and Benchmarking of Perceptual Image Hash Functions", in *Master's thesis, University of Applied Sciences Hagenberg, Austria*, 2010.

[4] F. Schroff, D. Kalenichenko and J. Philbin: "Facenet: A unified embedding for face recognition and clustering", in *IEEE conference on computer vision and pattern recognition*,pp. 815–823, 2015.

[5] J. Hongjun and M. M. Aleix: "Support vector machines in face recognition with occlusion", in *Proceedings of the IEEE Conference Computer Vision and Pattern Recognition*, pp. 136–141, 2009.

[6] M. Rui, H. Abdenour and D. Jean-Luc: "Improving the recognition of faces occluded by facial accessories", in *Automatic Face and Gesture Recognition and Workshops, IEEE*, pp. 442–447, 2011.

[7] C. Qin, X. Chen, J. Dong and X. Zhang: "Perceptual image hashing with selective sampling for salient structure features", in *Displays*, Vol. 45, pp. 26-37, 2016.

[8] K. Zhang, Z. Zhang, Z. Li and Y. Qiao: "Joint face detection and alignment using multitask cascaded convolutional networks", in *IEEE Signal Processing Letters*, Vol. 23, n. 10, pp. 1499–1503, 2016.

[9] O. García-olalla; L. Fernández-Robles; E. Alegre; M. Castejón-Limas; E. Fidalgo. Boosting Texture-Based Classification by Describing Statistical Information of Gray-Levels Differences. Sensors 2019, 19, 1048.

[10] O. García-olalla; Alegre, E.; Fernández-Robles, L.; Fidalgo, E.; Saikia, S. Textile retrieval based on image content from CDC and webcam cameras in indoor environments. Sensors 2018, 18, 1329.

[11] J. Pan and J. Zhang: "Large Margin Based Nonnegative Matrix Factorization and Partial Least Squares Regression for Face Recognition", in *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1822-1835, 2011.

Vulnerabilidades en altavoces inteligentes

Marván Medina Raúl*, Alejandra Guadalupe Silva Trujillo*, Bacasehua Morales Luis Carlos*
Nava Torres Claudio Isauro*, Ana Lucila Sandoval Orozco†

* Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí
Av. Manuel Nava No. 8, Zona Universitaria, C.P. 78290, San Luis Potosí, S.L.P., México
Email: raulmarvan@gmail.com, asilva@uaslp.mx, luca.bacasehua@gmail.com, ictnava@hotmail.com

†Department of Electrical Engineering, Faculty of Technology, University of Brasilia (UnB)
Campus Universitario Darcy Ribeiro,Brasilia CEP 70910-900, Brazil
Email: asandoval@redes.unb.br

***Resumen*—El uso de dispositivos en el hogar conectados a Internet alrededor del mundo ha aumentado considerablemente en los últimos años. Los "hogares inteligentes" tienen cada vez más dispositivos conectados. El IoT (Internet of Things) ha cambiado la forma de interactuar con nuestros dispositivos y por eso mismo, ha aumentado la preocupación sobre el trato que se le da a la información recopilada por parte de las empresas fabricantes. El objetivo de este trabajo es compendiar la evidencia de vulnerabilidades en la seguridad y privacidad de altavoces inteligentes, que pueden poner en riesgo la confidencialidad de los usuarios. Y derivado de ello, identificar los retos donde se requiere de mayor investigación.**

***Index Terms*—Security, IoT, Privacy, Vulnerabilities**

Tipo de contribución: *Investigación en desarrollo*

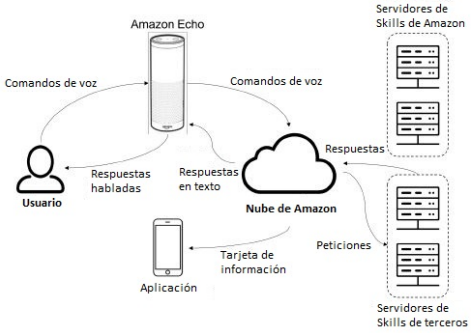


Figura 1. Arquitectura del sistema de comandos de voz de Alexa[1]

I. Introducción

Un altavoz inteligente funge como un asistente personal digital. Estos dispositivos funcionan con base en el reconocimiento de comandos por voz, con los cuales el usuario puede: i) Solicitar información sobre el clima, noticias, recetas de cocina, fecha y hora; ii) reproducir música, audiolibros y podcast; iii) realizar compras por Internet y uso de aplicaciones de terceros; iv) controlar otros dispositivos del hogar que estén conectados a Internet.

Debido a la cantidad de información de carácter personal que alojan los servidores de estas empresas, el impacto de una brecha en la seguridad sería catastrófico. El IoT es el blanco de miles de ataques masivos que buscan controlar un gran número de dispositivos, sin autorización del propietario, para acceder a sus activos, como datos confidenciales, recursos informáticos y de red.

Como se muestra en la Fig. 1, en el modelo actual de operación de algunos dispositivos del IoT se deben abrir puertos, dejando la red doméstica susceptible a ataques. Además, debido a la mala práctica actual de la administración de dispositivos, una gran parte de estos, carecen de firmware actualizado o dependen de contraseñas predeterminadas o débiles para la autenticación.

Si la idea del IoT es colocar un chip dentro de cualquier cosa y hacerlo inteligente, estaríamos hablando de carreteras, casas, edificios y ciudades inteligentes. En otras palabras, ya sea que lo sepamos conscientemente o no, estamos construyendo un planeta inteligente donde todos y todo está conectado y comunicándose sin parar. Esto se convierte, sin duda, en un gran desafío para construir las bases que garanticen los servicios sin dejar de lado los aspectos básicos de seguridad y privacidad de la información.

El objetivo de este trabajo es compendiar la evidencia de vulnerabilidades en la seguridad y privacidad de altavoces inteligentes, que pueden poner en riesgo la confidencialidad de los usuarios; con el fin de identificar algunos de los retos donde se requiere mayor investigación. En la sección II hablamos de algunas vulnerabilidades a las que se exponen los usuarios de un altavoz inteligente. Luego, en la sección III abordamos algunos de los posibles ataques. Finalmente las conclusiones se exponen en la sección IV.

II. Vulnerabilidades

Un sistema es seguro cuando la información de carácter personal del usuario es tratada de manera confidencial y segura. Es preocupante la cantidad de información que estos dispositivos pueden recopilar y la cantidad de personas que tienen acceso a ella.

