ESCUELA POLITÉCNICA

UNIVERSIDAD DE EXTREMADURA

ESCUELA POLITÉCNICA

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

Análisis y caracterización de herramientas de seguridad en un entorno empresarial y creación de nuevas firmas para detección y control de acceso a aplicaciones.

UNIVERSIDAD DE EXTREMADURA

ESCUELA POLITÉCNICA

MÁSTER EN INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

Análisis y caracterización de herramientas de seguridad en un entorno empresarial y creación de nuevas firmas para detección y control de acceso a aplicaciones.

Autora: Guadalupe Barba Vázquez

Tutor: Rafael Martín Espada

Resumen

En este trabajo vamos a tratar sobre el análisis y caracterización de herramientas de seguridad, concretamente aquellas que se agrupan en distribuciones que consolidan módulos específicos de un entorno de seguridad en un entorno empresarial, como es el caso de Security Onion. Además, se realizarán firmas TLS y DNS para la detección y control de acceso a aplicaciones de redes sociales y proxys. El objetivo concreto es particularizar y extender el uso de estas herramientas para complementar la seguridad a los casos de comportamientos voluntarios de los usuarios inducidos por aplicaciones de uso común en dichos entornos.

Se estudiarán los tipos de seguridad y el sistema de detección de intrusos(IDS) centrándonos en su clasificación, su funcionalidad y su arquitectura; además se estudiarán las herramientas IDS más utilizadas (Snort, Suricata, Bro, OSSEC). También se analizará la gestión de eventos con herramientas SIEM, en las que se encuentran Squert, Kibana, Snorby, Sguil, ELSA y Splunk.

Llevaremos a cabo la instalación de Security Onion en modo evaluación, que es la forma más sencilla, y en una arquitectura de referencia de producción, en la que se confugurarán varias máquinas, unas dispuestas como sonda y una de ellas realizando la función de máster,. Esta última recolectará toda la información enviada desde de las máquinas que actúan como sondas.

En este entorno de producción se establecerán las comunicaciones entre las máquinas sondas y el determinado como máster mediante túneles VPN, de forma que se garantice la seguridad de estos datos, altamente sensibles e importantes para los agentes maliciosos. Adicionalmente, se instalará un agente OSSEC en los ordenadores

personales y servidores de la empresa para monitorizar todos los eventos que se sucedan en los mismos, como puede ser el hecho de que un usuario instale un programa o aplicación cuyo uso no esté permitido por la gerencia de la propia empresa.

Como innovación de este proyecto se han implementado herramientas adicionales, como los sistemas de monitorización Trisul o Ntop, e integrándolas con Security Onion , se ha logrado disponer de un mayor control de los eventos que ocurren en la red empresarial que está siendo monitorizada.

Finalmente, se han estudiado y analizado la creación de nuevas firmas TLS y DNS, inexistentes en las listas de firmas más habituales, y estandarizadas con las herramientas. Particularmente importante es la personalización de las mismas, que permite una adaptación refinada de la seguridad a las distintas tipologías de empresas, como se demostrará en los capítulos finales de este documento.

Abstract

In this paper we will discuss the analysis and characterization of tools of security, specifically those that are grouped into distributions that consolidate specific modules of a security environment in a business environment, such as Security Onion. Also this will make TLS and DNS signatures for the detection and control of access to applications social networks and proxies. The specific objective is to particularize and extend the use of these tools to complement security to cases of voluntary behavior of users induced by common applications in such environments.

The types of security and the intrusion detection system (IDS) will be studied focusing on their classification, functionality and architecture; In addition, the most used IDS tools will be studied (Snort, Suricata, Bro, OSSEC). The event management will also be analyzed with SIEM tools, which include Squert, Kibana, Snorby, Sguil, ELSA and Splunk.

We will carry out the installation of Security Onion in evaluation mode, which is the simplest way, and in a production reference architecture, in which several machines will be confgured, some arranged as a probe and one of them performing the master function. The latter will collect all the information sent from the machines that act as probes.

In this production environment, communications will be established between the machines probes and the one determined as a master through VPN tunnels, in such a way as to guarantee the security of these highly sensitive and important data for malicious agents. Additionally, it will install an OSSEC agent on personal computers and servers of the company to monitor all the events that occur in them, such as the

fact that a user installs a program or application whose use is not allowed by the management of the own company.

As an innovation of this project, additional tools have been implemented, such as the monitoring systems Trisul or Ntop, and integrating them with Security Onion, it has been possible to have Greater control of the events that occur in the business network that is being monitored.

Finally, we have studied and analyzed the creation of new TLS and DNS signatures, nonexistent in the lists of most common signatures, and standardized with the tools. Particularly important is the personalization of the same, which allows a refined adaptation of the security to the different types of companies, as will be demonstrated in the final chapters of this document.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

Es posiblemente en estos tiempos, comenzando el siglo XXI, superado en los países llamados occidentales unas mínimas garantías de seguridad personal, cuando el concepto amplio de seguridad de la información copa de preocupaciones a los ciudadanos. Y es que términos como noticias falsas o fake news, hackings, estafas en internet, etc. están inundando las noticias de los medios de comunicación de masas. [1]

Figura 1.1: Marz Zuckerberg antes de declarar ante el Senado de EEUU por la venta de datos personales de millones de usuarios a la empresa Cambridge Analitycs Fuente: Flickr (Etiquetada con derechos de reutilización no comercial)

Posiblemente, el término más espectacular y más oscuro es la palabra hacker, que ha pasado de una definición positiva, en los tiempos en que los estudiantes del MIT se entretenían indagando en los sistemas informáticos de los años 70 para lograr acceder a los sistemas de control de la todopoderosa ATT [2](2bis Ref: Exploding the pone, Author: Phil Lapsley Editor: Barnes Noble), a una definición actual lasa, pero muy seria, refiriéndose a todo aquel que intenta alguna acción maliciosa en un sistema informático ajeno. [3]

También el término virus ha evolucionado, desde aquella demostración de programa auto replicante, como había predicho Von Neumann, desarrollado por Bob Thomas en 1971 y conocido como Creeper (aparecía en la pantalla I´m a creeper. Catch me if you can) [4] (Ref: https://history-computer.com/Internet/Maturing/Thomas.html) y cuando ni siquiera existía la denominación virus, a las actuales nociones más generales de malware, que abarcan todo tipo de códigos no permitidos por el usuario con intenciones diversas y que ya no precisan de un medio de contagio, sino que la propia dinámica de uso de Internet fomenta su propagación.

En todo caso, actualmente los ”hackeos”, soportados por algún tipo de malware, son uno de los términos que más aparecen en las noticias más habituales en los últimos tiempos. Tan sólo en 2018 se estiman alrededor de 65 millones de víctimas en el mundo, cifra que no hace más que seguir aumentando año tras año. [5]

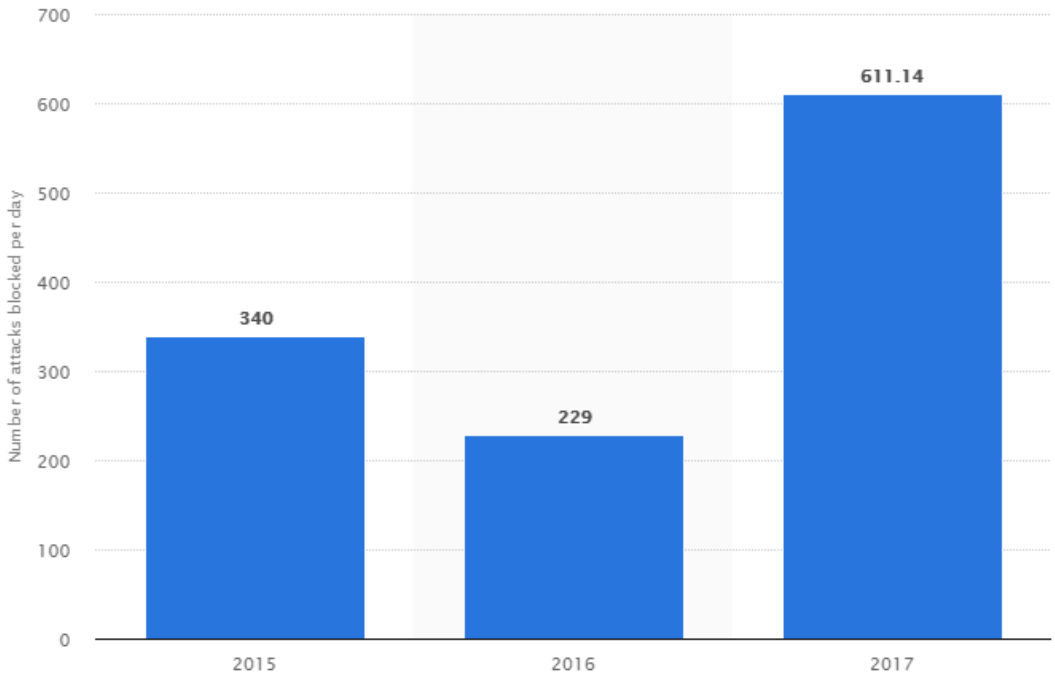


Figura 1.2: Número global de ataques web Fuente: (https://www.statista.com/statistics/494961/web-attacks-blocked-per-day-worldwide)

Este crecimiento se puede observar claramente en la figura 1, donde se muestra una gráfica con el número global de ataques web bloqueados diariamente desde 2015 a 2017. En el año 2017, se bloquearon 611,141 ataques web a diario, frente a los 229,000 ataques bloqueados cada día del año anterior. Como se puede observar, a finales de 2017, el número de ataques web bloqueados era más del doble que al comienzo del año. [6]

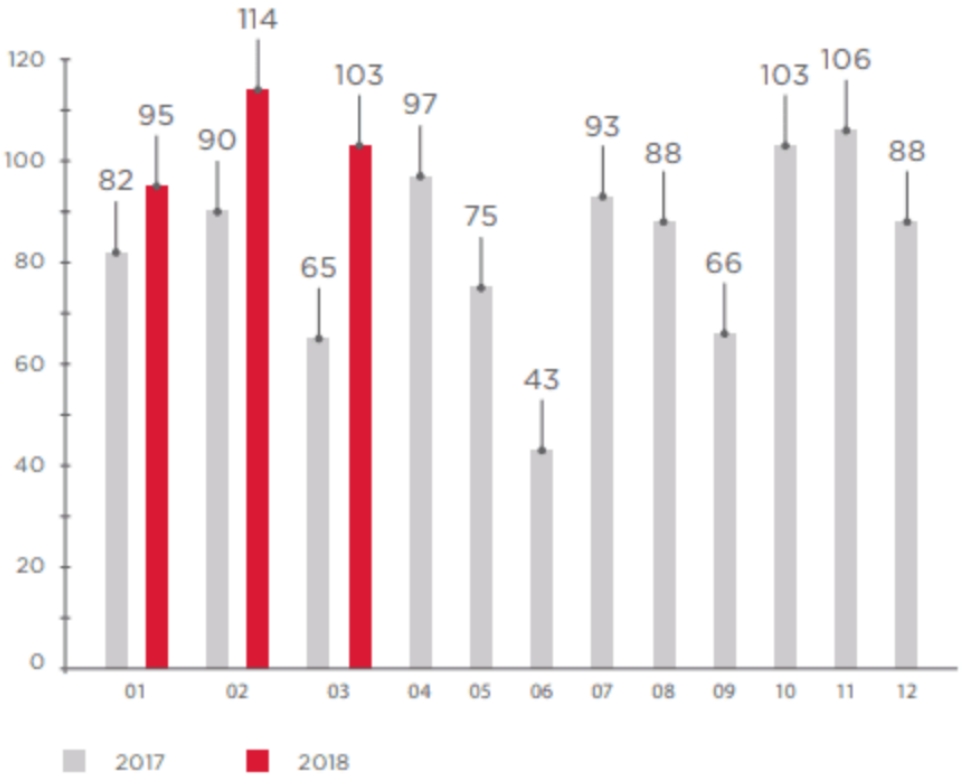


Figura 1.3: Fuente: https://www.ptsecurity.com/ww-en/analytics/cybersecurity-threatscape-2018-q1/

Asimismo, de la gráfica anterior se extrae que en el pasado año 2018 ha aumentado el número de ataques registrados en un 32 %, lo que no permite a ninguna organización relajar los controles de seguridad.

Concretamente una de las noticias más destacadas sobre hackeos en el año 2018, es ha sido la sustracción a la todopoderosa compañía Facebook© de los datos de más de 50 millones de cuentas de usuarios en todo el mundo. [7]

Y es que los ataques contra redes sociales se han convertido en el objetivo prioritario de los agentes maliciosos, debido al aumento de forma exponencial del número de usuarios y el formato de las mismas, que conlleva una problemática relajación a la hora de acceder a sus contenidos. Tan solo en España, el 85,6 % de los internautas utilizan la red social Facebook para comunicarse habitualmente, amén de otras muchas para diferentes servicios. [8]

Figura 1.4: Uso de redes sociales en España Fuente: https://www.trecebits.com/2018/03/06/datos-actualizados-uso-redes-sociales-espana-marzo-2018/

Debido a este espectacular aumento, hasta el punto en que se ha convertido prácticamente en una herramienta de comunicación ubicua, utilizada tanto en el hogar como en la oficina, multidispositivo, generalizada a todo tipo de usuarios, etc. se estima conveniente una monitorización de las condiciones de seguridad de las redes sociales. Y es en este punto desde donde parte este proyecto, de la intención de realizar una monitorización del uso de las principales redes sociales utilizadas en España en un entorno controlado, como puede ser una pequeña empresa; redes como Facebook©, Whatsapp© o Instagram© en las que su simple uso puede conllevar la instalación de malware sin que sea detectado por los clásicos antivirus por provenir de una ejecución consciente de los usuarios.

No permitir utilizar redes sociales en el ámbito empresarial parece totalmente desaconsejable, a tenor de numerosos informes y propuestas de expertos. No fiarse de los empleados resulta a todas luces contraproducente con el rendimiento. [9]

Pero permitir el acceso sin control alguno ya no resulta desaconsejable, parece un suicidio. [10] (https://www.independent.co.uk/life-style/gadgets-and-tech/news/digmine-facebook-messenger-cryptocurrenc
La forma de proteger los accesos a redes sociales exige la monitorización activa de los accesos a las mismas y la ejecución dinámica de salvaguardas. Para este proyecto se pretende elaborar un sistema de alerta de los accesos a elementos sospechosos de las redes sociales, de forma que el responsable de seguridad de una organización tenga información precisa de sus usos y riesgos.

Para dicha monitorización se utilizarán herramientas de seguridad informática (Computer Security) , encargada de mantener la integridad, la confidencialidad y la disponiblidad de los datos de los usuarios en una computadora o circulante entre los sistemas, como subconjunto de la más amplia disciplina Seguridad de la Información (Information Security) , que cubre aspectos más amplios como la gobernanza o las políticas de seguridad de una organización [11](Ref: https://nvlpubs.nist.gov/nistpubs/ir/2013/nist.ir.7298r2.pdf) . Conviene destacar en este punto que para 2019 se prevé un aumento del 12 % del gasto que se destina a ciberseguridad en el mundo [12], lo que evidencia la importancia que está adquiriendo.

Además, el proyecto propone ampliar el acceso seguro a los usuarios itinerantes de la organización de los recursos empresariales. En efecto, una de las principales tareas de la seguridad informática es intentar que las comunicaciones en la red siempre sean seguras para el usuario y que no se puedan interceptar sus datos. Para ello, se aprecia cómo se están cifrando cada día una mayor cantidad de datos, mensajes, archivos, ... Además, también se observa cómo se ha extendido la utilización de protocolos seguros, como https o o TLS, configuraciones de transmisión cifrada como las redes privadas virtuales (VPN) y otros sistemas de securización de los canales de comunicación. Por esta razón, también en este proyecto se implementarán soluciones con sistemas

similares, como certificados SSL y redes privadas virtuales VPN al objeto de garantizar la seguridad de las comunicaciones entre sistemas.

Capítulo 2

Objetivos

El objetivo de este proyecto es detectar y monitorizar el acceso a redes sociales en una red empresarial, con la intención de prevenir el acceso oculto a enlaces o contenidos que los sistemas de seguridad tradicionales no detectan. Como objetivos específicos se encuentran:

- Entender y aplicar la seguridad informática que se caracteriza por la confidencialidad, la integridad y la disponibilidad.
- Monitorización de la red, mediante el uso de la herramienta Security Onion©, lo que deberá llevar consigo un análisis profundo del funcionamiento de la herramienta.
- Conocer los sistemas de detección de intrusos, IDS en su terminología anglosajona (Instrusions Detection Systems), así como las herramientas para gestión de eventos e información de seguridad (conocidos como SIEM, por Security Information and Event Management).
- Introducir nuevas herramientas de monitorización como son Trisul© y Ntop en Security Onion©.
- Creación de firmas, esto es, aquellos patrones de comportamiento que serán identificados como anómalos o maliciosos, en Security Onion© para conseguir nuevas alarmas que puedan interesar para el proyecto.

- Establecer un túnel VPN (Red Privada Virtual) para la comunicación segura entre distintos servidores.
- Generar certificados SSL, con el objeto de asegurar la autenticidad de los accesos al túnel VPN.
- Analizar el modo producción de Security Onion© en el que se estudiará el significado y funcionamiento de los elementos máster y sonda.

Capítulo 3

Estado del Arte

3.1. SEGURIDAD

Los sistemas informáticos se diseñan generalmente para realizar funciones concretas y específicas y así proporcionar servicios esenciales o importantes para una organización. Realmente pueden contener y procesar datos importantes, incluso vitales, que comprometen a los usuarios y a los dueños de los datos. Se han convertido en una parte esencial de los negocios y de la administración moderna y se espera de ellos un buen rendimiento, un funcionamiento correcto y una alta disponibilidad, cada vez más exigente si cabe por la clara habituación a los sistemas informáticos de la población.

Cuando citamos el término sistema seguro, éste se utiliza para indicar que un sistema se ha diseñado para estar a salvo de ataques o fallos, algo que comienza a ser una característica mínima exigible a cualquier elemento digital. [13] El paso dado para proteger contra ataques o fallos a estos sistemas es establecer las medidas de seguridad adecuadas y proporcionales [14]. Estas incluyen desde controles de validación de datos de entrada a cajas de seguridad resistentes al fuego para proteger diferentes medios de almacenamiento y los datos que contienen. La encriptación de datos en los circuitos de comunicación, y los lectores de credenciales personales son otros ejemplos de medidas de seguridad.

Hay un sinnúmero de amenazas, algunas obvias y otras que no se descubren hasta que ya es, desgraciadamente, demasiado tarde. Por ejemplo, una fuente de alimentación de un ordenador puede fallar de vez en cuando, y si no tenemos una medida de seguridad que evite el apagón repentino del ordenador, podemos llegar a perder datos. Los programas de software también pueden contener errores no detectados. Y de forma idéntica, soportes de almacenamiento como los discos o las cintas, también se pueden dañar y provocar una pérdida importante de datos vitales para una organización. [15]

En una primera clasificación, se pueden identificar las amenazas dentro de dos tipologías:

- Accidentales: factores humanos, fallos en los sistemas de procesamiento y desastres naturales.

- Deliberadas: virus o código malicioso, robo de información, fraudes basados en el uso de ordenadores, suplantación de identidad, denegación de servicios, ataques a fuerza bruta, alteración y divulgación de la información, espionaje, etc.

Si bien el primer grupo son amenazas estables, digamos estáticas, las del segundo grupo tienen como límite la creatividad humana y el carácter malicioso de algunos individuos.

3.1.1. Tipos de seguridad

Las medidas de seguridad que se aplican en una organización se pueden agrupar bajo cuatro tipologías distintas, en función de los activos que protegen y la forma de hacerlo.