Los dispositivos de IoT, como el altavoz inteligente Amazon Echo desarrollado por Amazon, son sin duda una gran fuente de evidencia digital debido a su empleo ubicuo y su modo de funcionamiento siempre activo. El Amazon Echo, desempeña un papel central para Alexa, el asistente virtual inteligente basado en la nube [2]. Alexa, es el servicio de voz ubicado en la nube disponible en los dispositivos de Amazon, el cual cuenta con diversas funcionalidades o Skills [3]. En abril de 2019, se reportó que Amazon ha empleado a miles de trabajadores de tiempo completo y contratistas en varios países. Esto con el fin de escuchar hasta 1.000 archivos de audio en turnos que duran hasta nueve horas. Ante tal

filtración la empresa argumentó que sus empleados no tienen acceso directo a la información que puede identificar a la persona o la cuenta vinculada al dispositivo [4].

Un estudio de la Universidad de Virginia [5] encontró las siguientes vulnerabilidades específicas dentro del dispositivo Amazon Echo, que de ser aprovechadas sin el consentimiento del usuario, pueden poner en riesgo su privacidad y confidencialidad:

- Todas las solicitudes a Alexa se almacenan como archivos de audio en la cuenta de Amazon del usuario y se cifran antes de ser enviados a la nube, lo que dificulta la captura de los mismos, incluso si la red doméstica tiene pocas protecciones.
- Cualquier persona que se encuentre cerca del micrófono puede hacer solicitudes a Alexa. Lo anterior se ha visto demostrado cuando, por error, se realizaron compras a través de Alexa debido a una noticia televisiva que activó al dispositivo [5].
- Investigadores de la Universidad de Zhejiang explotaron otra vulnerabilidad conocida como DolphinAttack [5]. Este ataque convierte los comandos de voz en frecuencias demasiado altas para que el oído humano las escuche, pero que son detectadas fácilmente por Amazon Echo.
- Un problema más de privacidad es que Amazon puede ver las conversaciones de los usuarios con el dispositivo y usarlos en beneficio de la red neuronal de aprendizaje de Alexa. El mal uso de esta información por otros, presenta la mayor amenaza potencial para la privacidad del usuario.

III. Ataques

Existe casos registrados donde se logró identificar y atacar diferentes vulnerabilidades en el altavoz inteligente Amazon Echo, tales como ataques de comando y ataques a través de la red [1].

III-A. Ataques de Comando

La forma de atacar a Alexa fue creando una onda de sonido que Echo reconoce como la palabra de activación pero que suena como ruido para un humano [5]. El resultado de este ataque no arrojó resultados contundentes ya que los sonidos generados no eran del todo irreconocibles para el ser humano.

Las Skills son tareas que Amazon Echo puede realizar para un usuario, como verificar el clima o pedir una pizza. Amazon ofrece algunas skills que vienen como una funcionalidad integrada para los usuarios de Amazon Echo, las cuales son gestionadas por sus propios servidores [1].

En otro estudio se sugirió un ataque conocido como Skill Squatting [6]. La idea central del ataque es simple, dado que la pronunciación de algunas palabras o frases pueden llegar a sonar muy parecido, un atacante puede aprovechar el hecho de que un usuario esté intentando solicitar una skill mediante una palabra específica y Alexa la malinterprete por otra. Esto ocasionaria un direccionamiento hacia una skill maliciosa, debido a un error en la interpretación de la entrada. Este ataque es muy similar a la tipificación de nombres de dominio, donde un atacante predice un error común para llevar a un usuario a un sitio malicioso. La investigación probó 27 ataques de tipo Skill Squatting y fueron capaces de redireccionar con éxito 25 de las Skills al menos una vez, lo que representa un 92.6 % del total y demuestra la viabilidad del ataque [6].

III-B. Ataques a través de la red

En [5] se describen varios ataques a la red del dispositivo Amazon Echo. A través de un análisis de tráfico, se llevó a cabo la captura de una gran cantidad de paquetes enviados entre el dispositivo y algunas otras direcciones. Las acciones que se lograron realizar con estos paquetes fueron limitadas, debido a que se encontraban cifrados y solamente se obtuvo la cabecera del paquete. Con tal información se puede inferir el promedio de tiempo de uso del dispositivo y las horas en las que el usuario se encuentra en su hogar [5].

IV. Conclusiones

Una estudio realizado sobre usuarios de altavoces inteligentes concluyó que muchas personas justifican su falta de preocupación por la privacidad basándose en una comprensión incompleta de los riesgos de privacidad y en su mayoría muestran resignación hacia la pérdida de privacidad [7].

La confidencialidad e integridad de la información de carácter personal con la que tratan las empresas debe ser prioridad a la hora de desarrollar un servicio o producto. No se puede poner en riesgo a los usuarios con el fin de disminuir costos para ofrecer en el mercado dispositivos económicos. Como uno de los mayores retos, es importante que los usuarios puedan identificar los beneficios y vulnerabilidades en el uso de dispositivos IoT. Otras de las áreas donde se requiere mayor empeño, es en la gestión de identidad, al implementar protocolos para autenticación. Por otro lado, las compañías fabricantes deberían mejorar sus interfaces guiando al usuario a posibles intrusiones de privacidad por una mala configuración. Así como también ofrecer actualizaciones periódicas o parches de seguridad.

Referencias

[1] W. Haack, M. Severance, M. Wallace, and J. Wohlwend, "Security analysis of the amazon echo," in *Security Analysis of the Amazon Echo*, 2017.

[2] H. Chung, J. Park, and S. Lee, "Digital forensic approaches for amazon alexa ecosystem," *Digital Investigation*, vol. 22, pp. S15 – S25, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287617301974

[3] Amazon, "Alexa skills kit - desarrolla para la voz con amazon." [Online]. Available: https://developer.amazon.com/es/alexa-skills-kit

[4] J. Valinsky, "Amazon emplea a miles de personas para escuchar lo que le dices a alexa, según reportes," Apr 2019. [Online]. Available: https://cnnespanol.cnn.com/2019/04/11/amazon-alexa-conversaciones-escucha-miles-personas-emplea/

[5] C. Jackson and A. Orebaugh, "A study of security and privacy issues associated with the amazon echo," *International Journal of Internet of Things and Cyber-Assurance*, vol. 1, no. 1, pp. 91–100, 2018. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJITCA.2018.090172

[6] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, "Skill squatting attacks on amazon alexa," in *Proceedings of the 27th USENIX Conference on Security Symposium*, ser. SEC'18. Berkeley, CA, USA: USENIX Association, 2018, pp. 33–47. [Online]. Available: http://dl.acm.org/citation.cfm?id=3277203.3277207

[7] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 102:1–102:31, Nov. 2018. [Online]. Available: http://doi.acm.org/10.1145/3274371

Visión General de las Técnicas de Identificación de la Fuente de Vídeos Digitales

Raquel Ramos López, Elena Almaraz Luengo, Ana Lucila Sandoval Orozco, Luis Javier García Villalba*

Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
Emails: raqram01@ucm.es, ealmarazluengo@mat.ucm.es, {asandoval, javiergv}@fdi.ucm.es

***Resumen*—Los dispositivos móviles han tenido un impacto enorme en nuestra sociedad, no sólo sirven para que las personas se comuniquen sino que gracias a la diversidad de aplicaciones que contienen, hacen que sea una práctica habitual la creación de vídeos digitales con este tipo de dispositivos dada su facilidad de grabación y distribución. Sin embargo, estos vídeos pueden mostrar en ocasiones actos ilegales por lo que el análisis forense de dispositivos móviles adquiere una trascendental relevancia para deslindar responsabilidades o como parte de la evidencia en procesos judiciales siendo el propósito de esta rama del análisis forense relacionar a un individuo con un dispositivo. En este documento se presenta un resumen de las principales contribuciones en el mundo de la identificación de la fuente de vídeos digitales de dispositivos móviles que utilizan el patrón de ruido PRNU, y como afecta la compresión de un vídeo al patrón PRNU.**

***Index Terms*—Video Forensics, PRNU, SPN, H.264/AVC.**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

La evolución de las cámaras digitales de los teléfonos inteligentes ha sido de tal magnitud que han conseguido poner fin al rápido crecimiento de las cámaras digitales. Precisamente, la posibilidad de usar el teléfono móvil como videocámara es una de las funcionalidades a la que los usuarios dan mayor importancia. Como consecuencia de este fenómeno y de la gran cantidad de tiempo que una persona pasa junto a un teléfono inteligente, éste se ha convertido en el primer dispositivo de grabación de vídeos para muchos usuarios en la sociedad actual. A su vez, estos vídeos digitales pueden ser realizados en cualquier momento y lugar con diferentes fines, distribuyéndose en Internet en un corto periodo de tiempo y mostrando en ocasiones actos ilegales como pueden ser los relacionados con el terrorismo, pornografía infantil, espionaje industrial, etc.

Dentro del análisis forense de vídeos digitales existen tres ramas principales: la identificación de la fuente de adquisición detección ilegal de reproducción de vídeos y compresión de vídeos [1]. Este artículo se centra en recopilar cómo se ha abordado la problemática de la identificación de la fuente de adquisición en vídeos digitales por parte de la comunidad científica. Además de esta sección de introducción, el trabajo se estructura en IV secciones más: En la Sección II se realiza una descripción de las técnicas de identificación de la fuente de vídeos digitales disponibles en la literatura. En la Sección III se detallan las contribuciones que investigan los efectos que tiene la compresión del vídeo en la técnica forense de identificación de vídeos digitales. Por último, la Sección IV resume las principales conclusiones obtenidas en este artículo.

II. Resumen de las Técnicas de Identificación de la Fuente de Adquisición de Vídeos Digitales

El análisis de la fuente de adquisición de vídeos es uno de los primeros problemas que han surgido en las técnicas de análisis forense. Dentro de la identificación de la fuente de adquisición existen dos grandes enfoques: escenarios cerrados o escenarios abiertos. Un escenario cerrado es aquel en el cual la identificación de la fuente del vídeo se realiza sobre un conjunto de cámaras concreto y conocido. Para este enfoque, normalmente se utiliza un conjunto de vídeos de cada dispositivo para entrenar un clasificador y posteriormente se predice la fuente de adquisición de los vídeos objeto de investigación. En un escenario abierto, el analista forense no conoce a priori el conjunto de dispositivos a los que pertenecen los vídeos a identificar su fuente de adquisición. El objetivo no es identificar la marca y modelo de los vídeos sino poder agruparlos en conjuntos disjuntos en los que todos sus vídeos pertenecen al mismo dispositivo. Este último planteamiento es más realista puesto que en muchos casos el analista desconoce por completo el conjunto de dispositivos a los que puede pertenecer un conjunto de vídeos.

Identificar el dispositivo que genera un contenido digital es muy importante en el contexto de un proceso judicial debido a que puede incriminar o deslindar responsabilidades a un sospechoso ante un acto delictivo. Para realizar cualquier tipo de clasificación de vídeos ya sea en escenarios abiertos o cerrados, se necesitan obtener ciertas características que permitan a las técnicas de clasificación realizar su tarea. Según [2] se establecen cuatro grupos de técnicas para este fin: basadas en la aberración de las lentes ([3],[4]), en la interpolación de la matriz ([5],[6]), en las imperfecciones del sensor ([7],[8]) y por ultimo en el uso de las características de la imagen ([9],[10]),[11]).

Este documento se centra en analizar las contribuciones existentes en la literatura en relación a la técnica de las imperfecciones del sensor.

Uno de los primeros trabajos relacionados con la técnica de las imperfecciones del sensor surgió por primera vez en 1999. En [12] se demostró que las corrientes oscuras de los chips CCD *Charge Coupled Device* de las videocámaras forman un patrón fijo de ruido, este patrón se utiliza como una

”huella”para identificar el dispositivo de origen. Este enfoque es limitado porque el ruido térmico sólo puede extraerse en los fotogramas de color negro y la propiedad de las corrientes oscuras es una señal débil que no sobrevive a la técnica de compresión del vídeo.

El tiempo ha demostrado que la técnica desarrollada en [7] que identifica sensores de imágenes basados en el ruido de respuesta no uniforme PRNU *(Photo-Response Non-Uniformity Noise)* proporciona una “huella digital” mucho más robusta y fiable.

El patrón de ruido PRNU se produce por la variación de sensibilidad de los píxeles individuales a la luz, debido a la falta de homogeneidad e impurezas en los chips de silicio, y a las imperfecciones introducidas en el proceso de fabricación del sensor. En el caso de los vídeos puede parecer que la estimación del patrón PRNU de una cámara de un vídeo a partir de una secuencia del vídeo es más sencilla que para el caso de las imágenes fijas, debido a la gran cantidad de fotogramas disponibles que hay en un vídeo. Sin embargo, esto no es cierto por dos razones principales; en primer lugar, la resolución espacial de vídeos es mucho menor que la de las imágenes fijas y, en segundo lugar, los fotogramas de vídeos generalmente contienen ratios de compresión más elevados que las imágenes comprimidas en formato JPEG.

En [13] se consiguió mejorar la propuesta anterior [7] donde se investigó el problema de atribución del dispositivo de la fuente de vídeo y se demostró que el ruido PRNU podría usarse para identificar la videocámara origen de un vídeo digital, incluso en vídeos de baja resolución, mediante la estimación de la huella de PRNU de los fotogramas de un vídeo. Un vídeo con una duración de diez minutos fue suficiente para identificar el dispositivo origen de vídeos con una resolución de (264X352 píxeles).

[14] mejoró los resultados anteriores aplicando un filtro MACE *(Minimum Average Correlation Energy)* a la huella del ruido PRNU mientras se probaba su similitud con el ruido de patrón del sensor (SPN) de un video de entrada. Mediante esta técnica, se logró una mejora de hasta el 10 % de la precisión en comparación con el método [13] para vídeos con baja resolución 128x128 píxeles).