Figura 3.1: Clasificación de la seguridad informática.
Fuente: http://alaveraderuger8.blogspot.com/2017/05/diferencias-entre-seguridad-activa-y.html

Seguridad activa

La seguridad activa es aquella que se utiliza día a día para evitar cualquier tipo de ataque, esto es, medidas preventivas para evitar los incidentes de seguridad antes de que estos ocurran. Si bien existen infinidad de recomendaciones, dependiendo del sistema que estemos utilizando, evidentemente no se puede tratar de la misma manera un servidor que un equipo cliente. No obstante, se puede realizar una lista de las más comunes [16]:

- Utilizar usuarios que no sean administradores, para abrir una aplicación como puede ser word o para navegar por internet.

- Tener contraseñas fuertes. Existen virus que intentan averiguar las contraseñas de administrador, de forma que, si se le facilita el acceso con contraseñas sencillas de adivinar, podría bloquearnos o incluso secuestrarnos todo nuestro sistema. Una contraseña fuerte es recomendable que contenga o se componga de una frase con letras mayúsculas y minúsculas, así como algún número o símbolo adicional. Un hándicap de este tipo de contraseñas es que son difíciles de recordar incluso para el propietario del sistema, lo que desanima sobremanera a dichos usuarios, que acaban cayendo en la mala práctica de utilizar una contraseña común para

todos los sistemas o una sencilla de recordar, representación de datos públicos y accesibles para cualquiera.

- Disponer de la última actualización del antivirus y del sistema operativo.
- Efectuar copias de seguridad.
- Encriptar aquellos datos importantes.

Seguridad pasiva

La seguridad pasiva en el ámbito informático comprende aquellas medidas orientadas a minimizar los daños causados por un incidente de seguridad, ya sea por accidente o por un malware en los sistemas (medidas correctoras). Igual que con la seguridad activa, existen varias prácticas para cada situación [17].

Algunas de ellas son:

- Desconectar la máquina de la red hasta que se encuentre una solución.
- La realización de copias de seguridad de los datos y del sistema operativo en más de un soporte y en distintas ubicaciones físicas.
- Creación de particiones lógicas en el disco duro para poder almacenar archivos y copias de seguridad (back-up) en una unidad distinta a la del sistema operativo.

Una clasificación distinta es la que diferencia entre seguridad lógica y física, centrada en los activos que protege.

Seguridad física

La seguridad física trata de la protección de los sistemas ante amenazas físicas. Consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas, ante amenazas a los recursos e informaciones confidenciales. Desastres naturales, sabotajes internos o externos, etc., forman parte de este tipo de seguridad. [18]

Seguridad lógica

La seguridad lógica informática es una referencia a la protección por el uso de software (y los datos, todo aquello que no es físico, incluso la reputación) en una organización, e incluye identificación de usuarios y contraseñas de acceso, autenticación, derechos de acceso y niveles de autoridad [19]. Estas medidas sean para asegurar que sólo los usuarios autorizados son capaces de realizar acciones o acceder a información en una red o un equipo concreto.

Una brecha de seguridad lógica informática afecta a los datos y el software sin tener que afectar necesariamente a los elementos físicos, o más concretamente el hardware. El daño frecuentemente permanece invisible hasta que algún usuario de la organización intenta procesar o visualizar los datos afectados.

La seguridad lógica incluye, entre otros elementos que la comprometen, los siguientes:

- Los virus.
- Programas no testeados.
- Errores de usuario.
- Error del operador.
- Mal uso del ordenador.
- Fraude informático.
- Investigación de accesos no autorizados internos.
- Accesos no autorizados externos.

Uno de los problemas con cualquier violación de la seguridad lógica informática es que el daño resulta invisible y su extensión es desconocida hasta que se toma conciencia de su existencia. El coste de investigación de las causas y efectos probablemente sea alto.

Esto es particularmente cierto con las infecciones por virus. Un tipo de malware muy reconocido son los virus, que tienen como objetivo alterar el funcionamiento de un equipo informático sin el permiso de su propietario. Sus consecuencias son muy variadas, desde la destrucción de datos hasta ser simplemente una "broma". En su comportamiento, buscan infectar archivos ejecutables, agregando parte de su código al código del archivo "víctima"; también usan esta técnica como forma de propagación. [20]

Hay infecciones maliciosas que pueden borrar (o, peor aún, corromper) los datos de todo un disco. El virus puede ser transferido a cualquier otra unidad que entra en contacto con un PC infectado sin que se perciba. Cada PC que entra en contacto con esa unidad infectada tiene que comprobarse, y cada unidad adicional que se haya instalado en cualquiera de esos ordenadores infectados, también tendría que comprobarse, y después todas las unidades que hayan sido instaladas en esos ordenadores, y así sucesivamente. Pueden llegar a ser miles de ordenadores y unidades las que tienen que ser revisadas en una organización grande. Y eso sin tener en cuenta que además los virus también pueden ser transmitidos a otros ordenadores a través de la red.

Entre los elementos que forman parte de la arquitectura de seguridad de una organización para conferir seguridad lógica, se pueden citar:

- Firewalls
- Antivirus
- IPS
- IDS (Intrusion Detection Systems)
- Web Application Firewall (WAF)

En el siguiente apartado se aborda el análisis de los IDS, como elementos indispensables para la monitorización de la seguridad lógica de los sistemas pertenecientes a una red de datos corporativa.

3.2. SISTEMA DE DETECCIÓN DE INTRUSOS (IDS)

Un sistema de detección de intrusos es un programa de detección de accesos no autorizados a un ordenador o a una red, lo que extiende enormemente las capacidades y funcionalidades, al proyectarse sobre un escenario dinámico y de límites difusos. Una de las definiciones de la detección de intrusos es la propuesta por el NIST (Institute of Standards and Technology)[21], que la define como el proceso de monitorización de eventos que suceden en un sistema informático o red y el análisis de dichos eventos en busca de signos de intrusiones. Hoy en día existen multitud de fabricantes de IDS, lo que escenifica la progresiva implantación de este tipo de herramientas[22]. Más allá de su éxito como elemento de seguridad, es obvio que los IDS confieren a los administradores de redes y sistemas de un punto de monitorización centralizada y lo que es más importante, la sensación de control. Un IDS es un sistema que se suele instalar en puntos de concentración de tráfico, ya sea el punto de acceso a Internet de la empresa o algún punto de red intermedio donde se pueda analizar todo el tráfico de red o el de acceso a un host, en función del tipo de IDS que se desee instalar, que se detallará más adelante.

3.2.1. Clasificación

Existen tres tipos de IDS según las fuentes de información que se utilicen:

- HIDS (HostIDS): el principio de funcionamiento de un HIDS depende del éxito de los intrusos, que generalmente dejaran rastros de sus actividades en el equipo atacado cuando intentan adueñarse del mismo para ejecutar su estrategia maliciosa. El HIDS intenta detectar tales modificaciones en el equipo afectado y hacer un reporte de sus conclusiones. Además:
    - Monitoriza múltiples fuentes de datos, incluyendo sistemas de archivos tipo meta-data, y archivos tipo log.

- Actúa como un firewall, bloquea o acepta conexiones o logins de acuerdo con políticas previamente establecidas.

Los HIDS tienen una ventaja sobre el NIDS que se describe a continuación en el hecho de que pueden detectar paquetes de red anómalos originados dentro de la organización, puesto que analiza el comportamiento de los sistemas, de sus logs y ficheros y no se limita al tráfico de red. HIDS también puede identificar el tráfico malicioso que se origina en el propio host, cuando éste ha sido infectado con algún malware que intenta propagarse a otros sistemas. [23]

- NIDS (NetworkIDS): un IDS basado en red, detectando ataques a todo el segmento de la red. Su interfaz debe funcionar en modo promiscuo capturando así todo el tráfico de la red. Se implementa un sistema de detección de intrusos en la red (NIDS) en un punto o puntos estratégicos dentro de la red, donde puede monitorizar el tráfico entrante y saliente hacia y desde todos los dispositivos en la red [24].

  NIDS se basa para la detección de intrusos en las anomalías de red, como pueden ser el tamaño de los paquetes que circulan, destinos, protocolos, .. ; el problema es que resulta difícil determinar que tráfico es el normal. También está basado en firmas o patrones que modelan los vectores de ataque conocidos.

- IDS Híbridos: Los sistemas híbridos recogen lo mejor de ambos tipos HIDS y NIDS. Permiten una detección local de los sistemas y un sensor en cada segmento de red se encarga de la vigilancia. De esta forma complementan las necesidades HIDS con las del NIDS, permitiendo el aprovechamiento de las ventajas de ambas arquitecturas. [25]

Aunque los sistemas de detección de intrusos monitorizan las redes en busca de actividad potencialmente maliciosa, también son propensos a falsas alarmas (falsos positivos), definidas como aquellas alertas que no son incidentes de seguridad, sino desviaciones del comportamiento de la red con respecto al patrón preestablecido en el sistema IDS. En consecuencia, las organizaciones necesitan ajustar sus productos

IDS cuando los instalan por primera vez. Y esto significa configurar correctamente sus sistemas de detección de intrusos para determinar el tráfico normal en su red en comparación con el que se deriva de una actividad potencialmente maliciosa.

Históricamente, los sistemas de detección de intrusos se clasificaron como pasivos o activos; un IDS pasivo que detectó actividad maliciosa generaría entradas de alerta o registro, pero no realizaría ninguna acción. Un IDS activo, a veces llamado sistema de prevención y detección de intrusiones, generaría alertas y entradas de registro, pero también podría configurarse para realizar acciones concretas, como bloquear determinadas direcciones IP o cerrar el acceso a recursos restringidos.

3.2.2. Funcionalidad

Los principales métodos utilizados por NIDS para informar y bloquear intrusiones son: [26]

- Reconfiguración de dispositivos externos (firewalls o ACL en routers): Comando enviado por el NIDS a un dispositivo externo (como un filtro de paquetes o un firewall) para que se reconfigure inmediatamente y así poder bloquear una intrusión. Esta reconfiguración es posible a través del envío de datos que expliquen la alerta (en el encabezado del paquete).

- Envío de Trap SNMP a un hipervisor externo: Envío de una alerta (y detalles de los datos involucrados) en forma de mensaje SNMP a una consola externa como HP Open View, Tivoli, Cabletron, Spectrum, etc.

- Envío de un correo electrónico a determinados usuarios: Envío de un correo electrónico a uno o más buzones de correo para informar sobre una intrusión real.

- Registro del ataque: Se guardan los detalles de la alerta en una base de datos central, incluyendo información como el registro de fecha o la dirección IP del intruso.

- Almacenamiento de paquetes sospechosos: Se guardan todos los paquetes originales capturados o los paquetes que dispararon la alerta.

- Ejecución de una aplicación: Se lanza un programa externo que realice una acción específica (envío de un mensaje de texto SMS o la emisión de una alarma sonora).

- Envío de un ResetKill": Se construye un paquete de alerta TCP para forzar la finalización de una conexión (sólo válido para técnicas de intrusión que utilizan el protocolo de transporte TCP).

- Notificación visual de una alerta: Se muestra una alerta en una o varias de las consolas de administración.

3.2.3. Arquitectura

Originalmente los IDS se diseñaron por cuestiones de auditoría en los entornos financieros. No se conocía qué era lo que se debería detectar, analizar o proteger. Entre 1984 y 1986, Dorothy Denning y Peter Neumann desarrollaron un primer modelo de IDS, un prototipo nombrado como IDES (Sistema Especialista en Detección de Intrusión). [27] Su concepción se basaba en la diferencia en los patrones de uso que exhibían un usuario legítimo y un intruso; patrones basados en variables estadísticas.

Uno de las primeras investigaciones sobre detección de intrusos comienzan en 1980 en un trabajo de consultoría realizado para el gobierno norteamericano por James P. Anderson. [28]

Anderson expuso la idea de que el comportamiento normal de un usuario podría caracterizarse mediante el análisis de su actividad en los registros de auditoría.

En paralelo prácticamente, en 1988, en los laboratorios Lawrence Livermore de la Universidad de California, se ejecutaba el proyecto Haystack para las fuerzas aéreas de EE.UU, [29] aún en vigor en la actualidad . Haystack era el primer IDS comercial (CMDS)que analizaba los datos extraídos de una auditoría y los comparaba

con vectores de ataque predefinidos. Aparecía por tanto el primer sistema de detección de usos indebidos basado en firmas, el tipo de IDS más extendido en el mercado actual.

Ligeramente posteriores, de 1990, son los primeros proyectos de IDS basados en red. Todd Heberlein introduce tal idea y desarrolla NSM (Network Security Monitor)[30] en la Universidad de California.

En esa misma fecha, en Los Alamos National Laboratory de EEUU realizan un prototipo de un sistema experto que monitoriza la actividad de red. Su nombre es NADIR (Network Anomaly Detector and Intrusion Reporter). [31]

CIDF

El Marco de Detección de Intrusos Común fue un primer intento de estandarización de la arquitectura de un IDS. No logró su aceptación como estándar, pero estableció un modelo y un vocabulario para discutir sobre las intrusiones. Mucha gente que trabajó en el proyecto original está fuertemente involucrada en los esfuerzos del Grupo de Trabajo de Detección de Intrusos (Intrusion Detección Working Group, IDWG) del Internet Engineering Task Force (IETF). El grupo de trabajo del CIDF se creó originalmente en enero de 1997 a petición de Teresa Lunt en la organización DARPA con el objetivo de desarrollar estándares que permitieran a los diferentes IDS interoperar y se pudieran reutilizar sus componentes. Preocupada por los diversos esfuerzos de los sistemas de detección de intrusiones en diferentes plataformas [32], propuso un modelo para utilizar y reutilizar diversos IDS de forma conjunta. Como ya se ha comentado, CIDF tiene como objetivo principal el desarrollo de un medio por el cual los sistemas y componentes de detección, análisis y respuesta ante intrusiones desarrollados de forma independiente pueden compartir información y, por lo tanto, interoperar. El principal elemento de CIDF es el Lenguaje de Especificación de Intrusiones Común (CISL), en el que se pueden generar y codificar expresiones sobre ataques, anomalías y prescripciones de respuesta. En este trabajo, discutimos el desarrollo y la estructura del lenguaje.

Autopost de AusCERT

A diferencia de CIDF/CISL, el CERT australiano (AusCERT) desarrolló un sistema de trabajo sencillo que permitía que se analizara y se agregara un informe en una base de datos con tan solo un par de líneas de Perl. La forma que toma al informar de un incidente podría ser la siguiente:

Source: 216.36.45.84

Ports: tcp 111

Incident type: Networkscan

re-distribute: yes

timezone: GMT + 1300

reply: no

Time: Web 15 Mar 2000 at 14:01 (UTC)

Este sistema tiene una alta interoperabilidad y es muy sencillo de construir y analizar. El problema es que los analistas a menudo necesitan un gran nivel de detalle (una fidelidad alta) acerca del evento, por ejemplo, para análisis forense, y en este modelo toda esa información se perdería. La solución de interoperabilidad que parece ser la elegida es IDWG, que detallaremos a continuación. Según progresan los trabajos en este sentido, parece ser que la fidelidad de los datos se ha convertido en el indicador más importante que se pretende conseguir.

Arquitectura de IDWG (Intrusion Detection Working Group)

El IETF rechazó el enfoque de CIDF, seguramente por antipatía a CISL, debido a su complejidad, y creó un grupo de trabajo llamado IDWG (Intrusion Detection Working Group) que tenía como objetivo el de definir formatos y procedimientos de intercambio de información entre los diversos subsistemas del IDS.

Los resultados de este grupo de trabajo serán:

- 1. Documentos que describan los requerimientos funcionales de alto nivel para la comunicación entre sistemas de detección de intrusos y entre los sistemas de detección de intrusos y sus sistemas de gestión.

- 2. Un lenguaje común de especificación que describa el formato de los datos.

- 3. Un marco de trabajo que identifique los mejores protocolos que se pueden usar para la comunicación entre los IDSs y que defina como se mapean en éstos lo formatos de datos.

Podemos concluir que prácticamente todos los IDS en la actualidad observan el estándar IDWG, aunque la profusión de datos, la aparición de nuevos vectores, etc. hace que la integración no sea inmediata. Abordaremos en los próximos apartados las principales herrmientas IDS de cara a analizar la posible integración de algunas de ellas. [32]

3.2.4. Herramientas IDS

[33]

Snort

Snort es un “sniffer” de software libre construido sobre libpcap y tcpdump, que permite capturar todo el tráfico que llega al equipo donde está instalado. Snort está diseñado para ser preciso en el registro de actividades en la red y está en continua búsqueda de posibles coincidencias entre el flujo de datos y los ataques que tiene registrados en base a diferentes reglas.

Snort tiene una base de datos de ataques que se está actualizando constantemente, que, además, permite añadir o actualizar a través de Internet. Los usuarios pueden crear ’firmas’ basadas en las características de los nuevos ataques de red y enviarlas a la lista de correo de firmas de Snort17. El dinamismo de esta comunidad ha convertido a Snort en uno de los IDS más populares, actualizados y robustos.

Otra de las características más importantes de Snort es que es utilizado por los principales fabricantes de IDS/IPS, pudiendo utilizarse sus firmas en casi cualquier dispositivo.

Suricata

Suricata es el nombre de un proyecto de software libre, desarrollado por la comunidad OISF (Open Information Security Foundation). Es un motor de detección de amenazas basado en un conjunto de reglas IDS/IPS para monitorizar el tráfico en la red y proporcionar alertas al administrador del sistema cuando ocurre un evento que considera sospechoso. Está diseñado para ser compatible con otros componentes de seguridad existentes y, además, acepta llamadas desde otras aplicaciones.

Suricata puede funcionar como IDS de tiempo real, IPS, Intrusion Prevention System o Sistema de Prevención de Intrusos, monitorizador de seguridad de la red (NSM) y como analizador de ficheros pcap (ficheros con capturas de tráfico).

El funcionamiento para analizar la red se basa en reglas y firmas, aunque también dispone de soporte para crear nuevos scripts mediante un lenguaje sencillo de programación denominado LUA. Dispone de entradas y salidas estandarizadas a formatos estándar para serializar datos que pueden ser fácilmente tratados y entendidos por humanos, como YAML que le permiten integrarse fácilmente con otras herramientas como SIEM o bases de datos.

Al involucrar a la comunidad de código abierto y el conjunto de los recursos más importantes de reglas IDS/IPS disponible, OISF ha construido el motor Suricata para simplificar el proceso de mantenimiento del nivel de seguridad óptimo. A través de asociaciones estratégicas, OISF está aprovechando la experiencia de Amenazas Emergentes19, Emerging Threats, como centro de Investigación Abierta de Seguridad, y otros recursos importantes en la industria para proporcionar las reglas más actualizadas y completas disponibles.

Bro

Bro es otra herramienta que funciona como IDS/IPS, debido a sus características de análisis de red, al igual que Snort y Suricata. Se basa en un potente motor de análisis que permite un alto rendimiento en la monitorización de la red, analiza protocolos, y la información de la capa de aplicación en tiempo real.

Al igual que otras herramientas, Bro también hace uso de la librería libpcap para su funcionamiento y además es capaz de funcionar en varias redes de manera simultanea.

Además de la portabilidad adquirida mediante el uso de libpcap, Bro también puede ser una herramienta de red pasiva, lo que significa que puede actuar supervisando una red sin que sea un nodo con una dirección IP asignada.

OSSEC

OSSEC es un IDS basado en Hosts (HIDS). Realiza análisis de logs, comprobación de la integridad, la supervisión del registro de eventos de Windows, detección de rootkits, alerta basada en tiempo y respuesta activa. Proporciona detección de intrusiones para la mayoría de sistemas operativos, incluyendo Linux, OpenBSD, FreeBSD, OS X, Solaris y Windows.

OSSEC tiene una arquitectura centralizada, multiplataforma que permite a varios sistemas ser controlados y manejados fácilmente.

OSSEC se basa en nombrar a cada host como server o sensor, según sean sus características. Será necesario un sensor en cada zona que se quiera inspeccionar la red en busca de amenazas, y un servidor al menos para poder leer los datos que llegan de los sensores.