En [15] se investigó el impacto que tenía el ruido PRNU en los vídeos procedentes de Youtube. Inicialmente se utilizaron un conjunto de cámaras web y códecs para grabar y codificar los vídeos, después se subieron y se descargaron desde Youtube, sobre estos vídeos se estimó el ruido SPN del sensor para determinar el origen, aunque la técnica dio buenos resultados, a día de hoy y debido a la evolución de los dispositivos digitales estos resultados ya están desactualizados.

[16] propuso un enfoque híbrido a la identificación de la fuente de un vídeo. Por cada dispositivo sobre el que se realizaron los experimentos, se seleccionaron imágenes y establecieron funciones de transferencia entre las huellas digitales estimadas desde las imágenes y los fotogramas de un vídeo, esta técnica se denomina coincidencia de imagen a vídeo. Utilizaron un amplio conjunto de cámaras procedentes de dispositivos móviles (teléfonos y tabletas). Estas funciones de transferencia están relacionadas con los parámetros de recorte y escala que mejor coinciden con las huellas estimada a partir de imágenes fijas y de los fotogramas de un vídeo (no estabilizado), antes de correlacionarlos con el ruido del sensor (PRNU). El método propuesto tiene buenos resultados de identificación en vídeos nativos, pero la precisión de la identificación de la fuente de vídeos procedentes de Youtube no es tan alta en relación a los dispositivos móviles. Además, en el caso de las imágenes compartidas en Facebook, la estimación de una huella digital utilizando imágenes de una fuente desconocida no es realista, ya que se supone que todas provienen del mismo dispositivo.

En [17] presenta un esquema de identificación de fuente de vídeos digitales basado en el ruido PNRU y las máquinas de soporte vectoriales, SVM *Support Vector Machine*. Dado un vídeo de entrada se extraen los fotogramas con cambios de escena más significativos mediante el histograma de color. Un total de 81 funciones, que son los componentes de Wavelet del sensor se utilizan para entrenar el clasificador SVM con vídeos de entrenamiento. Se utilizaron en total 5 dispositivos distintos de 5 marcas diferentes para entrenar el clasificador SVM. Los resultados obtenidos muestran una tasa de acierto es del 87 % o 90 %, dependiente de la resolución del vídeo.

III. Impacto de la compresión de vídeos digitales sobre el patrón PRNU

La compresión es el proceso de reducir la cantidad de información que contiene un vídeo para que se pueda almacenar y transmitir. En general, la compresión puede ser con o sin pérdidas en función de si la información que se recupera coincide exactamente con la original o es sólo una aproximación. La compresión con pérdidas es la más habitual en la codificación de vídeos. Para ello se basa en la eliminación de tres tipos de redundancias:

- Redundancia espacial: Está relacionada con los píxeles que están próximos unos de otros, puesto que tienen un parecido muy grande entre ellos. Se utilizan métodos transformados como la cuantificación o DCT *(Discrete Cosine Transform)* que eliminan esta redundancia. Estas técnicas serán conocidas como codificación *intra-trama*.
- Redundancia temporal: aprovecha la idea de que un píxel se repite a lo largo del tiempo entre todos los fotogramas que componen un vídeo. Para eliminarla, se utilizan métodos de predicción que tratan de deducir la posición futura de los píxeles, utilizando una predicción *inter-trama* junto con la técnica de compensación de movimiento que obtiene el fotograma de predicción a partir de fotogramas pasados y/o futuros. Esta eliminación de redundancia es la que más comprime el vídeo.
- Redundancia estadística: trata de determinar que valores de bit se repiten con más en una secuencia. Se utlizaran métodos como VLC *(Variable Length Code)* y el RLC *(Run Length Code)* para poder eliminarla.

Hoy en día, el estándar de compresión más utilizado por los dispositivos móviles y redes sociales es H.264/AVC *(Advanced Video Coding)* o MPEG-4 Part 10 es un método y un formato de compresión de vídeo capaz de convertir un vídeo digital en un formato que ocupa menor espacio para ser almacenado y transmitido. Su evolución es H.265 o MPEG-H Part 2 o HEVC *(High Efficiency Video Coding)*se presenta como una evolución del H.264/AVC. H.265 mejora en un 40 % la tasa de compresión del H.264/AVC pero sus

necesidades de cálculo son hasta 4 veces superiores, apenas hay chips para las tecnologías actuales, es capaz de manejar resoluciones de 320x240 a 7680x4320.

Una secuencia de vídeo se compone de grupos de imágenes(GOP) compuestos por tres tipos de imágenes diferentes denominadas fotogramas tipo I, P y B. Los fotogramas tipo I sólo explotan la redundancia espacial y se codifican por si mismos. Los fotogramas tipo P se codifican mediante predicción a partir de un fotograma de referencia pasado que puede ser de tipo I o P. Por último, los fotogramas tipo B se codifican mediante la predicción bidireccional a partir de un fotograma de referencia pasado y otro futuro que pueden ser de tipo I o P.

Debido a que la compresión del vídeo es mucho más compleja que en el caso de la compresión de imágenes fijas [18], es necesario averiguar el impacto que tiene la compresión del vídeo sobre el patrón de ruido PRNU. Así, en la literatura encontramos en [19] se asume que los fotogramas tipo I son los mejores para estimar la huella digital, otros autores como [20] dan la misma importancia a los fotogramas tipo I, P y B y son utilizados para obtener la estimación de la huella. Por otro lado, detectaron que se obtiene una baja precisión en la identificación de la fuente cuando se realiza en vídeos comprimidos por YouTube o Whatsapp (en comparación con vídeos procedentes de dispositivos móviles). Por lo tanto, es obvio que la compresión de vídeo afecta significativamente o degrada el ruido PRNU en los cuadros de vídeo. Este hecho debe tenerse en cuenta al estimar la huella del patrón de ruido PRNU a partir de vídeos altamente comprimidos. [18]. En la literatura [18] se demuestra que el ruido PRNU de un bloque sobrevive a la compresión si los coeficientes DCT-AC del residuo de predicción del bloque no son todos cero. Por lo tanto, para estimar el ruido PRNU de vídeo, solo se utilizan los bloques que tengan al menos un coeficiente DCT-AC no nulo en las tramas I, P y B. Por el contrario, si los coeficientes DCT-AC de un residuo de predicción de un bloque dado son todos nulos, entonces el ruido PRNU que contiene se pierde irreversiblemente y se reemplaza por el de sus bloques de predicción.