Snort y Suricata, son unos de los sistemas de detección de intrusos más utilizados, son un NIDS de código abierto, de libre acceso y ligero, que se utiliza para detectar amenazas emergentes. Snort se puede compilar en la mayoría de los sistemas operativos Unix o Linux, y también hay una versión disponible para Windows.

En este proyecto trabajaremos con Snort y Suricata por ser herramientas ampliamente extendidas en entornos de seguridad por empresas especializadas.

En el siguiente apartado vamos a detallar la forma de actualizar las firmas o patrones, elemento muy importante a la hora de seleccionar los sistemas de seguridad. Téngase en cuenta que la sujeción a un fabricante puede convertirse en una opción excesivamente onerosa para empresas de tamaño reducido o medio, pero que son idénticamente objetivo de actividades maliciosas que las grandes corporaciones. La

actualización continuada de firmas se convierte en la única forma de mantener actualizada las políticas y prevenciones de seguridad y tener control sobre el entorno.

3.2.5. Actualización de firmas

[34]

Hoy en día, prácticamente cada día, nos encontramos con muchos fallos de seguridad en el software y esto conduce a que ciberdelincuentes intenten aprovecharse para realizar nuevos ataques a nuestros sistemas informáticos, estos, no detectaran los ataques y serán totalmente vulnerables.

Por lo cual, es muy importante tener todos nuestros IDS (Sistemas de detección de intrusos) completamente actualizados. En este trabajo utilizaremos una herramienta para la actualización de reglas de Snort, llamada PulledPork.

PulledPork es nada menos que un script escrito en Perl que descarga, instala y actualiza las reglas de múltiples sitios que serán utilizadas por el IDS Snort.

Podemos acceder a su repositorio de github oficial desde el siguiente enlace: https://github.com/shirkdog/pulledpork

Existen varios rule sets que pueden dividirse en tres grandes grupos: [35]

- Las community rules son las reglas que “por defecto” de Snort. Se actualizan con baja frecuencia y tienen un conjunto de reglas muy básico. Cualquier persona puede descargarlas para uso no comercial.

- Reglas VRT, VRT (Vulnerability ResearchTeam) es un grupo de profesionales en seguridad respaldado por Snort cuyo fin es encontrar y estudiar amenazas de red.

  Desarrollan un conjunto de reglas complejo, de alta calidad y que se actualiza tan pronto como aparece una nueva amenaza.

  Las reglas VRT, de uso propietario, las puede descargar cualquier usuario que se registre de forma gratuita en la web de Snort. Así mismo, un usuario puede

comprar una suscripción a las reglas de VRT. Diferencias entre versión gratuita y de pago:

- La versión gratuita recibe las actualizaciones 30 días más tarde que los usuarios suscriptores. Debemos ser conscientes de lo que este retraso significa, evidenciando un nuevo modelo de negocio de las empresas de desarrollo de código abierto.
- Los usuarios suscriptores pueden contactar con el VRT para obtener soporte (falsos positivos/negativos, ayuda en la creación de reglas, etc...)

- Por último, existen distribuciones "no oficiales". Son distribuciones de reglas realizadas por terceros, ya sea de forma gratuita o de pago.

  Cualquiera puede programar sus propias reglas a través del lenguaje de reglas de Snort.

3.2.6. Gestión de eventos con herramientas SIEM

SIEM [36](información de seguridad y gestión de eventos), es una tecnología capaz de detectar rápidamente, responder y neutralizar las amenazas informáticas. Su objetivo principal es el de proporcionar una visión global de la seguridad de la tecnología de la información.

Un sistema SIEM permite tener control absoluto sobre la seguridad informática de la empresa.

Al tener información y administración total sobre todos los eventos que suceden segundo a segundo, resulta más fácil detectar tendencias y centrarse en patrones fuera de lo común. La tecnología SIEM nace de la combinación de las funciones de dos categorías de productos: SEM (gestión de eventos de seguridad) y SIM (gestión de información de seguridad).

SEM centraliza el almacenamiento y permite un análisis casi en tiempo real, solamente retardadas por la capacidad de proceso del sistema motor, de lo que

está sucediendo en la gestión de la seguridad, detectando patrones anormales de accesibilidad y dando mayor visibilidad a los sistemas de seguridad.

Mientas que SIM recopila los datos a largo plazo en un repositorio central para luego analizarlo, proporcionando informes automatizados al personal de seguridad informática.

Ambas funciones permiten que se pueda actuar más rápidamente sobre los ataques, ya que por un lado ofrecen más visibilidad y por otro permiten utilizar los datos para la supervisión y el análisis de la seguridad en tiempo real, avisando de los ataques que se están produciendo, o incluso los que se van a producir.

Squert

Squert es una aplicación web que se utiliza para consultar y ver los datos de eventos almacenados en una base de datos Sguil (normalmente, datos de alerta de IDS). Squert es una herramienta visual que intenta proporcionar un contexto adicional a los eventos mediante el uso de metadatos, representaciones de series de tiempo y conjuntos de resultados ponderados y agrupados lógicamente.

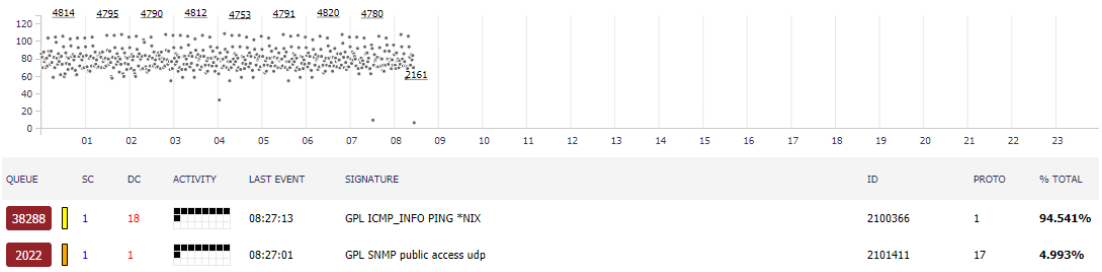


Figura 3.2: Squert.
Fuente: Programa Squert

El panel de alertas consta de varias columnas que se explican a continuación:

- QUEUE: número de eventos agrupados en la cola.
- SC: número de direcciones IPs de origen distintas para la alerta dada.
- DC: número de direcciones IPs de destino distintas para la alerta dada.

- ACTIVITY: número de eventos para una alerta dada por hora.
- LAST EVENT: hora en la que ocurrió el último evento de esa alerta.
- SIGNATURE: firma IDS del evento.
- ID: identificador de la firma del evento.
- PROTO: protocolo en relación con el evento.
- TOTAL: porcentaje en el que aparecen los eventos en relación con el total.

Kibana

Kibana es un complemento de visualización de datos de código abierto para Elasticsearch. Proporciona capacidades de visualización sobre el contenido indexado en un clúster de Elasticsearch. Los usuarios pueden crear diagramas de barras, líneas y dispersión, o gráficos circulares y mapas sobre grandes volúmenes de datos.

Kibana también es conocido por el stack ELK:

- Elasticsearch
- Logstash
- Kibana

Snorby

Snorby es un interfaz para la monitorización de alertas basado en Ruby. La ventaja clave es la flexibilidad, es decir, se puede configurar la interfaz para que acepte eventos provenientes de diferentes aplicaciones, siendo solo necesario añadir determinados códigos. Snorby se utiliza para supervisar la seguridad de red gracias a la incorporación de eventos de IDS/IPS como Snort o Suricata.

Mediante la captura de paquetes (CapME21) proporcionados por Snorby, se puede hacer un filtrado con las características que se deseen, por ejemplo, seleccionar dirección del equipo y destino, el protocolo de transmisión, la fecha y hora en la

que queremos buscar en la base de datos para obtener todos los eventos relacionados. De esta forma, se simplifica la búsqueda y permite centrar el análisis en los eventos necesarios.

Sguil

La herramienta Sguil está construida por y para los analistas de seguridad de red. El principal componente de Sguil es una interfaz gráfica de usuario que proporciona acceso a los eventos en tiempo real, datos de sesión, y capturas de paquetes. Sguil facilita el seguimiento y análisis de eventos en la red. El cliente Sguil se puede ejecutar en multitud de sistemas operativos, incluyendo Linux, BSD, Solaris, MacOS y Windows.

Sguil proporciona visibilidad sobre los datos de evento recogidos y el contexto para validar la detección. Proporciona una única interfaz gráfica de usuario, en cual, se ven las alertas de Snort o Suricata, alertas OSSEC, eventos HTTP Bro, y las alertas del sistema de detección pasivo de activos en tiempo real (PRADS23).

Más importante aún, Sguil permite ver todo el tráfico asociado a una alerta, consultar todos los paquetes capturados, y también el tráfico que no tiene porqué pertenecer a esa alerta, pero podría estar asociado con la actividad maliciosa o no deseada.

Sguil se diferencia de otras interfaces de alerta en que permite la colaboración entre los analistas permitiendo comentar las alertas.

ELSA

La herramienta para búsqueda y almacenado de logs de empresa (ELSA - Enterprise Log Search and Archive) es un analizador de eventos que opera en tres niveles: receptor de log, base de datos o almacenador e interfaz web para syslog entrantes. Aprovecha un analizador basado en una base de patrones para la normalización de eventos y usa el motor de búsqueda Sphinx26 para la indexación de texto completo para realizar la búsqueda de eventos.

Si bien una mayoría de sistemas de opensource utilizan REGEX para buscar y parsear los datos de logs, cuando se escala a cantidades de miles e incluso millones de dispositivos, este sistema se vuelve inoperativo. De esta necesidad nace ESLA, utilizando un motor de búsqueda basado en el buscador de Google y la combinación de MySQL y Syslog-NG, que permite buscar por patrones en ficheros de log XML con un algoritmo ultrarápido (Aho-Corasick).

ELSA permite realizar una exploración que puede estar escalada en los diferentes nodos que tenga un sistema distribuido. El proceso de normalización asigna a cada usuario entrante un identificador según la clase de usuario.

Los usuarios pueden conceder permisos (listas blancas) granulares para un host o programa, es decir, un usuario puede limitarse a uno o varios hosts, pero es capaz de consultar cualquier programa o clase en estos equipos.

ELSA se divide en tres componentes principales: los nodos finales, el DAEMON27 (proceso demonio) que se ejecuta en el servidor web, y el propio sitio Web. Los nodos no tienen conocimiento de la interfaz web y responden a cualquier petición a su puerto de escucha.

ELSA permite realizar búsquedas de logs igual que si fuera un navegador web, tan solo insertar un espacio de tiempo para realizar un filtrado de todos los logs e indicar un filtro ya predefinido para realizar la búsqueda.

Splunk

Splunk es un sistema que permite la correlación de eventos y la incorporación de los datos de campo e informes para sguil, Bro IDS y OSSEC, e incluye varios cuadros de mando e interfaz de búsqueda para correlar eventos.

Proporciona una plataforma altamente escalable para los datos generados por todos los dispositivos de los sistemas de control, sensores, sistemas SCADA, redes, aplicaciones y usuarios finales conectados a estas redes industriales.

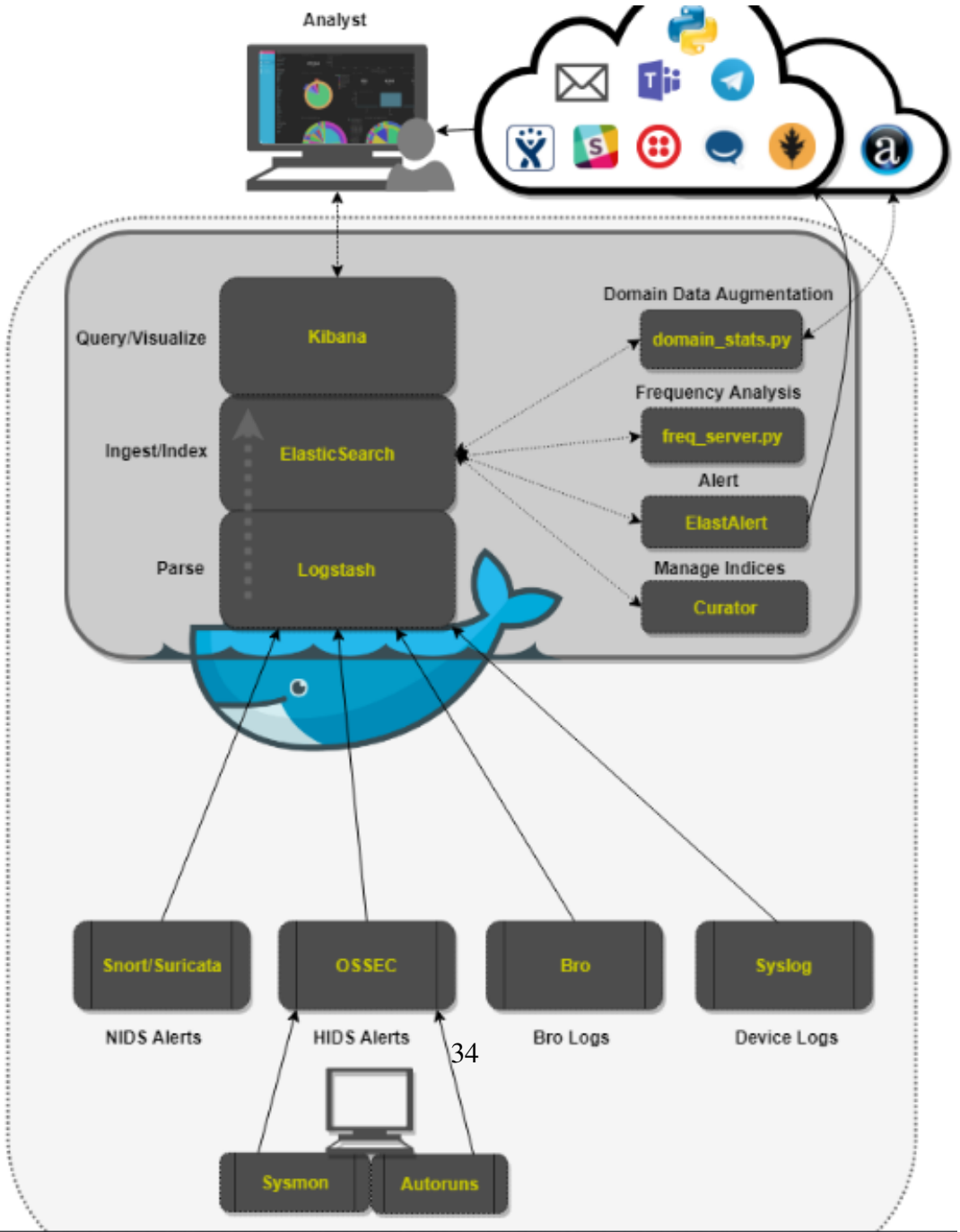
Splunk eleva la eficiencia operativa por medio de:

- La integración y agregación de datos a través de tecnología operativa.

- Busca, explora y correlaciona a través de múltiples fuentes para diagnosticar rápidamente los problemas operativos más costosos.
- Aprovecha la analítica avanzada, proporcionando la capacidad de detectar patrones, tendencias y anomalías.
- Entrega rápidamente valores a través de modelos de implementación flexibles.

Capítulo 4

Metodología



Para nuestro objetivo, utilizaremos la siguiente metodología:

- Simularemos un entorno empresarial real. Las pruebas se han realizado en una empresa y se hemos monitorizado todo el tráfico.
- Instalaremos las herramientas de seguridad con la arquitectura que hemos visto en la figura anterior.
- Parametrizaremos las herramientas para detectar los ataques estándares que recogen las firmas.
- Simularemos usuarios internos que accederán a malware a través de enlaces en redes sociales y comprobaremos como no son detectados.
- Desarrollaremos las nuevas firmas que detecten los accesos anteriores.
- Reconfiguraremos las herramientas para los nuevos patrones.

4.1. Security Onion

SecurityOnion es una distribución basada en GNU/Linux, concretamente en Ubuntu y contiene un conjunto bastante completo de herramientas para la detección/prevención de amenazas.

Básicamente, incorpora de forma preinstalada, diversas herramientas ampliamente conocidas, un IDS como Snort y/o Suricata (nos da a elegir qué herramienta deseamos usar), GUIs para monitorizar los eventos de forma automática (Squert, Sguil), herramientas específicas para análisis de PCAPs (Wireshark, NetworkMiner) y herramientas populares de análisis forense de red como (Bro o Xplico).

Figura 4.2: Logo Security Onion.
Fuente:Security Onion

Capítulo 5

Implementación, desarrollo y resultados

5.1. Security Onion

5.1.1. Instalación de Security Onion

Para realizar la instalación de la versión 16.04.5.2 de security onion hay que seguir los siguientes pasos que se han desarrollado en laboratorio virtual de la empresa Ariadnex Tecnología Flexible S.L., entre Septiembre y Diciembre de 2018. Se ha detallado prolijamente la instalación habida cuenta de los múltiples errores habituales cuando se integran e instalan este tipo de herramientas: [37]

- Descargar VMware vSphere Client versión 5.5.0.
- Descargar la versión 16.04.5.2 de Security Onion.
- Instalar VMware VSphere Client en el equipo.
- Una vez instalado ir a inventory.
- Doble click en datastore1.
- Upload files y buscar el security onion que se ha descargado en el portatil.

- Crear máquina virtual y configurarla.
- Apagar máquina virtual.
- Poner el CD correcto en edit settings.
- Seguir en edit settings en options - boot marcar Force BIOS Setup.
- Encender máquina.
- Abrir console para ver la interfaz gráfica.
- Mover hasta boot y comprobar que el CD-ROM Drive está en la primera opción, si es asi, dar a exit.
- Configurar puertos eth, en eth(ensXX) dar a wired setting -¿IPv4 -¿En addresses cambiar a manual y escribir la ip, mask y gw en DNS poner 8.8.8.8 y con eso ya está configurado el puerto para tener conexión a internet.
- Dar en el icono del escritorio de la interfaz donde pone install security onion, indicar el idioma en el que se quiera trabajar, señalar que se descargue automáticamente las actualizaciones, en el tipo de instalación poner ”borrar disco e instalar Security Onionz por último dar a instalar ahora, poner datos del equipo, el usuario y la contraseña.
- Reiniciar.
- Quitar el disco para que al reiniciar no vuelva a leer el disco y no se vuelva a hacer la instalación, para ello dar a edit setting -¿CD/DVD -¿Client Device.
- Setup - Configurar network, seleccionar la interfaz de gestión (la que da internet) poner ip, dns, mask y gw.
- Reiniciar.
- Setup - skip configure network y poner usuario y password.

Una vez instalado Security Onion y comprobado que funciona se recomienda hacer un snapshot, que es una copia de seguridad imagen del volumen completo en un instante determinado. De esta manera, en caso de corrupción en las sucesivas integraciones se facilitaría una marcha atrás ágil y restaurar el laboratorio en un tiempo razonablemente corto. Hay que tener en cuenta que al volver al snapshot la fecha y hora será la de cuando se hizo el snapshot por lo tanto habrá que cambiarla, la forma más sencilla es la siguiente:

sudo date –set .ºct 25 2018 11:53”

Para permitir el acceso por web y por terminal fuera de la máquina virtual, se tendrá que abrir un terminal en la máquina virtual donde está instalado security onion y teclear sudo so-allow, esto dará varias opciones y hay que elegir la primera, pedirá la ip de la red desde donde se va a conectar por web y por terminal, en el caso de la oficina sería 192.168.1.0/24.

5.1.2. Acceso a security onion

Para acceder a security onion fuera de la máquina virtual hay varias opciones:

- Por web: el acceso por web consiste en poner https:// y la ip que se ha configurado en este caso será https://172.30.2.34 al cargar pedirá usuario y contraseña que se habrá asignado anteriormente y una vez introducida aparecerá una página con opciones para poder entrar en Kibana, squert, entre otros.