En [18] consiguen responder a la siguiente cuestión: ¿Es mejor utilizar sólo los fotogramas tipo I o todos los fotogramas disponibles en un vídeo? Se pueden utilizar todos los fotogramas de un vídeo siempre y cuando se cumpla la siguiente hipótesis: Si los coeficientes DCT-AC del residuo de predicción del bloque de un fotograma no son todos cero se puede utilizar el bloque para estimar la huella PRNU, de lo contrario se descarta.

IV. Conclusiones

En este artículo se ha presentado una visión general y actualizada de las diferentes técnicas propuestas en la literatura de la identificación de la fuente de vídeos digitales de dispositivos móviles, llegando a la siguiente conclsuión:

A pesar de que los vídeos contienen mucha información temporal en relación a imágenes fijas, al ser la técnica de compresión de un vídeo mucho más compleja que en el caso de imágenes es necesario tener en cuenta el efecto de la compresión a la hora de estimar con fiabilidad el patrón del ruido del sensor PRNU.

Agradecimientos

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700326.

Referencias

[1] P. Bestagini, K. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and K. Tubaro, "An Overview on Video Forensics," in *Proceedings of the 20th European Signal Conference (EUSIPCO)*, Bucharest, Romania, August 2012, pp. 1229–1233.

[2] T. Van Lanh, S. Chong, S. Emmanuel, and S. Kankanhalli, "A Survey on Digital Camera Image Forensic Methods," in *International Conference on Multimedia and Expo. IEEE*, Beijing, China, July 2007, pp. 16–19.

[3] K. Choi, "Source Camera Identification Using Footprints From Lens Aberration," in *Digital Photography II. SPIE International Society For Optical Engineering*, February 2006, pp. 60 690J–60–690J–8.

[4] K. Choi, E. Lam, and K. Wong, "Automatic Source Camera Identification Using the Intrinsic Lens Radial Distortion," *Optics Express*, vol. 14, no. 24, pp. 11 551–11 565, November 2006.

[5] S. Bayram, H. T. Sencar, and M. N., "Improvements on Source Camera-Model Identification Based on CFA Interpolation," in *Working Group 11.9 International Conference on Digital Forensics.Springer*, February 2006, pp. 24—-27.

[6] Y. Long and Y. Huang, "Image Based Source Camera Identification using Demosaicking," in *IEEE 8th Workshop on Multimedia Signal Processing. IEEE*, October 2006, pp. 419—-424.

[7] J. Lukas, J. Fridrich, and M. Goljan, "Digital Camera Identification from Sensor Pattern Noise," *IIEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[8] J. Lukas, J. Fridrich, and M. Goljan, Chen, "Determining image origin and integrity using sensor noise," *Information Forensics and Security, IEEE Transactions*, vol. 3, no. 1, pp. 74–90, September 2008.

[9] Y. Hu and L. Chang-Tsun, "Selecting forensic features for robust source camera identification," in *Computer Symposium (ICS)*, 2010, pp. 506—-511.

[10] L. Ozparlak and I. Avcibas, "Differentiating Between Images Using Wavelet-Based Transforms:A Comparative Study," *Information Forensics and Security, IEEE Transactions*, vol. 6, no. 4, pp. 1418—-1431, December 2011.

[11] Q. Liu, X. Li, L. Chen, H. Chi, P. Cooper, Z. Chen, M. Qiao, and S. A.H, "Identification of Smartphone-Image Source and Manipulation," *Applied Artificial Intelligence, ser. Lecture Notes in Computer Science*, vol. 7345, no. 4, pp. 262—-271, June 2012.

[12] K. Kurosawa, N. Saitoh, and K. Kuroki, "CCD Fingerprint Method Identification of a Video Camera from Videotaped Images," in *Proceedings 1999 International Conference on Image Processing(Cat. 99CH36348)*, Kobe, Japan, October 1999, pp. 537–540.

[13] M. G. M. Chen, J. Fridrich and J. Lukas, "Source Digital Camcorder Identification using Sensor Photo Response Non-Uniformity," *Security,Steganography,and Watermarking of Multimedia Contents IX*, vol. 6505, p. 65051G, June 2007.

[14] J. James, *Camcorder Identification for Heavily Compressed Low Resolution Videos. CSA 2011 & WCC 2011 Proceedings*. Springer, September 2011.

[15] W. van Houten and Z. Geradts, "Source video camera identification for multiply compressed videos originating from youtube," *Digital Investigation*, vol. 6, pp. 48–60, September 2009.

[16] M. Iuliani, M. Fontani, D. Shullani, and A. Piva, "PA hybrid approach to video source identification," *CoRR abs/1705.01854*, May 2017.

[17] L. J. García Villalba, A. L. Sandoval Orozco, R. Ramos López, and J. C. Hernández Castro, "Identification of Smartphone Brand and Model via Forensic Video Analysis," *Expert Systems with Applications*, vol. 55, no. 15, pp. 59–69, August 2016.

[18] E. Kiegaing Kouokam and A. Emir Dirik, "PRNU-based source device attribution for YouTube Videos," *Expert Systems with Applications*, vol. 29, pp. 91–100, June 2019.

[19] S. Taspinar, M. Mohanty, and N. Memon, "Source camera attribution using stabilized video," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, Abu Dhabi, United Arab Emirates, December 2016, pp. 1229–1233.

[20] S. Dasara, F. Marco, S. Massimo, S. Omar, A. Piva, and P. Alessandro, "Vision: a video and image dataset for source identification," *EURASIP Journal on Information Security*, October 2017.

Seguridad en Dispositivos Médicos Implantables

Carmen Camara
Universidad Carlos III de Madrid
macamara@pa.uc3m.es

***Abstract*—La bioingeniería es un campo en expansión. Las nuevas tecnologías parecen proporcionar un tratamiento más eficaz de las enfermedades o de las deficiencias humanas. Los Dispositivos Médicos Implantables (DMIs) constituyen un ejemplo. Estos dispositivos poseen actualmente más capacidad de computación, toma de decisiones y comunicación. Varios trabajos de investigación en el campo de la seguridad informática han identificado serios riesgos de seguridad y privacidad en los DMIs que podrían comprometer el implante e incluso la salud del paciente que lo porta. La tesis examina los principales objetivos de seguridad para la próxima generación de DMIs, analiza los mecanismos de protección más relevantes propuestos hasta ahora, y plantea soluciones de seguridad, principalmente basadas en medidas biométricas, para la mitigación de los problemas de seguridad encontrados. Las propuestas de seguridad deben tener en cuenta las limitaciones inherentes de estos pequeños dispositivos implantados: energía, almacenamiento y potencia de cálculo. Por otra parte, las soluciones propuestas deben lograr un equilibrio adecuado entre la seguridad del paciente y el nivel de seguridad ofrecido, siendo la vida útil de la batería otro parámetro crítico en la fase de diseño.**

***Index Terms*—Dispositivo Médico, Seguridad en e-health, Biometría.**

Type of contribution: *Premio Doctoral RENIC*

I. Introducción

Los Dispositivos Médicos Implantables (DMIs) son dispositivos electrónicos implantados dentro del cuerpo para tratar una enfermedad, controlar el estado o mejorar el funcionamiento de alguna parte del cuerpo, o simplemente para proporcionar al paciente una capacidad que no poseía antes [1]. Ejemplos actuales de DMI incluyen marcapasos y desfibriladores para monitorear y tratar afecciones cardíacas; neuroestimuladores para la estimulación cerebral profunda en casos como el Parkinson o la Epilepsia; sistemas de administración de fármacos en forma de bombas de infusión; y una variedad de biosensores para adquirir y procesar diferentes bioseñales. Un ejemplo ilustrativo de un marcapasos (o neuroestimulador) genérico y una bomba de insulina se muestra en la figura 1.

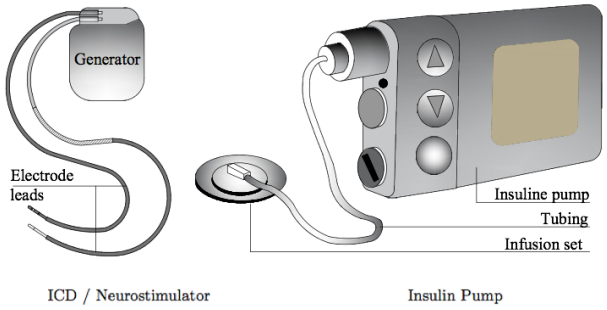


Figura 1. Ejemplo de DMIs

Los DMIs más modernos han comenzado a incorporar numerosas funciones de comunicación y redes (generalmente conocidas como telemetría) así como capacidades de computación cada vez más sofisticadas. Esto ha propiciado implantes con mayor inteligencia y pacientes con más autonomía, ya que el personal médico puede acceder a los datos y reconfigurar el implante de forma remota (es decir, sin que el paciente esté físicamente presente en las instalaciones médicas). Aparte de una importante reducción de costos, las capacidades de telemetría y cómputo también permiten a los profesionales de la atención médica monitorear constantemente la condición del paciente y desarrollar nuevas técnicas de diagnóstico basadas en una Intra Body Network (IBN) de dispositivos médicos [2], [3], [4].

Evolucionar desde un DMI electromecánico a uno con capacidades de cómputo y de comunicación más avanzadas tiene muchos beneficios pero también conlleva numerosos riesgos de seguridad y privacidad para el paciente. La mayoría de estos riesgos son relativamente bien conocidos en los escenarios clásicos de comunicaciones entre dispositivos, aunque en muchos aspectos sus repercusiones son mucho más críticas en el caso de los implantes. Los ataques contra un DMI pueden poner en riesgo la seguridad del paciente que lo porta, con consecuencias fatales en ciertos casos. Causar un mal funcionamiento intencionado en un implante puede causar la muerte y, tal como lo reconoce la Food and Drug Administration (FDA) de EE.UU, tales ataques deliberados podrían ser mucho más difíciles de detectar que los ataques accidentales [5]. Además, estos dispositivos almacenan y transmiten información médica muy delicada que requiere se protegida, según lo dictado por las directivas europeas (por ejemplo, la Directiva 95/46/ECC) y estadunidenses (por ejemplo, la Directiva CFR 164.312) [6], [7].

Si bien todavía no se conocen incidentes reales, se han demostrado con éxito varios ataques contra DMIs en el laboratorio [8], [9], [10], [11]. Estos ataques han demostrado cómo un adversario puede desactivar o reprogramar terapias en un marcapasos con conectividad inalámbrica e incluso inducir un estado de shock al paciente [12]. Otros ataques agotan la batería y dejan al dispositivo inoperativo [13], lo que a menudo implica que el paciente deba someterse a un procedimiento quirúrgico para reemplazar la batería del DMI. Además, en el caso de los implantes cardíacos, tienen un interruptor cuya posición de desconexión se consigue simplemente aplicando un campo magnético intenso [14]. La existencia de este mecanismo está motivada por la necesidad de proteger a los DMIs frete a posibles campos electromagnéticos, por ejemplo, cuando el paciente se somete a una cirugía cardíaca usando dispositivos de

electrocauterización [15]. Sin embargo, esto podría ser explotado fácilmente por un atacante, ya que la activación de dicho mecanismo primitivo no requiere ningún tipo de autenticación.

II. Contenido de la tesis

Para evitar ataques, es imperativo que la nueva generación de IMD esté equipada con mecanismos sólidos de ciberseguridad que garanticen las propiedades de seguridad básicas, como la confidencialidad, la integridad y la disponibilidad. La inclusión de estos mecanismos en la fase de diseño, así como el acceso abierto a todos sus detalles técnicos (facilitando su análisis) son dos propiedades que, en nuestra opinión, deberían garantizarse. La tesis doctoral se encuentra divida en cinco capítulos cuyo contenido, a continuación, se resume brevemente.

En el Capítulo 1, presentamos un estado de la cuestión sobre cuestiones de seguridad y privacidad en DMIs, discutimos los mecanismos más relevantes propuestos para abordar estos desafíos y analizamos su idoneidad, ventajas y principales inconvenientes.

En el Capítulo 2, mostramos cómo el uso de señales electrocardiográficas (ECGs) altamente comprimidas (sólo 24 coeficientes de la Transformada Hadamard) es suficiente para identificar inequívocamente individuos con un alto rendimiento (precisión de clasificación del 97% y errores del sistema de identificación del orden de 10^{-2}).

En el Capítulo 3 presentamos un nuevo esquema de Autenticación Continua (AC) que, contrariamente a los trabajos previos en esta área, considera las señales ECG como flujos de datos continuos. El sistema propuesto de AC basado en señales cardíacas está diseñado para aplicaciones en tiempo real y puede ofrecer una precisión de hasta el 96%, con un rendimiento del sistema casi perfecto (estadístico kappa > 80 %).

En el Capítulo 4, proponemos un protocolo de verificación de la distancia para gestionar el control de acceso al DMI: ACIMD. ACIMD combina dos características, verificación de identidad (autenticación) y verificación de la proximidad (comprobación de la distancia). El mecanismo de autenticación es compatible con el estándar ISO/IEC 9798-2 y se realiza utilizando la señal ECG con todas sus ondas, lo cual es difícilmente replicable por un atacante que se encuentre distante. Hemos evaluado el rendimiento de usando señales ECG de 199 individuos durante 24 horas, y hemos considerando tres estrategias posibles para el adversario. Los resultados muestran que se puede lograr una precisión del 87.07% en la autenticación.

Finalmente, en el Capítulo 5 extraemos algunas conclusiones y resumimos los trabajos publicados (es decir, revistas científicas con alto factor de impacto y conferencias internacionales prestigiosas).