- Por terminal Linux: en el terminal habrá que poner ssh ariadnex@172.30.2.34, pedirá una contraseña que en este caso es ”Wedwarect1”.

- Por putty (versión 0.70) en Windows: para poder entrar con putty hay que poner la ip 172.30.2.34, puerto 22, y señalar SSH, al iniciar pedirá usuario: .ªriadnexz contraseña: ”Wedwarect1”.

5.1.3. Creación de usuarios

Al instalar security onion solamente se puede crear un usuario, pero nos indica que si queremos crear más usuarios lo podemos hacer después de la instalación con el comando:

so-user-add

Al introducir este comando en modo root, nos pedirá el nombre y la contraseña del nuevo usuario como vemos a continuación:

```
ariadnex@ARDIDS02:~$ sudo su
root@ARDIDS02:/home/ariadnex# so-user-add

User Name
Enter the name of the new user that will be granted privilege to connect to Sgui
l/Squert/Kibana: Guadalupe

User Pass
Enter the password for the new user that will be granted privilege to connect to
 this server:
Verify:

Add User to Server
The following information has been collected:

  user:        Guadalupe

Do you want to create? (Y/N) [Y]: Y
Adding user: Guadalupe
Guadalupe successfully added.
```

Figura 5.1: Usuario nuevo.
Fuente:Elaboración propia

5.1.4. Implementación de Trisul en Security Onion

El trabajo principal de Trisul es monitorear las estadísticas de tráfico y correlacionarlas con los flujos de la red y respaldar todo mediante paquetes sin procesar. Esto se presenta en una interfaz web pulida para brindarle una gran visibilidad de su red con detalles detallados y laterales disponibles en cada etapa [38].

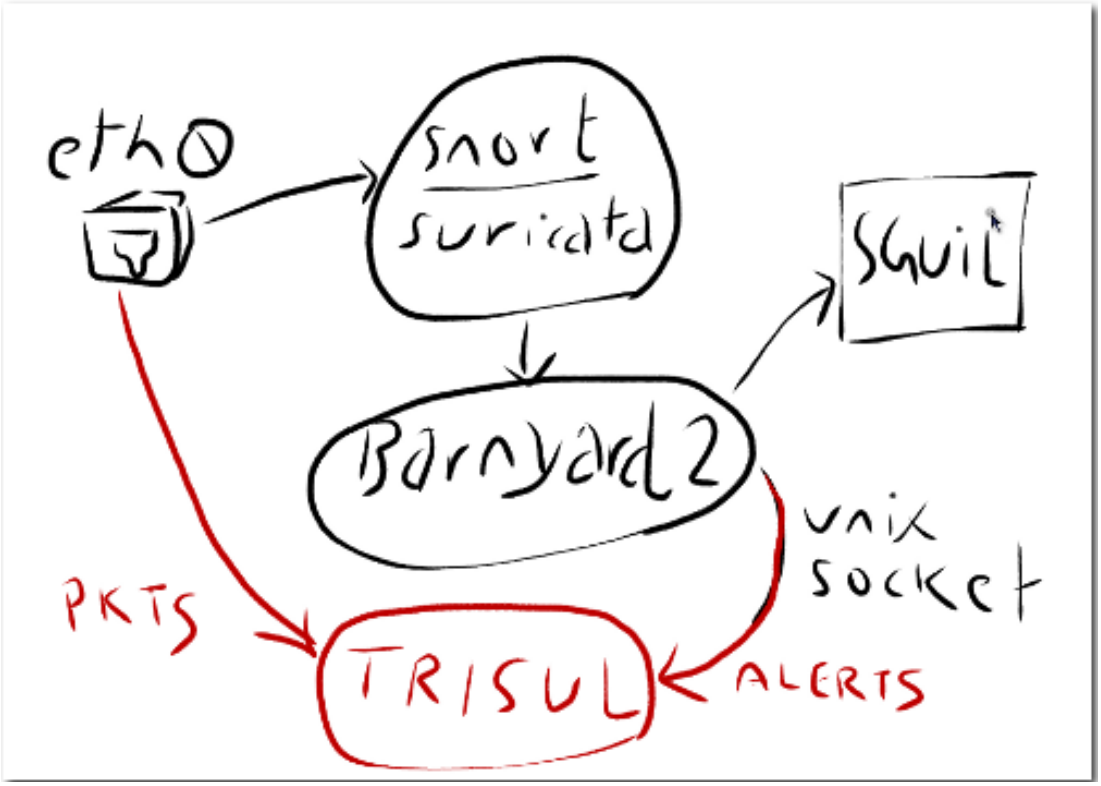


Figura 5.2: Trisul.
Fuente:Elaboración propia

Para instalar Trisul en Security Onion se utilizará el terminal de Ubuntu o el putty en el caso de Windows, en ambos se seguirán los siguientes pasos [39]:

- sudo add-apt-repository http://trisul.org/repos/apt/debian
- sudo apt-key adv –keyserver keyserver.ubuntu.com –recv-keys A6CC1B18 en este paso hay veces que dice que la key no es correcta o que no puede acceder al servidor, una de las posibles soluciones sería poner lo siguiente: gpg –keyserver hkp://keyserver.ubuntu.com:80 –recv-keys A6CC1B18 ya que por defecto se utiliza el puerto 11371 pero en la mayoría de las ocasiones el firewall bloquea el puerto, este comando fuerza a que se utilice el puerto 80 que el firewall no lo tiene bloqueado, otra opción es desbloquear el firewall.
- sudo apt-get update
- sudo apt-get install trisul-full

- sudo apt-get install trisul-badfellas trisul-urlfilter trisul-geo, este comando instala plugins, que normalmente se instalan con el comando anterior, pero si no funciona algún plugin se puede instalar aparte.

Con estos pasos ya está instalado trisul, los siguientes pasos son para implementar trisul en security onion [40]:

- sudo ldconfig, actualiza el caché de la biblioteca.
- sudo ufw allow 3000 sudo ufw allow 3003,abre los puertos 3000 y 3003 que son utilizados por trisul para la interface web.
- sudo trisulctl-hub changeuser domain domain0 sguil.sguil, cambia los permisos de hub a sguil.
- quit
- sudo trisulctl-probe changeuser domain domain0 sguil.sguil, cambia los permisos de probe a sguil.
- quit
- sudo service webtrisuld restart, reinicia el servidor web bajo la nueva propiedad de sguil.
- Abrir el archivo de configuración: sudo nano /usr/local/etc/trisul-probe/domain0/probe0/context0/trisulProbeConfig.xml
- Cambiar ¡IDSAlerts¿¡UnixSocket¿parameter to /tmp/barnyard2-alert o /nsm/sensor-data/ardids02-ens32/snort-1/snort.unified2.1538643212 o /var/log/nsm/ardids02-ens32/snortu-1.log
- Abrir el archivo: sudo nano /etc/nsm/templates/barnyard2/barnyard2.conf y poner alert-unixsock option
- Abrir sudo nano /etc/nsm/templates/barnyard2/barnyard2-1.conf y cambiar output alert-fast: stdout por alert-unixsock option

- barnyard2 -c /etc/nsm/ardids02-ens32/barnyard2-1.conf -l /tmp -o /nsm/sensor-data/ardids02-ens32/snort-1/snort.unified2.*

- sudo nsm-sensor-ps-restart –only-barnyard2

Con todos estos pasos ya está completa la instalación de trisul para acceder a la interface web, se deberá poner en el navegador la ip seguido del puerto 3000 ”http://ip:3000”.

Configurar para que el sensor no sea eth0, sino ens224 o ens33, según cuál de los dos sea el sensor, para ello abrir trisul como admin, seleccionar profile 0, create adapter y poner en el nombre ens224 o ens33, no hay que modificar ningún parámetro más, deshabilitar eth0 y habilitar ens.

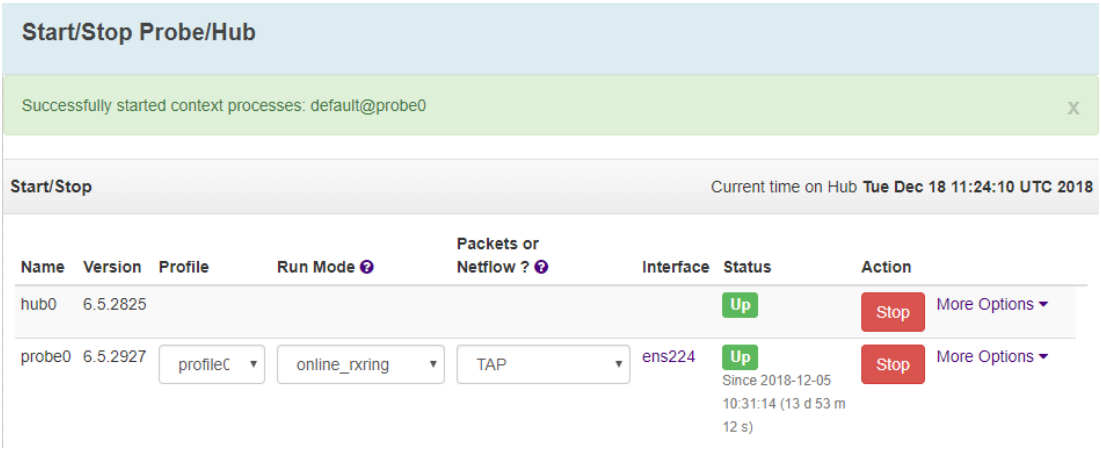


Figura 5.3: Trisul interface.
Fuente:Elaboración propia

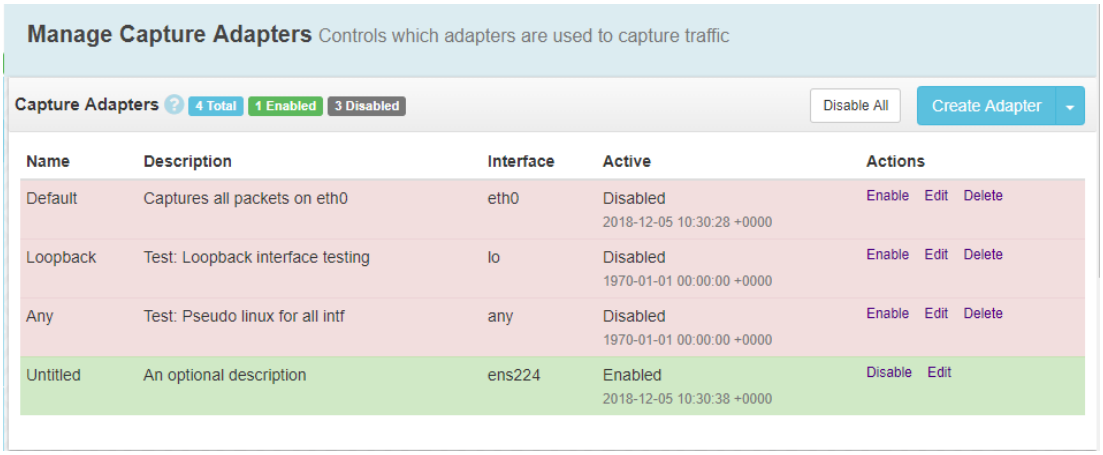


Figura 5.4: Trisul activar interface.
Fuente:Elaboración propia

Si no funcionara alguna de las alertas puede ser que esté deshabilitada. Para comprobarlo entramos en Trisul como administradores, vamos a profile, all alert groups y ahí están todas las alertas que se pueden habilitar y deshabilitar.

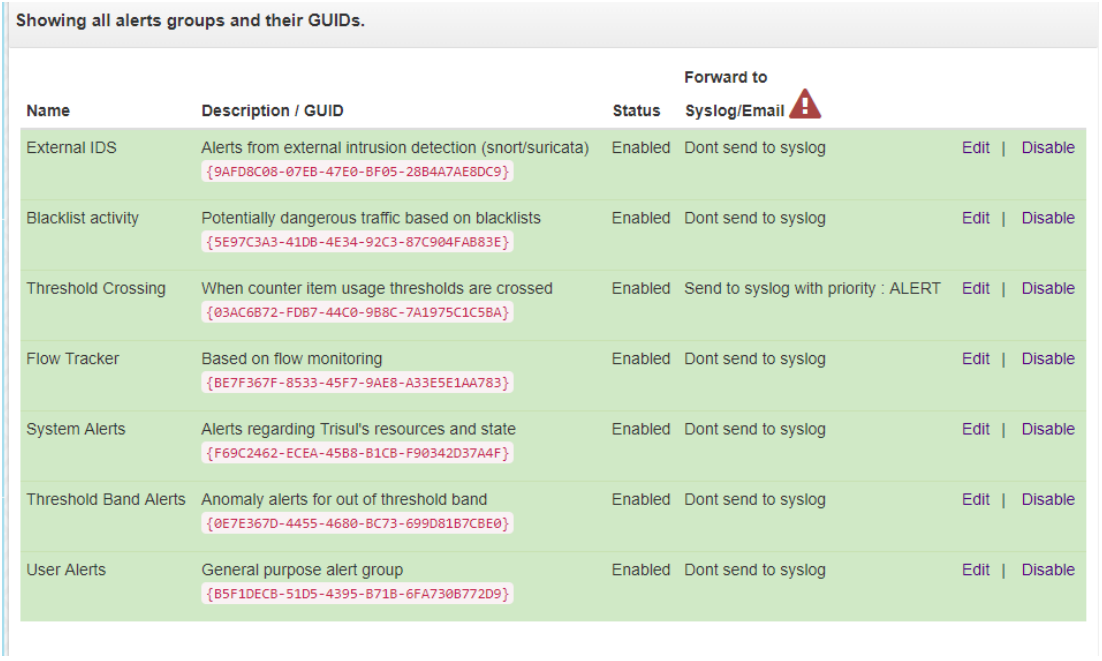


Figura 5.5: Trisul alertas.
Fuente:Elaboración propia

Una de las características de trisul es que podemos ver los siguientes gráficos:

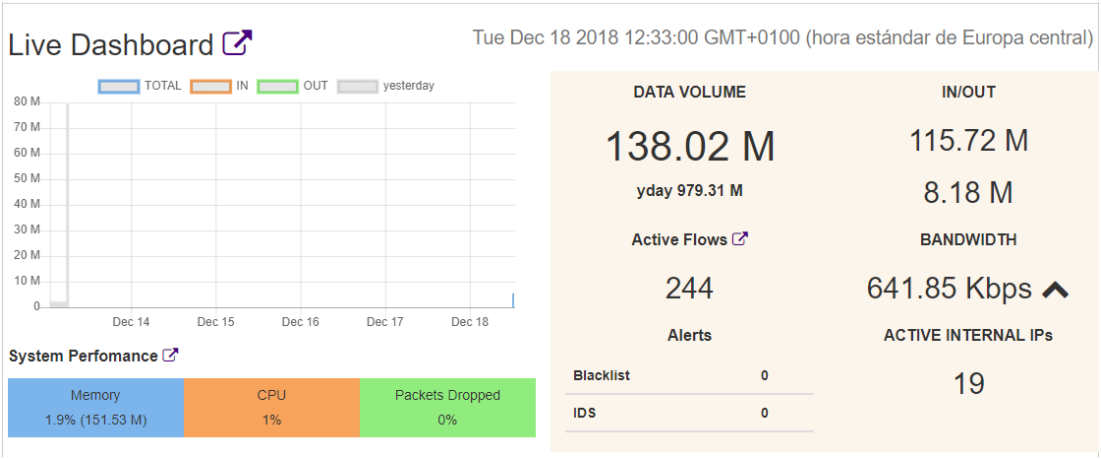
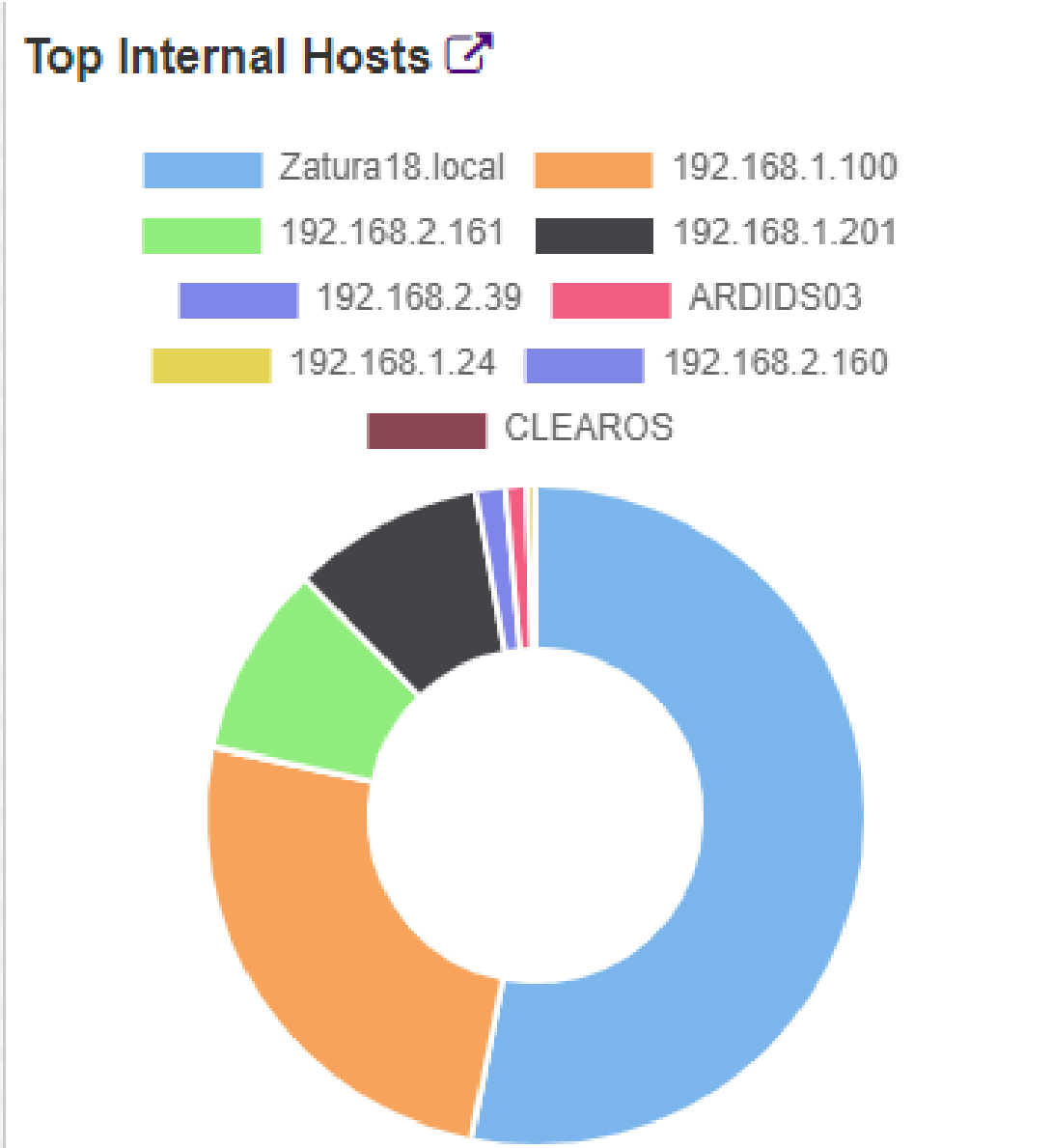