III. Enlace a la memoria

La tesis doctoral, la cual fue defendida el 19 de Febrero de 2018, puede ser descargada desde el siguiente enlace: https://e-archivo.uc3m.es/handle/10016/27319.

References

[1] J. A. Hansen and N. M. Hansen, "A taxonomy of vulnerabilities in implantable medical devices," in *Proc. of the second annual workshop on Security and privacy in medical and home-care systems*, ser. SPIMACS '10. New York, USA: ACM, 2010, pp. 13–20.

[2] M. A. Callejon, D. Naranjo-Hernandez, J. Reina-Tosina, and L. M. Roa, "A comprehensive study into intrabody communication measurements," *IEEE Transactions on Instrumentation and Measurement,*, vol. 62, no. 9, pp. 2446–2455, 2013.

[3] M. A. Callejon, L. M. Roa, J. Reina-Tosina, and D. Naranjo-Hernandez, "Study of attenuation and dispersion through the skin in intrabody communications systems," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, pp. 159–165, 2012.

[4] M. Seyedi, B. Kibret, D. T. H. Lai, and M. Faulkner, "A survey on intrabody communications for body area network applications," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 8, pp. 2067–2079, 2013.

[5] U. Food and D. A. (FDA), "Medical device safety," http://wireless.fcc.gov/services/index.htm?job=service_bandplan&id=medical_implant, Consulted on Nov. of 2013.

[6] HIPPA, "Security standards: Technical safeguards," vol. 2, no. 4, pp. 1–17, 2007.

[7] S. Shivshankar and K. Summerhayes, *Challenges of Conducting Medical Device Studies*. Institute of Clinical Research, 2007.

[8] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel, "Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses," in *Proc. of the 29th Annual IEEE Symposium on Security and Privacy*, 2008, pp. 129–142.

[9] C. Li, A. Raghunathan, and N. Jha, "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system," in *13th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, June 2011, pp. 150–156.

[10] E. Marin, D. Singelée, F. D. Garcia, T. Chothia, R. Willems, and B. Preneel, "On the (in)security of the latest generation implantable cardiac defibrillators and how to secure them," in *Proc. of ACSAC*. ACM, 2016, pp. 226–236.

[11] D. J. Slotwiner, F. Deering, K. Fu, A. M. Russo, M. N. Walsh, and G. F. Van Hare, "Cybersecurity vulnerabilities of cardiac implantable electronic devices," *Heart Rhythm*, May 2018. [Online]. Available: https://www.heartrhythmjournal.com/article/S1547-5271(18)30467-3/fulltext

[12] K. Fu, "Inside risks: Reducing risks of implantable medical devices," *ACM Communications*, vol. 52, no. 6, pp. 25–27, 2009.

[13] X. Hei, X. Du, J. Wu, and F. Hu, "Defending resource depletion attacks on implantable medical devices," in *Proc. of IEEE Global Telecommunications Conference (GLOBECOM)*, 2010, pp. 1–5.

[14] Medtronic, "Implantable pacemaker and defibrillator information," *Patient Services*, vol. 1, no. 800, pp. 551–5544, x41 835, 2006.

[15] M. de Sousa, G. Klein, T. Korte, and M. Niehaus, "Electromagnetic interference in patients with implanted cardioverter-defibrillators and implantable loop recorders," *Indian Pacing Electrophysiol Journal*, vol. 2, no. 3, pp. 79–84, 2002.

Ciberseguridad aplicada a la automoción. Smart car cibersecurity

Pablo Escapa Gordón
Universidad de León
Campus de Vegazana, s/n, 24071
León
pabloescapa@coitt.es

Director: Héctor Alaiz Moretón
Universidad de León
Campus de Vegazana, s/n, 24071
León
hector.moreton@unileon.es

Resumen- **El propósito general de este resumen extendido es explicar el desarrollo, objetivos y las conclusiones del trabajo final de máster denominado: "Ciberseguridad aplicada a la automoción" consistente en: describir los sistemas y redes de computación que equipan los automóviles modernos, la búsqueda de vulnerabilidades, la proposición de posibles soluciones para paliarlas y la realización de diversas pruebas de concepto en entornos reales.**

Los sistemas descritos son la base del futuro coche autónomo por eso debemos trabajar en securizar e intentar minimizar o mitigar los posibles ataques a los que puedan ser sometidos.

***Index Terms*- Car Hacking, hacking can bus, Smart car, ECU.**

Tipo de contribución: *Premio TFM RENIC*

I. Introducción

En la actualidad los automóviles han dejado de estar formados solo por elementos mecánicos y complejos sistemas hidráulicos, dando paso a la introducción de la electrónica moderna combinada con la incorporación de decenas de unidades y redes propias. Esto propicia un nuevo escenario en el cual todo está comandado por líneas de código informático, en la actualidad un vehículo tiene más de 100 millones de estas, la existencia de tal magnitud propicia la presencia de multitud de bugs y errores en el software que pueden ser explotados como vulnerabilidades.

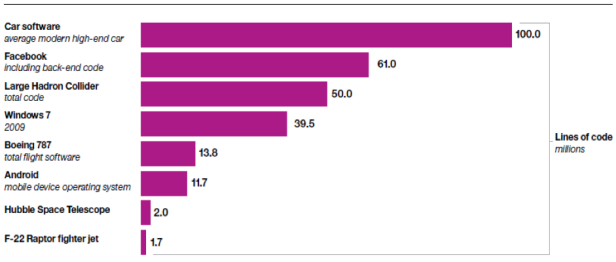


Fig. 1. Lineas de código IBM [1]

II. Novedad del Trabajo y aportación a la ciberseguridad

La principal aportación del trabajo consiste en la fusión de dos mundos aparentemente antagónicos, como son la automoción y la ciberseguridad, una vez repasado el estado del arte pudimos concluir que los automóviles no dejan de estar comandados por centralitas electrónicas basadas en miles o millones de líneas de código fuente, redes de comunicaciones, sistemas criptográficos, apps para dispositivos móviles e incluso sistemas telemáticos que ejercen funciones de routers y/o firewalls.

El TFM desarrollado es una completa guía en castellano que muestra los diferentes tipos de ataques cibernéticos que puede sufrir un vehículo.

En estos momentos la aparición de los vehículos conectados y/o con funcionalidades autónomas va a causar la necesidad de especialistas para securizar dichos sistemas, dado que ejecutan acciones críticas; y por otra parte los vehículos normalmente son una de las grandes inversiones que acometemos y por lo tanto requerimos de que sean un productos seguros y confiables.