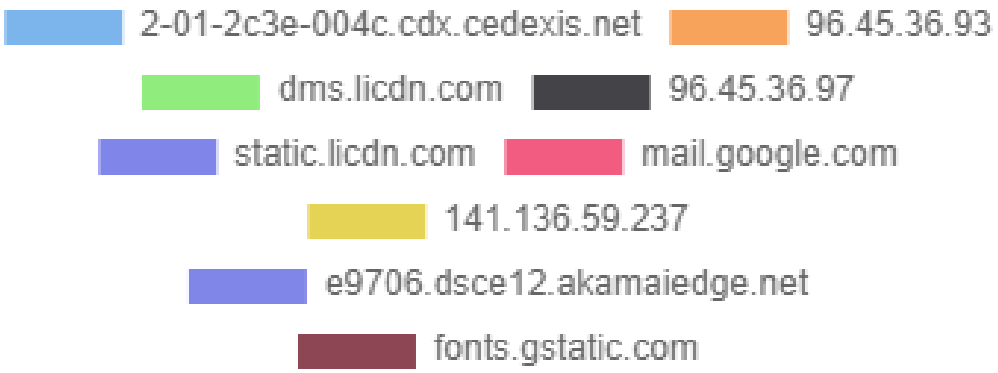
Figura 5.6: Trisul dashboard.
Fuente:Elaboración propia

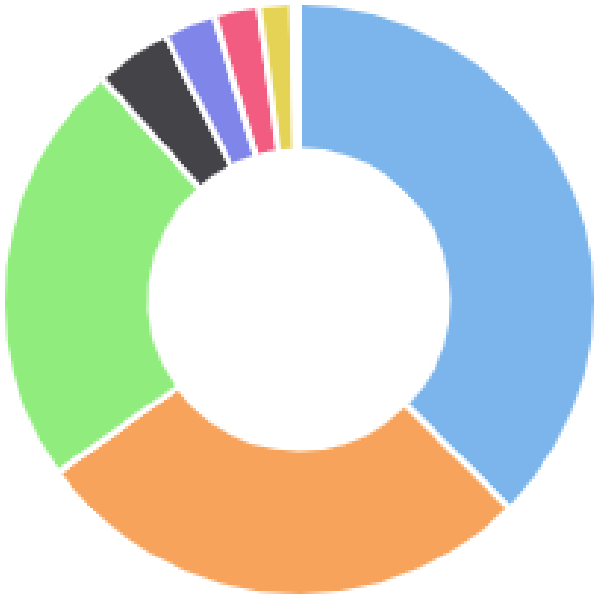


Zatura18.local	2.06 M
192.168.1.100	999.84 K
192.168.2.161	372.54 K
192.168.1.201	372.30 K
192.168.2.39	57.95 K
ARDIDS03	42.23 K

Figura 5.7: Trisul internal hosts.
Fuente:Elaboración propia

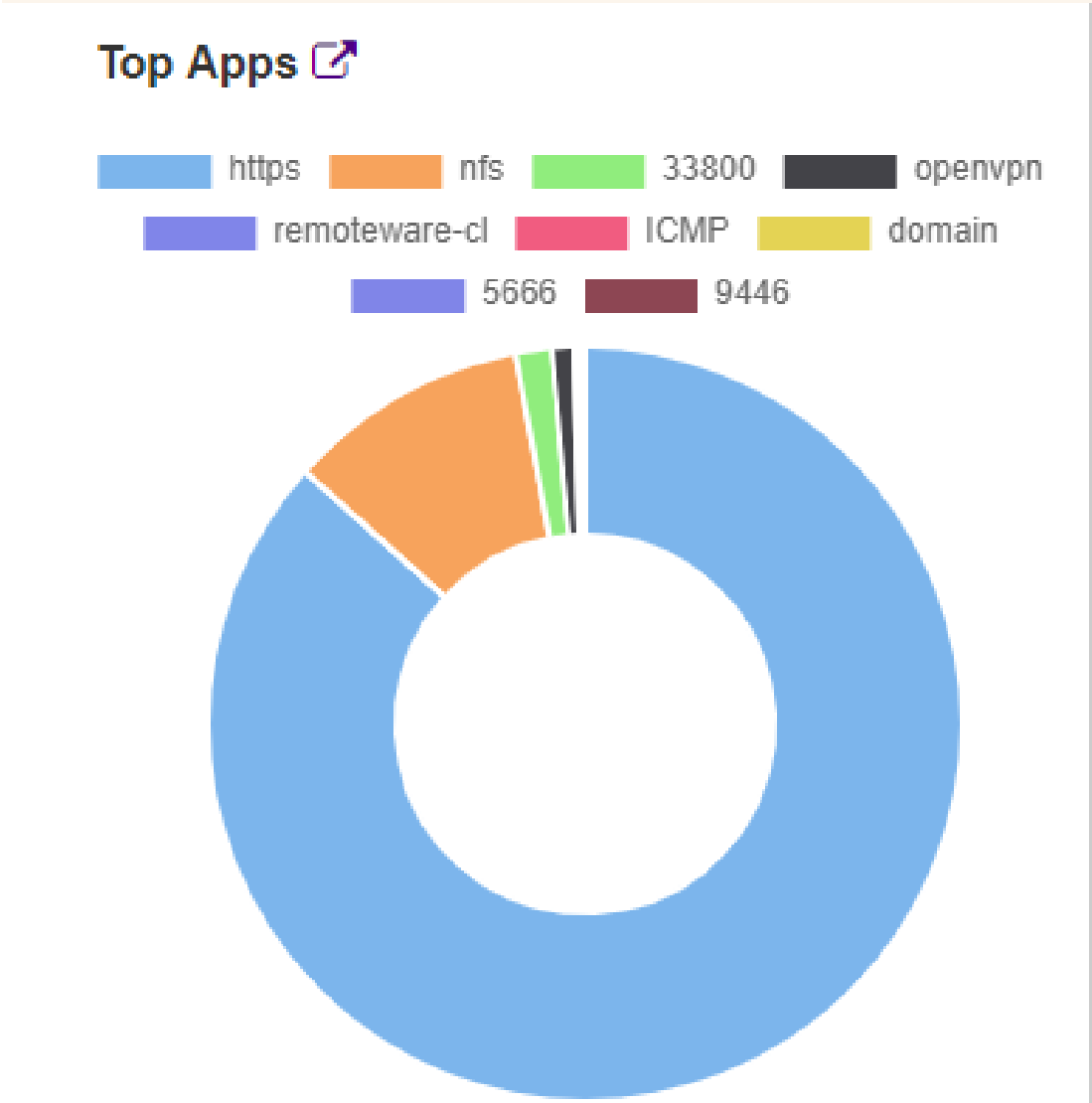
Top External Hosts





2-01-2c3e-004c.cdx.cedexis.net	1.14 M
96.45.36.93	863.09 K
dms.licdn.com	727.50 K
96.45.36.97	129.03 K
static.licdn.com	87.94 K
mail.google.com	76.34 K

Figura 5.8: Trisul external hosts.
Fuente:Elaboración propia



https	3.04 M
nfs	371.91 K
33800	56.47 K
openvpn	33.16 K
remoteware-cl	8.77 K
ICMP	4.04 K

Figura 5.9: Trisul apps.
Fuente:Elaboración propia

5.1.5. Implementación de Ntop en Security Onion

Ntop es una herramienta que permite la monitorización en tiempo real de una red.

Figura 5.10: Ntop.
Fuente:Elaboración propia

Es útil para controlar los usuarios y aplicaciones que están consumiendo recursos de red en un instante concreto y para ayudarnos a detectar malas configuraciones de algún equipo, (facilitando la tarea ya que, justo al nombre del equipo, aparece un banderín amarillo o rojo, dependiendo si es un error leve o grave), o a nivel de servicio.

Posee un microservidor web desde el que cualquier usuario con acceso puede ver las estadísticas del monitorizaje.

El software está desarrollado para platarfomas Unix y Windows.

En Modo Web, actúa como un servidor de Web, volcando en HTML el estado de la red. Viene con un recolector/emisor NetFlow/sFlow, una interfaz de cliente basada en HTTP para crear aplicaciones de monitoreo centradas en top, y RRD para almacenar persistentemente estadísticas de tráfico.

Los protocolos que es capaz de monitorizar son: TCP/UDP/ICMP, (R)ARP, IPX, DLC, Decnet, AppleTalk, Netbios, y ya dentro de TCP/UDP es capaz de agruparlos por FTP, HTTP, DNS, Telnet, SMTP/POP/IMAP, SNMP, NFS, X11.

Para la implentación de ntop en security onion debemos seguir los siguientes pasos

(24):

- sudo soup, este comando lo que hace es actualizar security onion, ya que ntop trabaja con la última versión de security onion, despues de esto nos pedirá reiniciar el equipo.
- rm -f install-ntopng-on-so-16
- wget –no-check-certificate https://github.com/branchnetconsulting/so-ntopng-installer/raw/master/instal
- chmod 700 install-ntopng-on-so-16
- sudo ./install-ntopng-on-so-16

Con estos 5 pasos ya estará instalado el ntop en nuestro security onion, para poder acceder a ntop debemos poner https://(ip):3000 , el usuario y contraseña por defecto es admin/admin, la primera vez que ingresamos pedirá cambiar la contraseña.

En la interfaz web de ntop podemos ver el tráfico que pasa por la sonda y lo clasifica en:

- Talkers: se puede ver el diagrama de Sankey, el diagrama de Sankey muestra los hosts activos actualmente en la interfaz supervisada o en la vista de interfaz. Los pares de hosts se unen mediante barras de colores que representan los flujos. El host del cliente siempre se coloca en el borde izquierdo de la barra. Del mismo modo, el servidor se coloca a la derecha. El ancho de la barra es proporcional a la cantidad de tráfico intercambiado. Cuanto más ancha sea la barra, mayor será el tráfico intercambiado entre el par de hosts correspondiente. Por defecto, el diagrama se actualiza cada 5 segundos. La frecuencia de actualización se puede configurar o deshabilitar desde el menú desplegable que se muestra justo debajo del diagrama. La información de host y flujo que se muestra en Sankey es interactiva. De hecho, tanto los nombres de host (direcciones IP) como los flujos son accesibles.

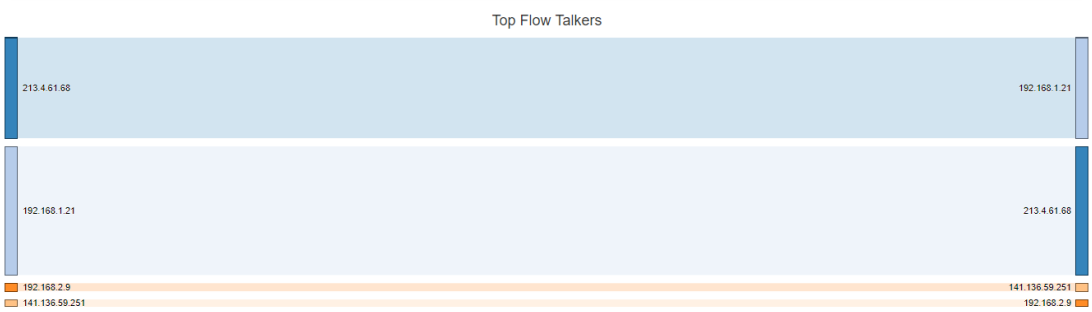


Figura 5.11: Talkers.
Fuente:Elaboración propia

- - Host: Hosts View proporciona una representación de gráfico circular del tráfico capturado. La agregación se realiza por host. Al igual que en el Diagrama de Sankey descrito anteriormente, se puede hacer doble clic en cualquier nombre de host (o dirección IP no resuelta) que se muestra para visitar la página correspondiente de 'Detalles del host'.

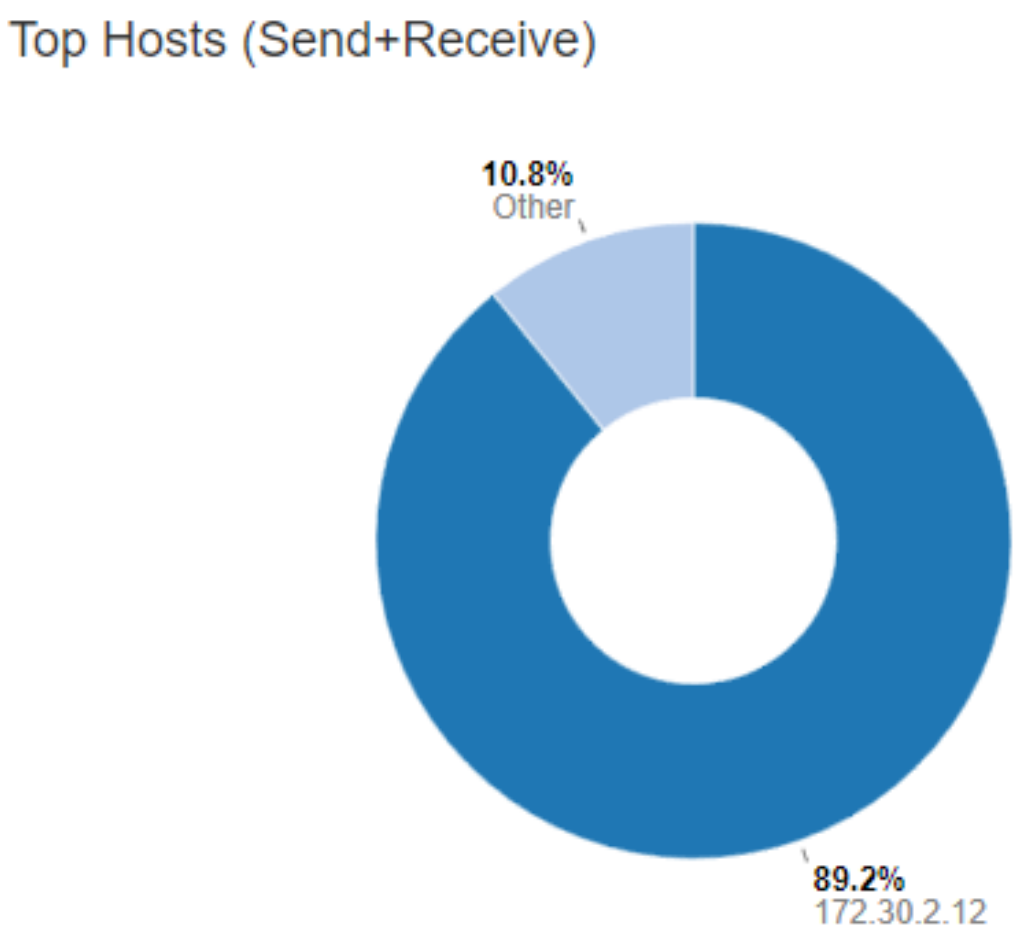


Figura 5.12: Host.
Fuente:Elaboración propia

- Ports: La vista de puertos proporciona dos gráficos circulares separados con los puertos más utilizados, tanto para clientes como para servidores. Cada gráfico circular proporciona estadísticas para los puertos del cliente y los puertos del servidor.

  Se puede hacer doble clic en cualquier número de puerto que se muestra para visitar la página 'Flujos activos'. Esta página enumera todos los flujos activos actualmente, de modo que el puerto del cliente o del servidor coincida con el que se hizo clic.

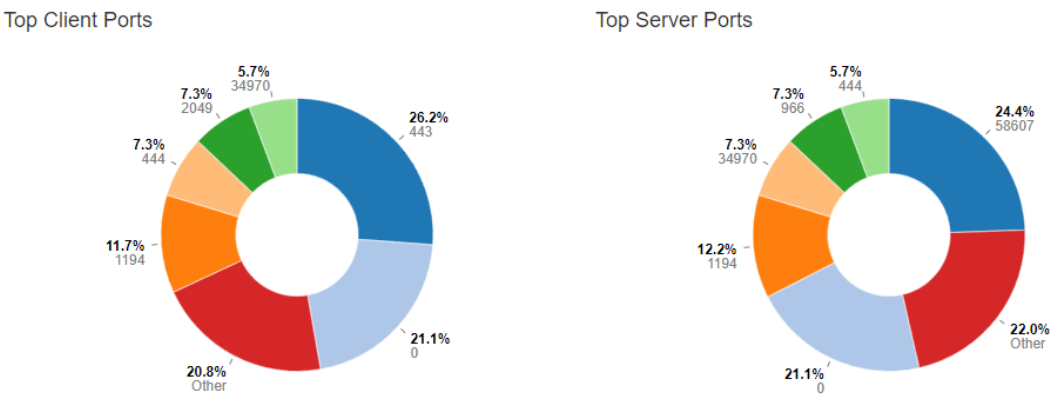


Figura 5.13: Ports.
Fuente:Elaboración propia

- Aplicattions: La Vista de aplicación proporciona otro gráfico circular que representa una vista del uso del ancho de banda dividido por protocolo de aplicación. La identificación del protocolo se realiza a través del motor nDPI ntopn. Los protocolos que no se pueden identificar están marcados como Desconocidos.

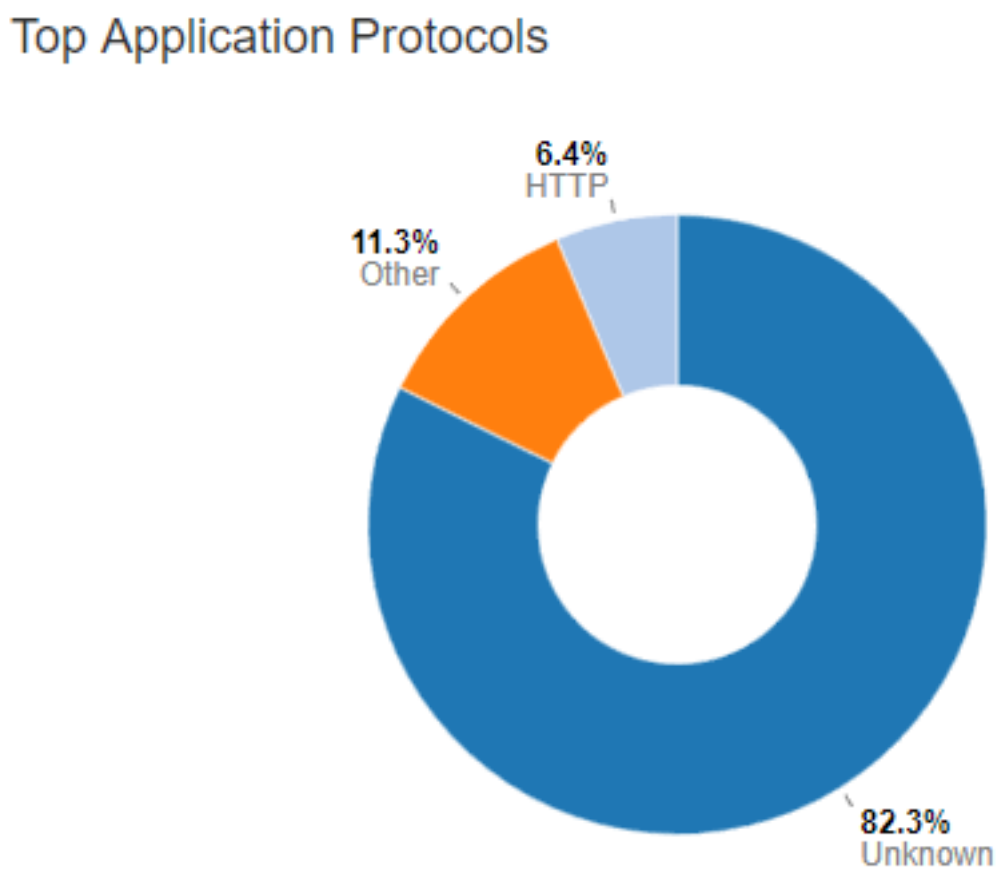


Figura 5.14: Aplicattions.
Fuente:Elaboración propia

- ASNs: La vista de ASN proporciona una representación de gráfico circular del tráfico agrupado por Sistema Autónomo (AS). Un AS es una red única o un grupo de redes, controlado por un administrador de red en nombre de una entidad administrativa única (como una universidad, una empresa comercial o una división de negocios). Un AS también se conoce como un dominio de enrutamiento. Se asigna un número único global llamado Número de Sistema Autónomo (ASN) a cada AS.

- Senders: Este gráfico muestra el porcentaje de tráfico que envían los puntos finales en redes locales o remotas.