III. Objetivos

1. Analizar la arquitectura hardware y software de los vehículos actuales.
2. Conocer cómo aplicar directivas de seguridad sobre los sistemas existentes.
3. Descubrir el estado actual de la ciberseguridad en los sistemas utilizados en la automoción, así como los riesgos reales y sus posibles consecuencias.
4. Demostrar la vulnerabilidad del can-bus y sistema de radio frecuencia.
5. Demostrar los fallos de seguridad existentes mediante demostraciones prácticas.
6. Identificar necesidades del mercado.
7. Intentar aportar solución a los fallos detectados.

IV. Resumen ejecutivo

El objetivo principal del trabajo es conocer la ciberseguridad que se aplica a los automóviles, comenzaremos por describir los sistemas y las redes de comunicación que equipan estos, incluyendo las tecnologías IoT (Internet of Things) que son un valor destinado a mejorar la experiencia de los conductores y acompañantes

Como podremos comprobar los autos están equipados por computadoras las cuales son tan vulnerables como cualquier otra, en la actualidad los sistemas y las comunicaciones de los vehículos no disponen de una seguridad optima dado que muchos de ellos han sido fabricados hace años y sería muy costosa una actualización además de que en el momento del diseño su ciberseguridad no era un requisito prioritario.

Un automóvil actual cuenta cantidad de líneas de código, generando una gran información de datos en cada instante, temporal, decenas de unidades, miles de señales y unidades de control se encargan de gestión de todas las funciones.

Dichos componentes sustentan la actual operatividad de los vehículos.

Debemos tener en cuenta que cuando hablamos de "hackear un vehículo". Erróneamente pensamos exclusivamente en sus sistemas antirrobo, cuando

realmente estos suponen una parte muy pequeña en comparación con el resto de componentes. La realidad actual es casi cualquier sistema del vehículo es proclive a ser manipulado, tanto de forma local como telemática, lo cual puede provocar graves problemas a los fabricantes y usuarios.

Las redes y elementos que conforman un vehículo pueden sufrir ataques comunes de los computadores modernos como Man in The Middle, desbordamiento de buffer, repetición, sustitución, denegación de servicio y otros específicos.

Todos los sistemas de los que dispone un automóvil deben ser seguros, especialmente los relacionados con el control del mismo, además deben ofrecer gran confianza al usuario. A través de este trabajo se muestran los diferentes sistemas que equipa un automóvil moderno, sus principales fallos de seguridad e intentaremos promover algunas medidas para contrarrestar o paliar los mismos. Nuestro estudio está basado en la utilización de métodos bibliográficos y estudios propios.

Para conseguir mejorar la seguridad se necesita una implicación total de todos los elementos y/o integrantes que forman cada sistema (fabricantes, aplicaciones, componentes de automoción, sistemas de diagnóstico, sistemas de radar, etc.) teniendo en cuenta como se explica durante el desarrollo que las soluciones aportadas deben mantener el actual sistema de diagnosis sin restringir su acceso a determinadas funciones.

Gracias a la informática moderna que nos ofrece soluciones y herramientas de seguridad, incluyendo técnicas criptográficas, podemos obtener medios óptimos y adaptados a este mercado.

En la actualidad existen pocos resultados de investigación acerca del tema propuesto cuando por el contrario es un mercado en plena expansión dado que los sistemas de ayuda a la conducción están sufriendo una gran penetración en los mercados actuales.

El trabajo desarrollado muestra varias pruebas de concepto sobre eventos reales siendo la mejor muestra para demostrar la hipótesis a partir del estudio realizado, dichos test se han efectuado como demostraciones utilizando recursos de hardware y software en su mayoría específicos del mundo de la automoción y sus principales fabricantes, esta base permite plantear desarrollos mucho más complejos. Como resumen final se dan las bases para la creación del "vehículo ciberseguro" así como establecer los estándares en cuanto a la implementación de soluciones y aplicaciones necesarias para mejorar el ecosistema actual, también se hace referencia a la creación de la nueva red del transporte basada en V2X necesaria para el desarrollo de los nuevos equipamientos de los vehículos.

La incursión del vehículo eléctrico junto con el autónomo propicia que actualmente el campo de investigación propuesto sea necesario e imprescindible para el desarrollo de las tecnologías venideras.

V. VEHÍCULO CIBERSEGURO

En la siguiente figura podremos ver que sistemas necesitan los futuros vehículos para que puedan protegerse de los atacantes: deberán esquiparse con sistemas de IDS/IPS sobre el can-bus además de mejorar este protocolo, tendrán que contar una correcta segmentación de redes, software con función de detección de manipulación con patrones heurísticos, zonas DMZ, actualización de sistemas inalámbricos, verificación de apps, V2X, configuraciones seguras de sistemas telemáticos… todo ello basado en técnicas informáticas.

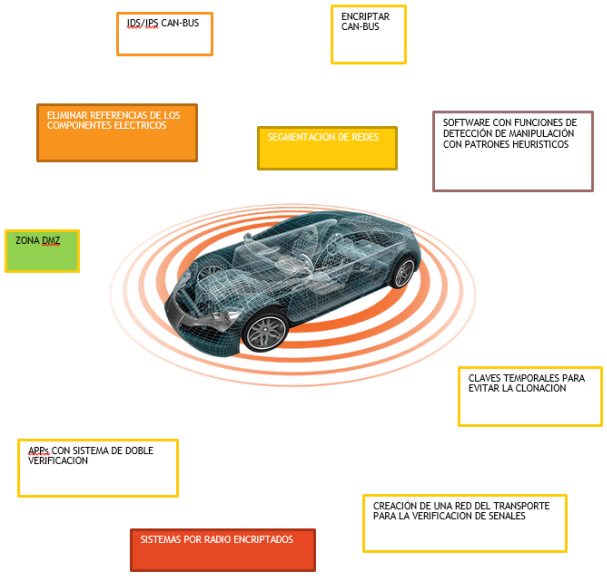


Fig. 2. Vehículo ciberseguro

VI. CONCLUSIONES

Los méritos más relevantes del trabajo son que:

1. Se ha demostrado la vulnerabilidad de los vehículos.
2. Se han analizado las diferentes redes que equipan así como sus principales componentes
3. Se ha creado un listado que incluye los principales fallos de seguridad.
4. Se ha justificado la necesidad de securizar los automóviles para poder madurar la tecnología del coche autónomo.
5. Se han implementado diferentes ataques en entornos reales y pruebas de concepto.
6. Se ha comprobado que existe la tecnología para implantar soluciones en los vehículos.

AGRADECIMIENTOS

A todas las personas que colaboraron en el desarrollo del mismo destacando al director D. Héctor Alaiz Moretón y D. Jorge Escapa Gordón.

REFERENCIAS

[1] "Ibm src codes : How to send money," 2014. [Online]. Available: http://alipotec.ga/tejy/ibm-src-codes-1293.php. [Accessed: 20-Feb-2018].

.

Índice de Autores

Patrocinadores

Patrocinadores Científicos