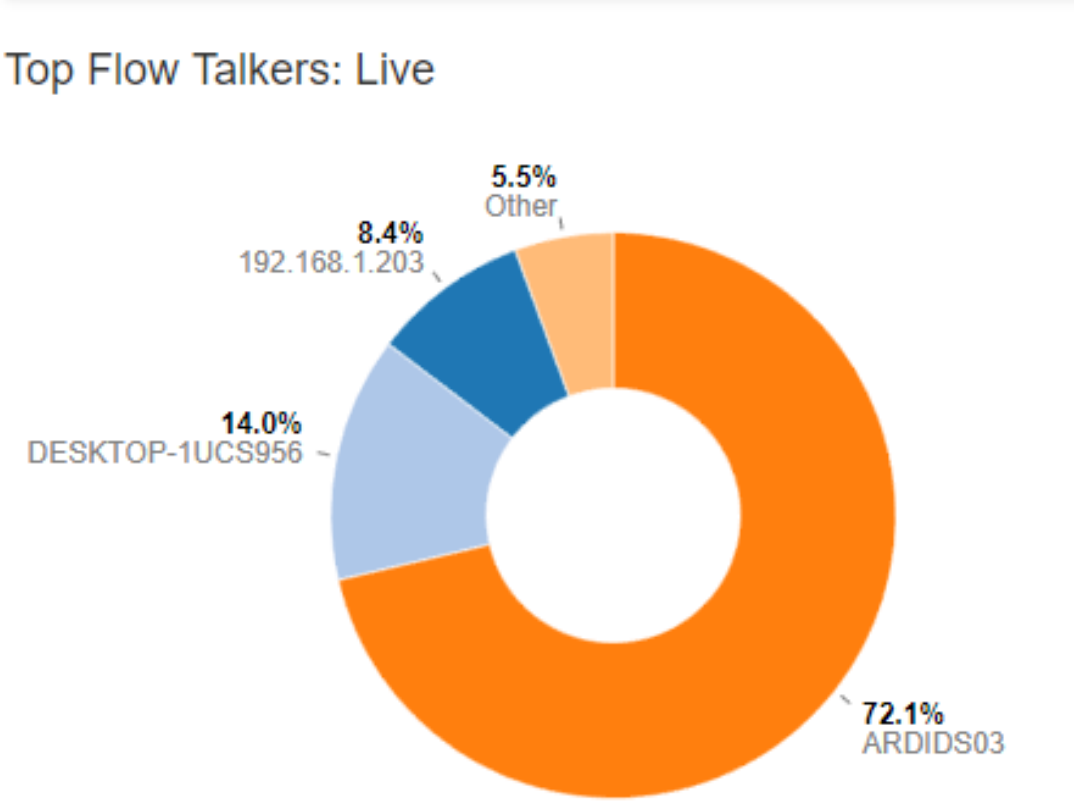


Figura 5.15: Senders.
Fuente:Elaboración propia

5.1.6. Creación de firmas locales

Para que funcionen las reglas locales tenemos dos opciones: o se incorpora en la regla local ”metadata:policy security-ips.º se descomenta la linea ips-policy=security del fichero de configuración ”pulledpork.conf”, que se encuentra normalmente en la siguiente ruta: /etc/nsm/pulledpork/pulledpork.conf, esta última opción deshabilita todas las reglas descargadas por lo que kibana y squert solo podrán detectar alertas locales.

Para crear una firma hay que localizar el fichero local.rules en nuestro caso se encuentra en usr/share/nsmnow/templates/snort/rules, dentro de local rules se crean las firmas locales, para ello hay que tener en cuenta los siguientes parámetros (25):

- SID: se utiliza para identificar de forma única las reglas de Snort, el archivo

sid-msg.map contiene todos los SID que están definidos, si creamos una firma hay que poner un SID que no esté definido para otra firma.

- REV: identificación de forma única las revisiones de las reglas de snort, esto permite que las firmas estén actualizadas.
- CLASSTYPE: se utiliza para categorizar una regla, solo se pueden utilizar clasificaciones que se hayan definido en snort.conf.

Al crear una firma se debe guardar el archivo y poner el siguiente comando sudo rule-update para que actualice el fichero ”sid-msg.map”donde se encuentran las firmas descargadas, además se reinicia snort y barnyard.

Si el fichero sid-msg.map no se actualiza hay que comprobar que en el fichero de configuración pulledpork.conf la ruta del fichero local.rules es la correcta.

5.1.7. Crear paquete para probar una firma

Para probar si una alarma funciona antes de que se produzca un evento del tipo en cuestión, se crean paquetes con scapy, que es una librería de phyton. En ella tenemos que especificar la dirección IP origen y destino, el puerto origen y destino y la carga o payload, que es lo que va a contener el paquete. Por ejemplo, si queremos que nos envíe un aviso de que han accedido a la aplicación Facebook, en el payload deberá escribirse ”facebook”.

Un ejemplo de la formación de un paquete sería la siguiente:

- ip= IP()
- ip.dst = ”8.8.8.8”
- ip.src = ”192.168.100.3”
- tcp = TCP()
- tcp.dport = 80

- tcp.sport = 1234
- payload = ”Facebook”
- send (ip/tcp/payload)

Es importante que se ejecute scapy como admin, es decir en la línea de comandos de Linux deberemos teclear ”sudo scapy.º en windows ejecutar cmd como administrador. Si no lo hiciéramos de esta manera, nos avisaría de un error advirtiendo de que no se habría podido enviar los paquetes.

5.1.8. Crear firma DNS

Domain Name System (DNS) es un sistema globalmente distribuido, escalable y jerárquico. Ofrece una base de datos dinámica asociando direcciones IP de computadoras, servicios o cualquier recurso conectado a internet o red privada con información de diverso tipo. Soporta tanto IPv4 como IPv6, y la información se almacena en forma de registros Resource Records (RR) de distintos tipos los cuales pueden almacenar direcciones IP u otro tipo de información. Esta información se agrupa en zonas, que corresponden a un espacio de nombres o dominio y que son mantenidas por el servidor DNS autoritativo de la misma. Fundamentalmente, DNS se encarga de traducir direcciones IP de recursos de red a nombres fácilmente legibles y memorizables por las personas, y viceversa. A esta acción se la conoce como “resolución DNS”. De esta forma, se establece un mecanismo amigable para la localización e identificación de recursos. Comúnmente se usa la analogía de una guía de teléfonos donde se puede localizar a partir de un nombre su número asociado, o a la inversa. En este símil, los números representarían direcciones IP y los nombres, registros del espacio de dominios. (26)

Resulta obvio pensar que si queremos implementar un entorno seguro deberemos ser advertidos de cuándo los usuarios acceden a determinados web sites, o, lo que es equivalente, qué nombres de dominios visitan. Por ello, al crear una firma DNS, el sistema nos debe indicar cuándo el usuario ha accedido a una determinada página que

hemos señalado previamente como potencialmente maliciosa. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

Ejemplo de firma DNS:

```
alert udp any any -> any 53
(msg:"Facebook DNS";
byte_test:1,!&,0xF8,2;
content:"|08|facebook|03|com|00|";
fast_pattern: only;
sid:1003;
rev:1;)
```

Figura 5.16: Ejemplo de firma DNS.
Fuente:Elaboración propia

En este ejemplo se comprueba que existe una petición DNS al dominio de facebook.com, lo podemos dividir en varias partes:

- Se trata de una alerta UDP en la que se utiliza el puerto 53 de nuestra red interna para recibir la petición.
- El mensaje puede ser el que el usuario quiera en este caso es: TFM GUADALUPEBV Facebook DNS.
- byte-test: nos permite coger una cantidad de bytes del paquete a partir de una posición y comprobar si coincide con otro valor. Lo primero que tenemos que mirar es el último campo, en nuestro caso “2”. Esto indica la posición dentro del paquete donde nos situaremos, teniendo en cuenta que siempre el primer byte vale 0. Por tanto, el valor “2” nos indica que se situará en el tercer byte del paquete. El tercer byte ya implica haber descontado la cabecera IP y la cabecera

UDP: ya se está trabajando sobre la cabecera DNS. Tras esto, se debe mirar el primer campo de byte-test, en este caso “1”. Esto indica cuántos bytes vamos a coger a partir de la posición “2”. Por tanto, utilizaremos un solo byte. En pocas palabras, vamos a trabajar con el tercer byte del protocolo DNS. Tras esto, se cogerá el segundo campo del byte-test cuyo valor es “! & “. Esto implica que con el valor del paquete se va a realizar una operación binaria “AND negada” sobre el valor indicado en el tercer campo de byte-test: “0xF8“. Si la condición es true hará “match” en la regla. Siempre que se utilizan operaciones binarias de tipo AND u OR lo que se busca es que el valor de un campo sea uno en concreto o que al menos algunos de los campos tengan un valor. En nuestro ejemplo lo que se está buscando es “F8“, en binario “1111 1000“. Por tanto, vamos a trabajar con los bits del 7 al 3 asignados al campo QR y OpCode pertenecientes al tercer byte del protocolo DNS. Si la comprobación fuera únicamente una operación AND (“ & ”) la regla lo que estaría intentado comprobar es que el campo QR valga 1 y que el campo OpCode valga “1111”. Pero al ser un AND negado lo que hace es lo contrario, busca que ambos campos valgan 0. Si se revisa el protocolo DNS ambos campos valen 0 cuando se trata de una consulta DNS (query). Por tanto “byte-test:1, ! & , 0xF8, 2;” lo único que hace es verificar que se trata de una consulta DNS.

- Content: en esta parte se comprueba que en el dominio el número hexadecimal que va primero sea el mismo que los caracteres que le siguen, por ejemplo, Facebook tiene 8 caracteres y en hexadecimal corresponde al ”08ƨ com tiene 3 caracteres y en hexadecimal es ”03”. Además, comprueba que la consulta DNS termina siempre con el valor “00”, indicando que no hay más caracteres en el dominio a consultar.

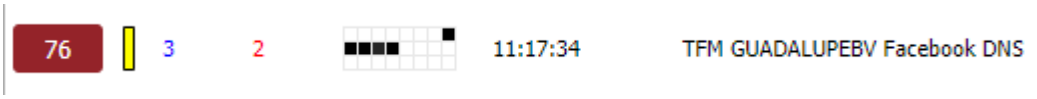


Figura 5.17: Ejemplo de firma DNS en Squert.
Fuente:Elaboración propia

5.1.9. Crear firma TLS

TLS (seguridad de la capa de transporte) y su antecesor Secure Sockets Layer (SSL en español capa de puertos seguros) son protocolos criptográficos, que proporcionan comunicaciones seguras por una red, comúnmente Internet (27).

Normalmente, los sistemas de seguridad ignoran los mensajes cifrados o con características específicas de seguridad, por razones de rendimiento. Por ello, detectar una conexión TLS puede ser importante para alertar a los administradores de URL ocultas.

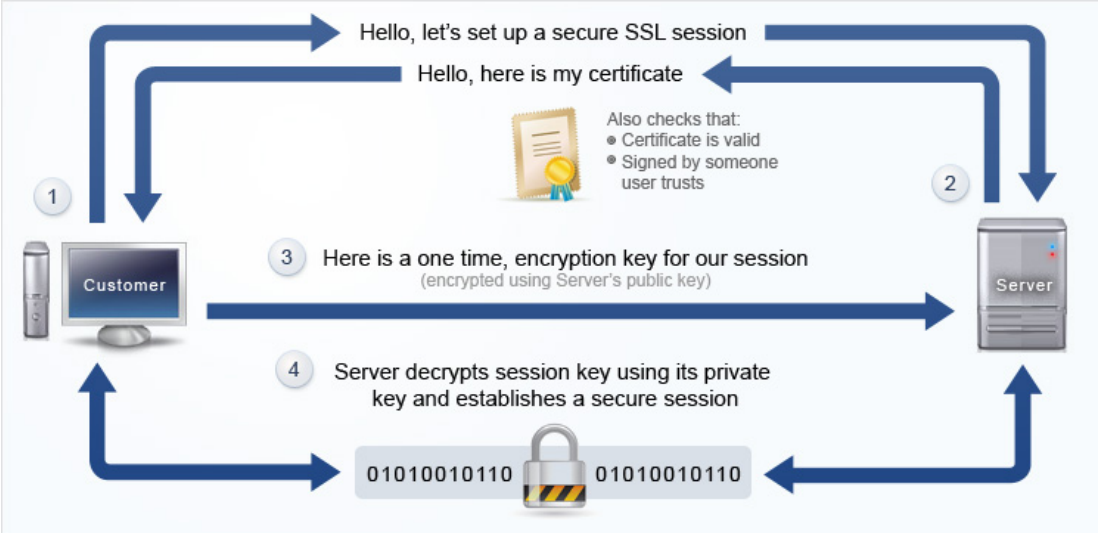


Figura 5.18: TLS.
Fuente:Elaboración propia

El navegador de un consumidor comienza el proceso de reconocimiento de SSL por medio de la solicitud de una página web segura mediante el protocolo HTTPS. Esto inicia una sesión segura con el sitio web mediante el envío de un mensaje del cliente que diga 'Hola' al servidor Web. El mensaje de Hola del cliente contiene información acerca del cifrado y algoritmos de compresión que el navegador soporta, así como un número pseudo-aleatorio. El servidor Web responde con un mensaje de saludo del servidor, que también incluye información acerca de los algoritmos soportados y un número pseudo-aleatorio. El servidor web elige el cifrado más fuerte que tanto el navegador como el servidor soporten. El servidor también envía su certificado digital

en el navegador para dar fe de la identidad de un individuo o de un sistema informático. El servidor web envía entonces un mensaje de saludo del servidor, con lo que indica que ha terminado y queda en espera de una respuesta del navegador.

Una vez que el navegador recibe el mensaje del servidor, se comprueba el certificado con una lista de entidades emisoras de certificados conocidos para asegurarse de que este sea válido. El certificado del servidor contiene su clave pública y el nombre del mismo, que debe coincidir con el nombre del servidor al navegador solicitado. Por ejemplo, si el usuario escribe la dirección URL ”https://www.secureserver.com.^{en} el navegador, el certificado debe contener un nombre de asunto de ”www.secureserver.com.^{o} ”* secureserver.com.».

La criptografía asimétrica (en inglés asymmetric key cryptography), también llamada criptografía de clave pública (en inglés public key cryptography) o criptografía de dos claves1(en inglés two-key cryptography), es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona que recibirá el mensaje. Una clave es pública y se puede entregar a cualquier persona, la otra clave es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. Además, los métodos criptográficos garantizan que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Llave o clave es lo mismo. Existiendo, por tanto: llave o clave privada y llave o clave pública.

Si una persona que emite un mensaje a un destinatario usa la llave pública de este último para cifrarlo; una vez cifrado, sólo la clave privada del destinatario podrá descifrar el mensaje, ya que es el único que debería conocerla. Por tanto, se logra la confidencialidad del envío del mensaje, nadie salvo el destinatario puede descifrarlo. Cualquiera, usando la llave pública del destinatario, puede cifrarle mensajes; los que solo serán descifrados por el destinatario usando su clave privada.

Si el propietario del par de claves usa su clave privada para cifrar un mensaje, cualquiera puede descifrarlo utilizando la clave pública del primero. En este caso se

consigue la identificación y autentificación del remitente, ya que se sabe que sólo pudo haber sido él quien empleó su clave privada (salvo que un tercero la haya obtenido). Esta idea es el fundamento de la firma electrónica, donde jurídicamente existe la presunción de que el firmante es efectivamente el dueño de la clave privada.

Los 'sistemas de cifrado de clave pública' o 'sistemas de cifrado asimétricos' se inventaron con el fin de evitar por completo el problema del intercambio de claves de los sistemas de cifrado simétricos. Con las claves públicas no es necesario que el remitente y el destinatario se pongan de acuerdo en la clave a emplear. Todo lo que se requiere es que, antes de iniciar la comunicación secreta, cada uno debe conseguir la llave pública del otro y cuidar cada uno su llave privada. Es más, esas mismas claves públicas pueden ser usada por cualquiera que desee comunicarse con alguno de ellos siempre que se utilice correctamente la llave pública de cada uno. Por tanto, se necesitarán sólo n pares de claves por cada n personas que deseen comunicarse entre sí. Cada una de las n personas tendrá su clave privada y n-1 llaves públicas (distintas) si quiere enviar mensajes a todas las n-1 personas restantes.

Snort debe ignorar el tráfico cifrado por razones de rendimiento y para reducir los falsos positivos. El preprocesador dinámico SSL (SSLPP) inspecciona el tráfico SSL y TLS y, opcionalmente, determina si se debe detener la inspección y cuándo.

Normalmente, SSL se usa sobre el puerto 443 como HTTPS. Al permitir que SSLPP inspeccione el puerto 443, solo se inspeccionará el protocolo de enlace SSL de cada conexión. Una vez que se determina que el tráfico está cifrado, no se realiza ninguna inspección adicional de los datos de la conexión.

Por lo tanto, para crear una alarma TLS se utilizará Suricata en vez de Snort. El primer paso que se debe hacer es cambiar el IDS de security onion y pasar a Suricata. Para ello se realizará lo siguiente (28):

- sudo so-sensor-stop
- sudo sed -i 's—ENGINE=snort—ENGINE=suricata—g' /etc/nsm/securityonion .conf

- sudo rule-updat
- sudo so-sensor-start Si por el contrario se quiere cambiar de suricata a snort se realizarán los siguientes pasos:
- sudo so-sensor-stop
- sudo sed -i 's—ENGINE=suricata—ENGINE=snort—g' /etc/nsm/securityonion. conf
- sudo rule-update
- sudo so-sensor-start

El archivo de reglas locales de Suricata está en /etc/nsm/rules/local.rules ahí se deben crear nuevas firmas. Además se deberá cambiar el archivo de configuración pulledpork.conf para indicar que las local.rules de suricata está en /etc/nsm/rules/local.rules

Ejemplo de alerta:

```
alert tls $EXTERNAL_NET any -> $HOME_NET any
(tls.fingerprint:"d1:d1:93:3e:21:98:81:20:2f:69:fa:fc:a8:98:bc:eb:3c:61:20:39";
msg:"SSL Twitter Huella";
sid:1008;
rev:1;)
```

Figura 5.19: Ejemplo de firma TLS.
Fuente:Elaboración propia

Distingue entre mayúsculas y minúsculas, no se puede usar 'nocase'. El búfer tls.fingerprint está en minúsculas, por lo que debe usar letras minúsculas para que coincida.

Para averiguar la huella digital podemos hacer varias cosas:

- Consultar la web https://www.dondominio.com/products/ssl/tools/ssl-checker/ donde solamente con poner el dominio salen todos los datos del certificado.

- Acceder a la página web de la que queramos el certificado a través del navegador, al lado de la url saldrá un candado, pinchar en él, ir a la opción de certificado y en detalles se puede ver la huella digital entre otras muchas cosas.

El resultado en squert de la firma TLS sería el siguiente:

Figura 5.20: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

5.2. SECURITY ONION EN MODO PRODUCCIÓN

Hasta ahora todo lo que se ha trabajado con Security Onion es en modo evaluación. A continuación vamos a probar el modo producción: para ello creamos un escenario en el que en uno o varios servidores se instalan y configuran las sondas y en otro servidor de mayor capacidad se ubicará el máster, conviene destacar que este servidor forma parte de una subred distinta, por lo que todo el tráfico que circule por las sondas podrá ser visualizado y analizado en el máster.

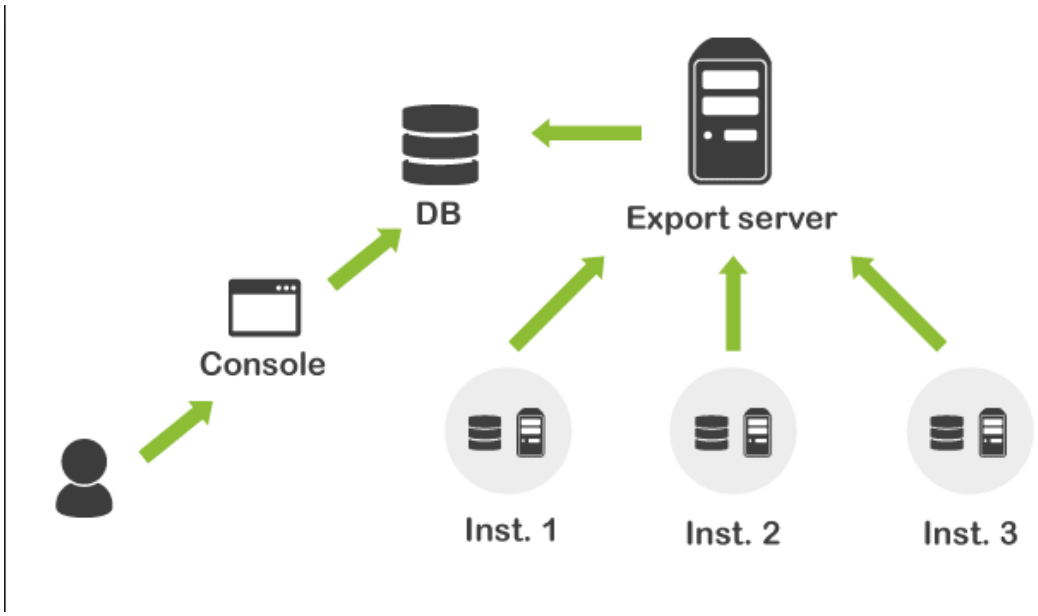


Figura 5.21: Modo producción.
Fuente:Elaboración propia

Las sondas se comunican con el máster mediante un túnel SSH por los puertos 22 y 7736 y mediante SALT por los puertos 4505 y 4506.

- SSH (Secure SHell): es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota, tales como FTP o Telnet, SSH cifra la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. Por defecto se utiliza el puerto 22 y el 7736 en Security Onion

- Salt: ofrece un bus de comunicación dinámico para infraestructuras que se pueden usar para orquestación (término común que se refiere a la coordinación y administración de distintos servicios), ejecución remota, gestión de configuración y mucho más.

OnionSalt es una herramienta creada para administrar múltiples sensores de Security Onion. Salt se configura de forma predeterminada al elegir best practices durante la configuración. Los sensores deben poder conectarse al servidor maestro en los puertos 4505 / tcp y 4506 / tcp.

Para hacer la instalación en modo producción deberemos hacer dos configuraciones:

5.2.1. Instalación modo MASTER

Para instalar el modo máster debemos seguir los siguientes pasos:

- Seleccionar production mode.

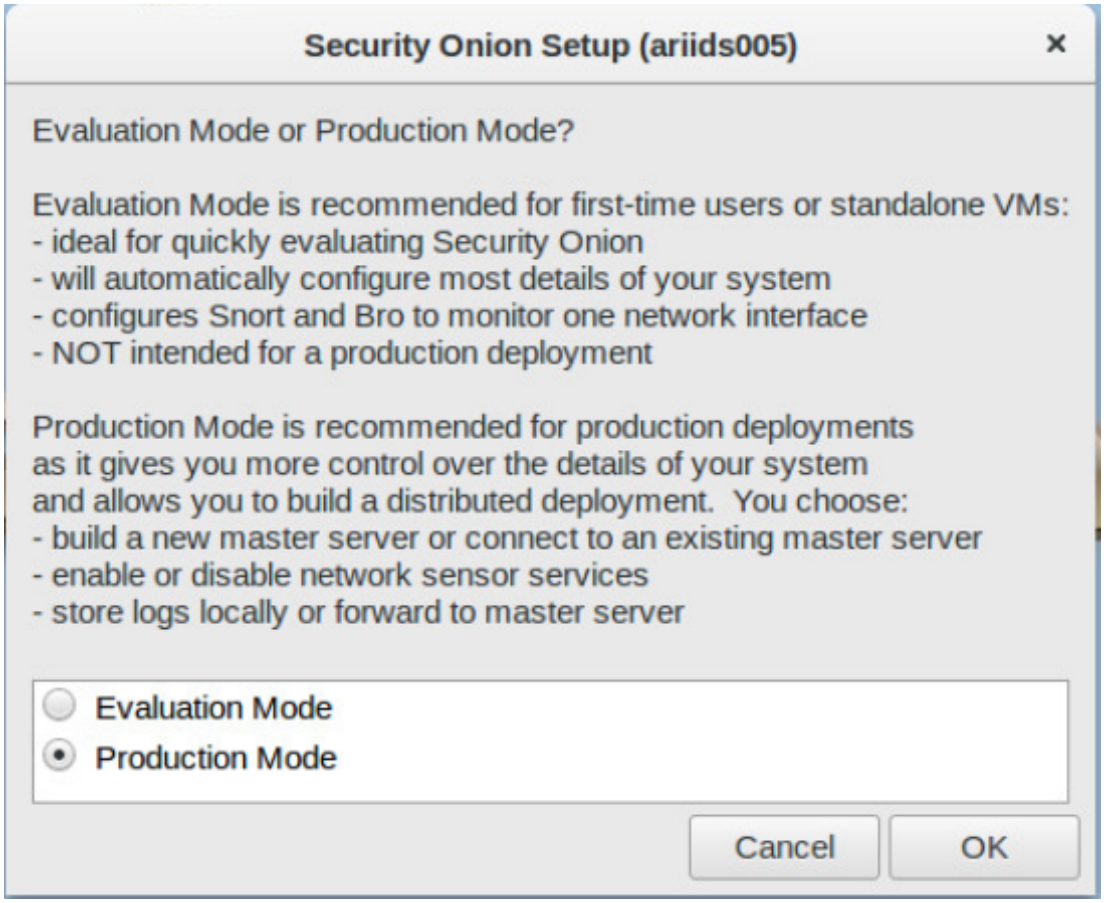


Figura 5.22: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- New Security Onion deployment.

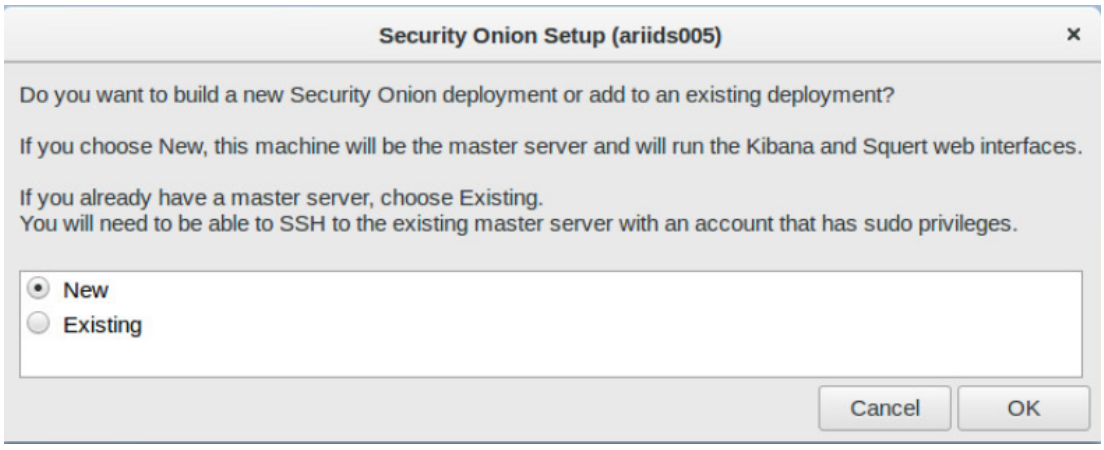


Figura 5.23: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Insertar usuario.

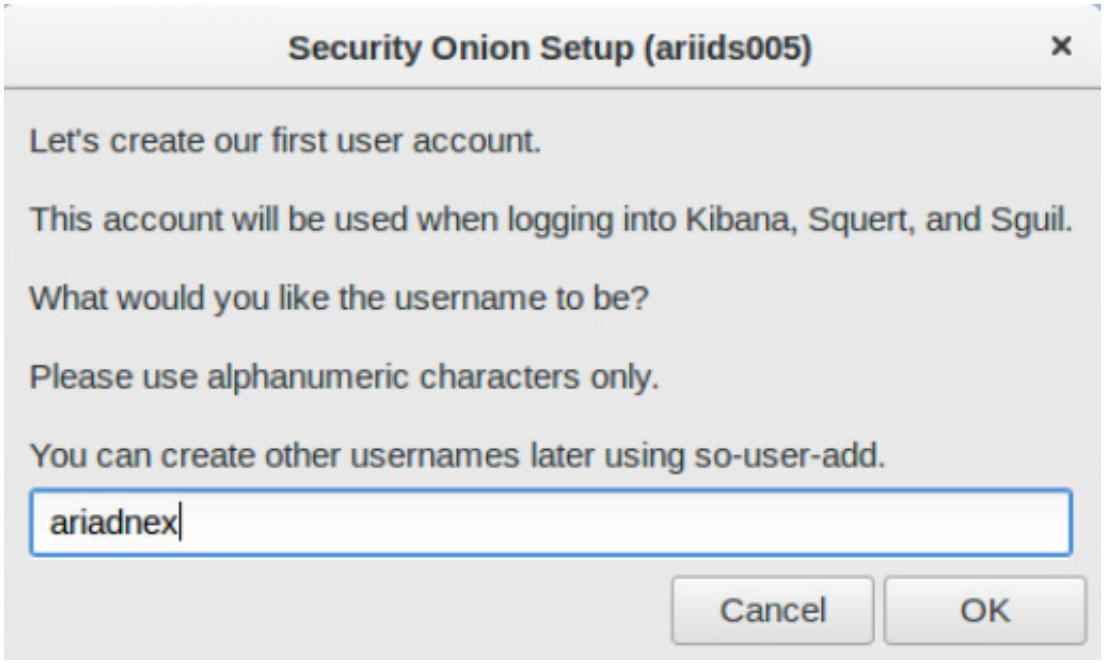


Figura 5.24: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Insertar contraseña.

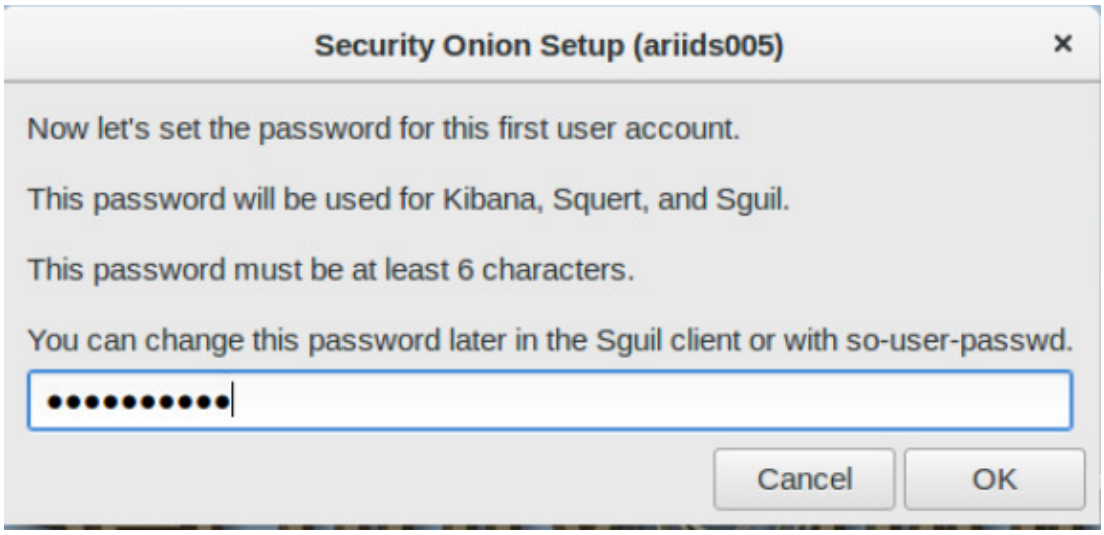


Figura 5.25: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Volver a poner contraseña.

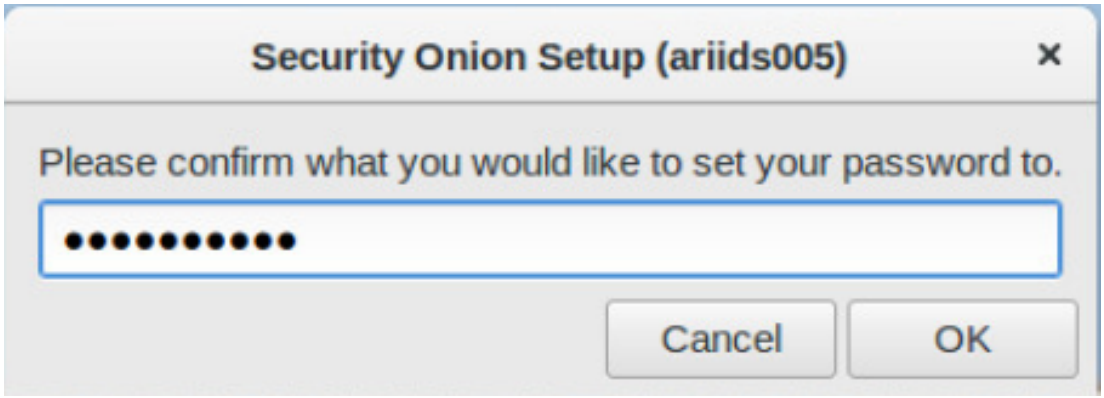


Figura 5.26: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Best practices.

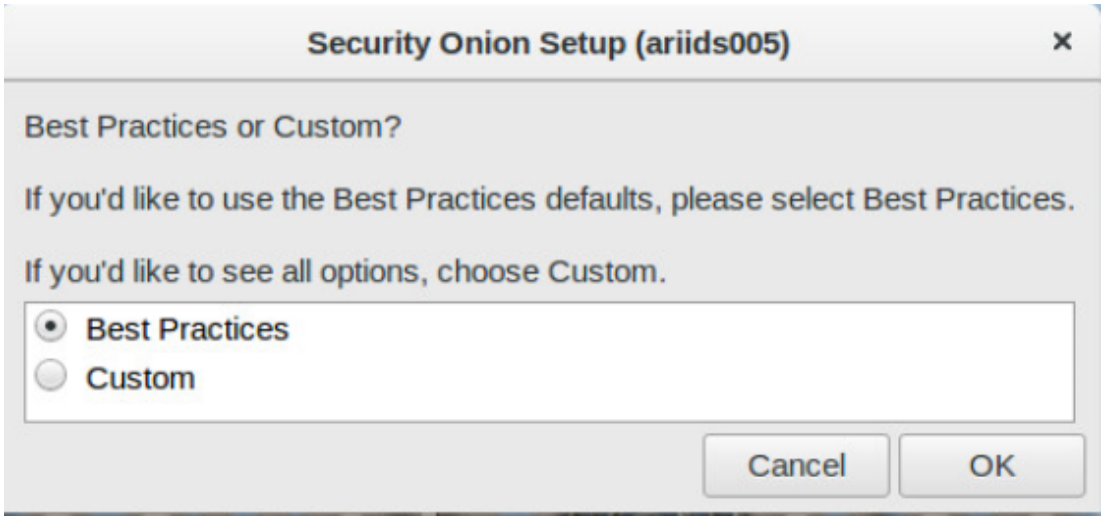


Figura 5.27: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Emerging threats open.

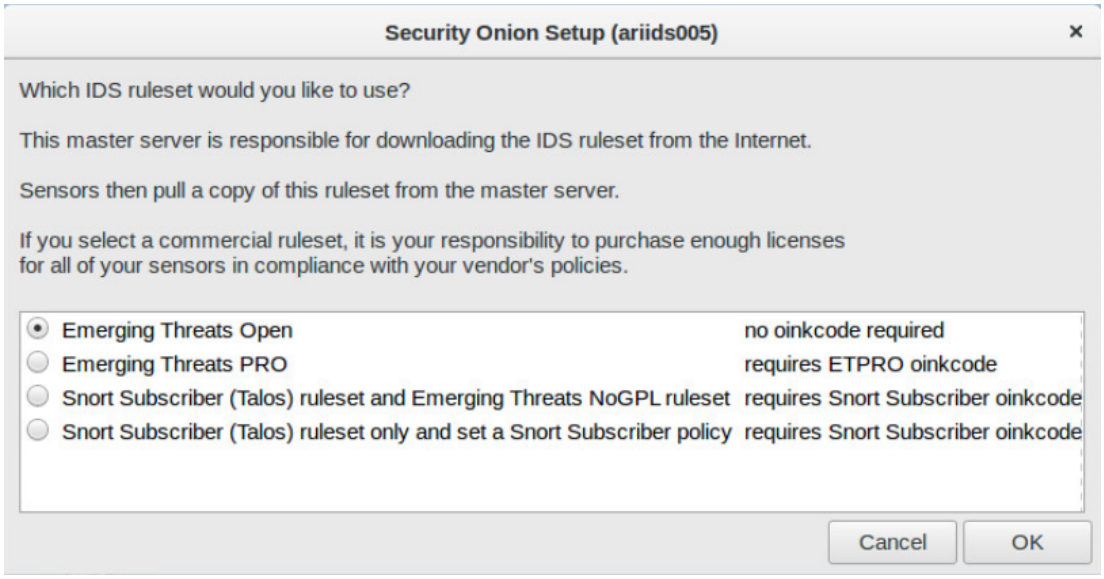


Figura 5.28: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Snort o Suricata. Nosotros elegimos Suricata porque para las firmas con certificado se han probado en múltiples ocasiones durante el proyecto con ambos y con Snort notamos que había múltiples fallos.

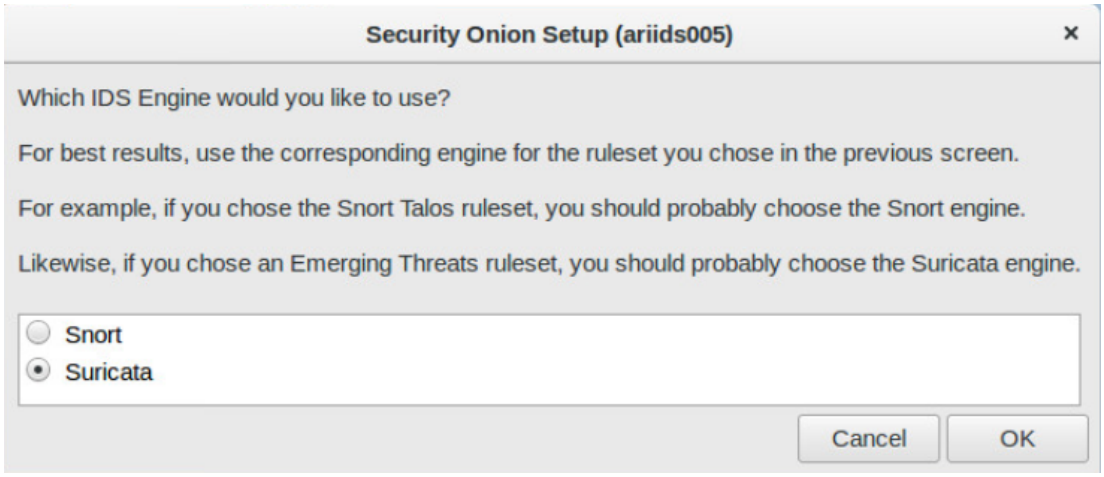


Figura 5.29: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Enable network sensor services.

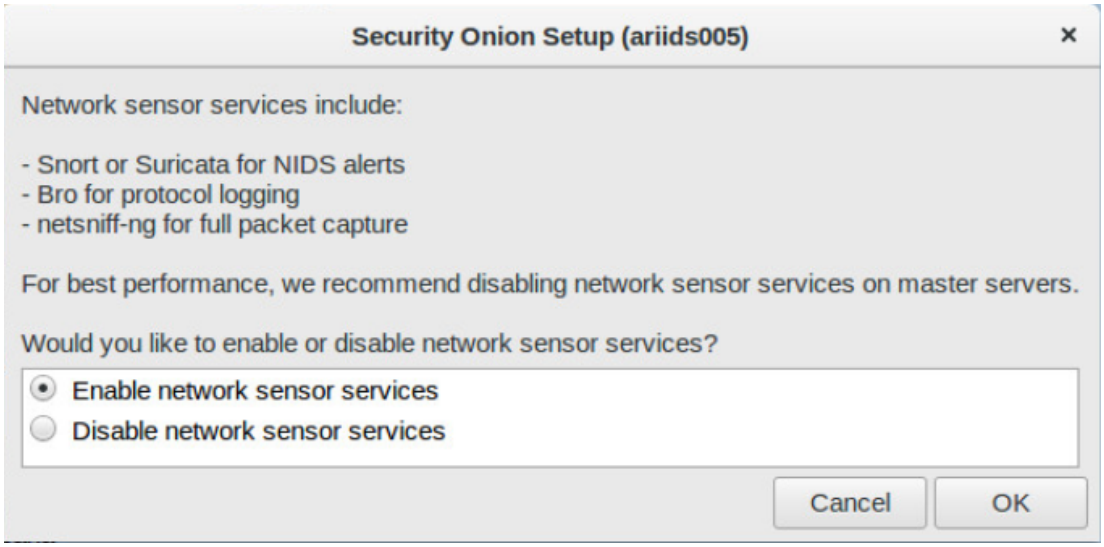


Figura 5.30: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Min num slot: 4096.

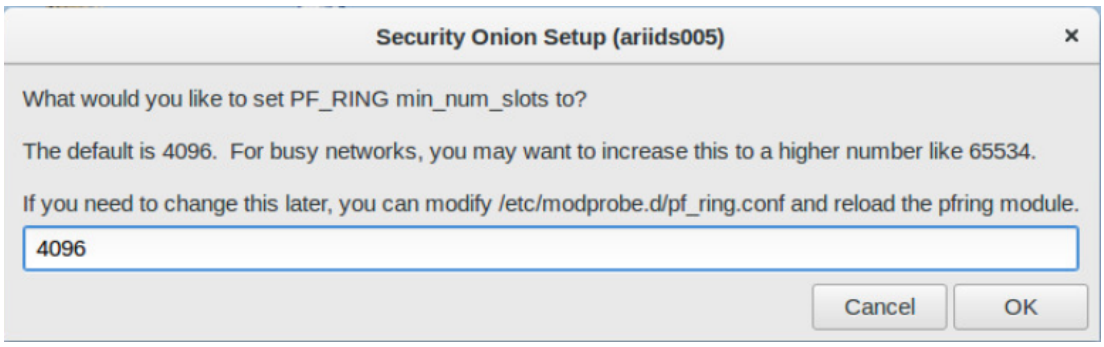


Figura 5.31: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Poner la interfaz del sensor del master.

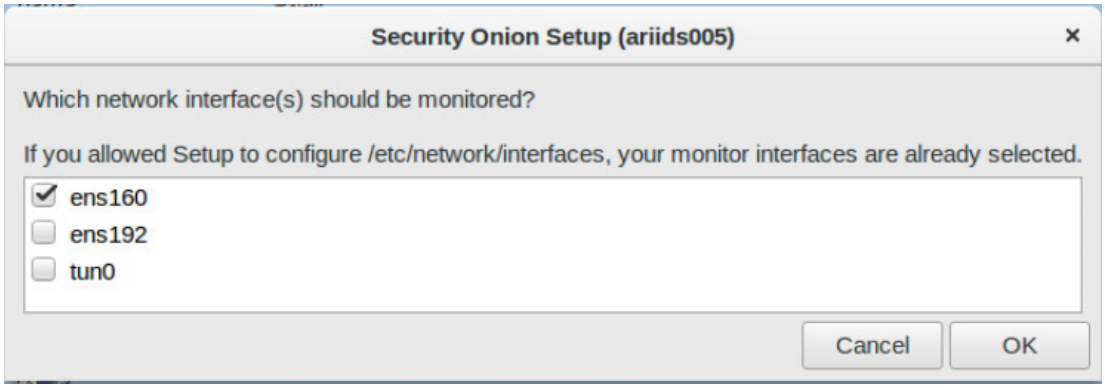


Figura 5.32: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- HOME NET lo que nos indica por defecto.

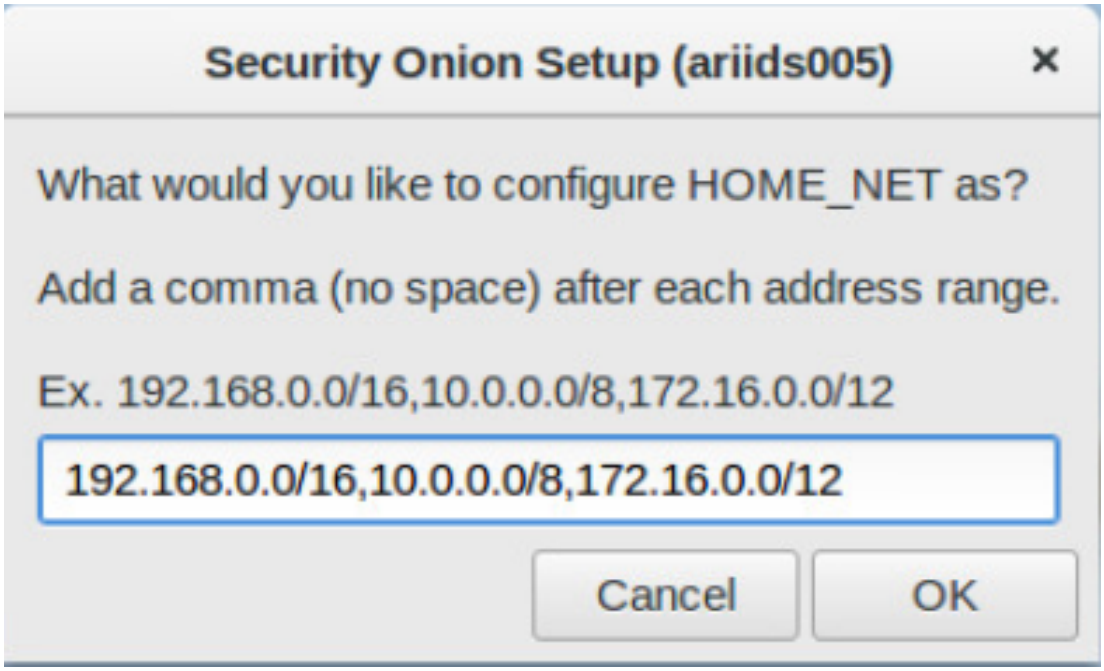


Figura 5.33: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

5.2.2. Instalación modo SONDA

Para instalar el modo sonda debemos seguir los siguientes pasos:

- Seleccionar production mode.

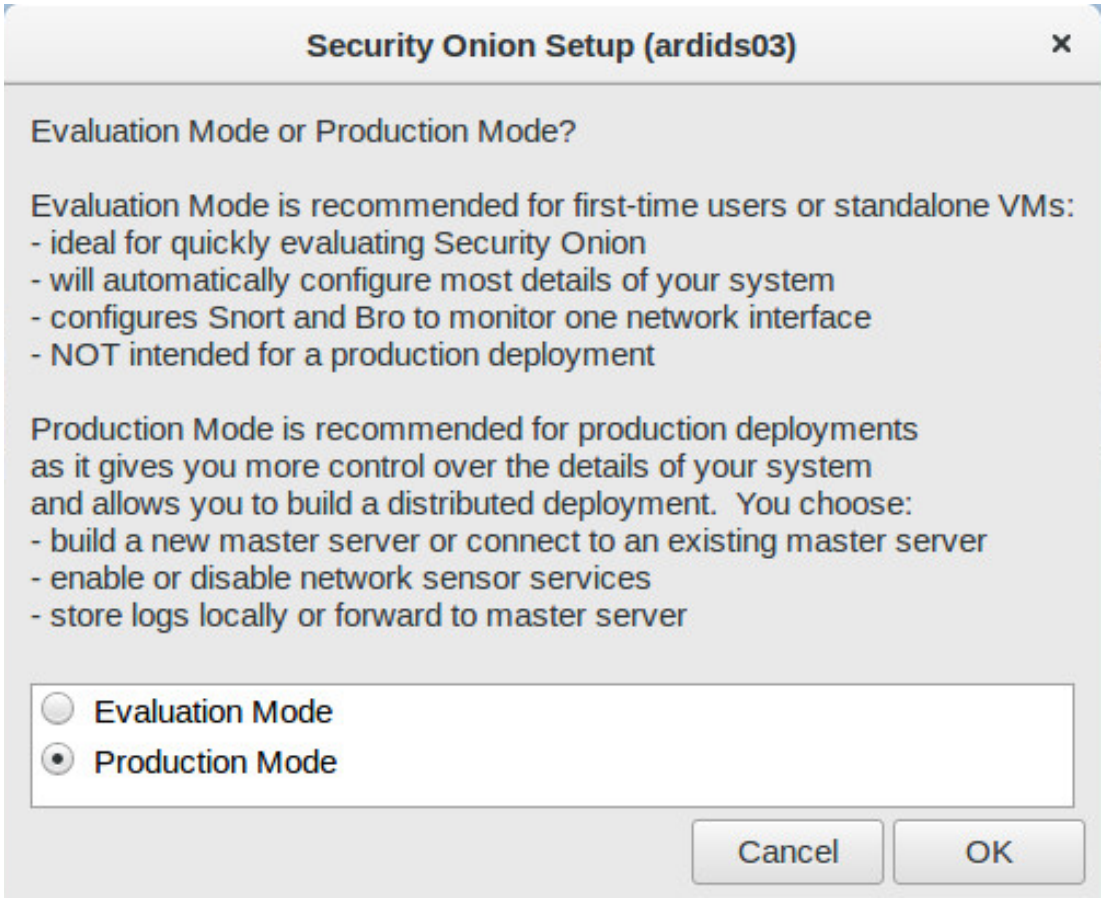


Figura 5.34: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Existing security onion deployment.

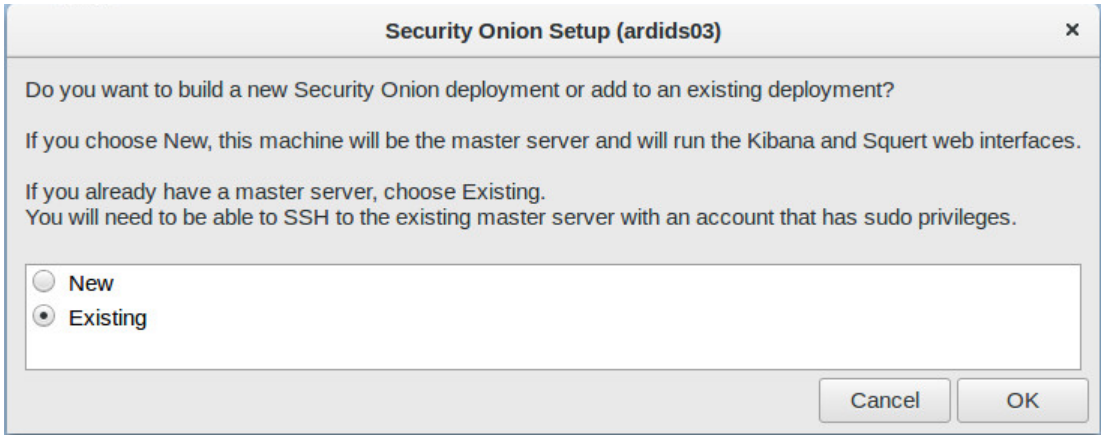


Figura 5.35: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Poner la IP pública del master.

Figura 5.36: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Nombre del usuario SSH.

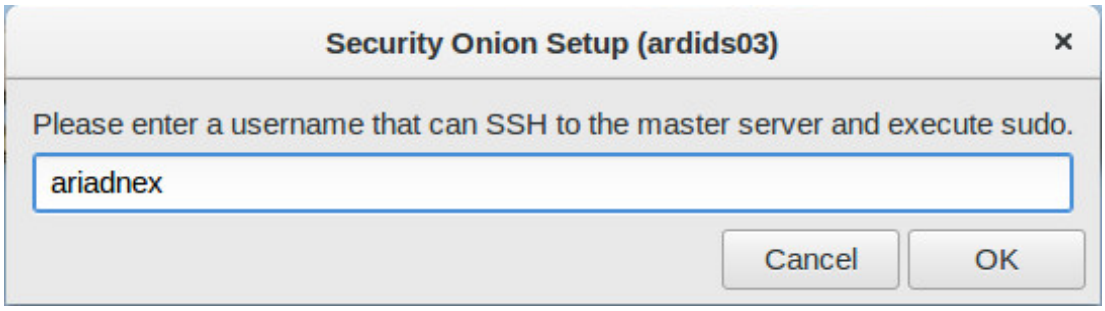


Figura 5.37: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Opción forward.

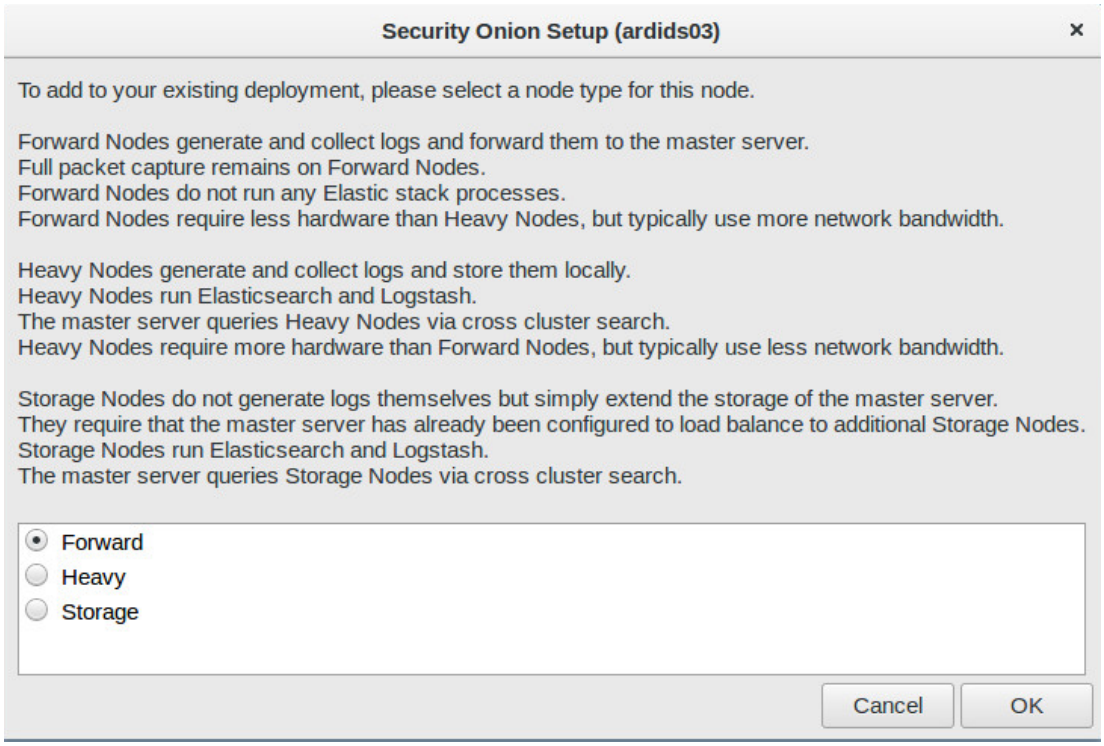


Figura 5.38: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Best practices.

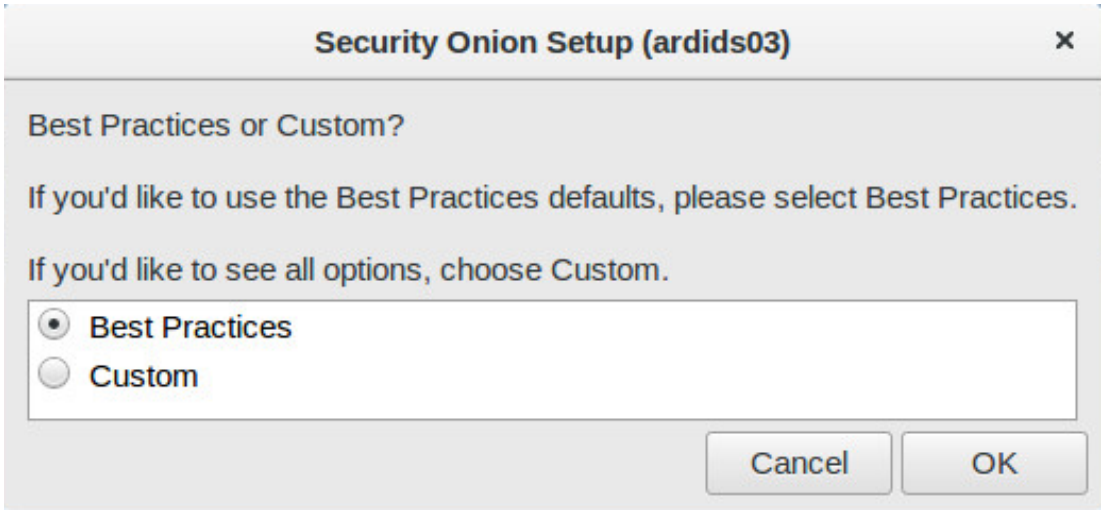


Figura 5.39: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Min num slot: 4096.

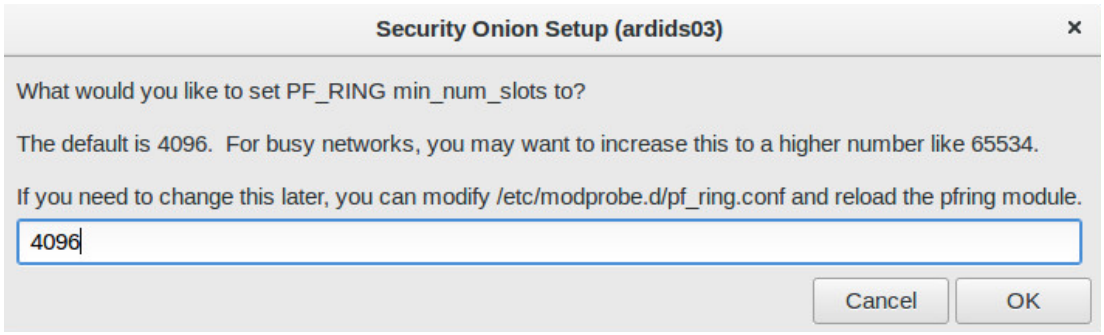


Figura 5.40: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- Poner la interface del sensor de la sonda.

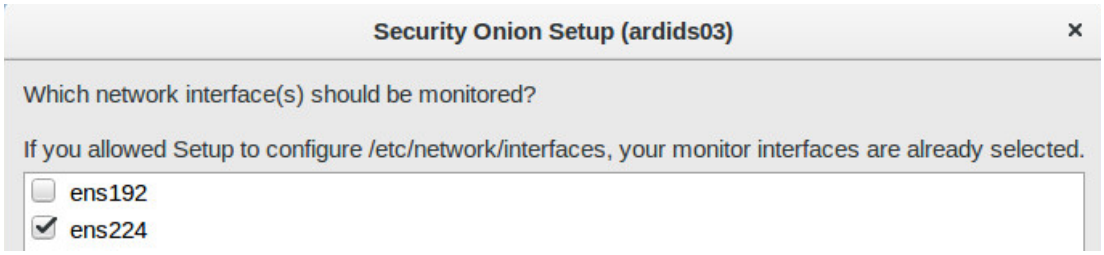


Figura 5.41: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

- HOME NET lo que nos indica por defecto.

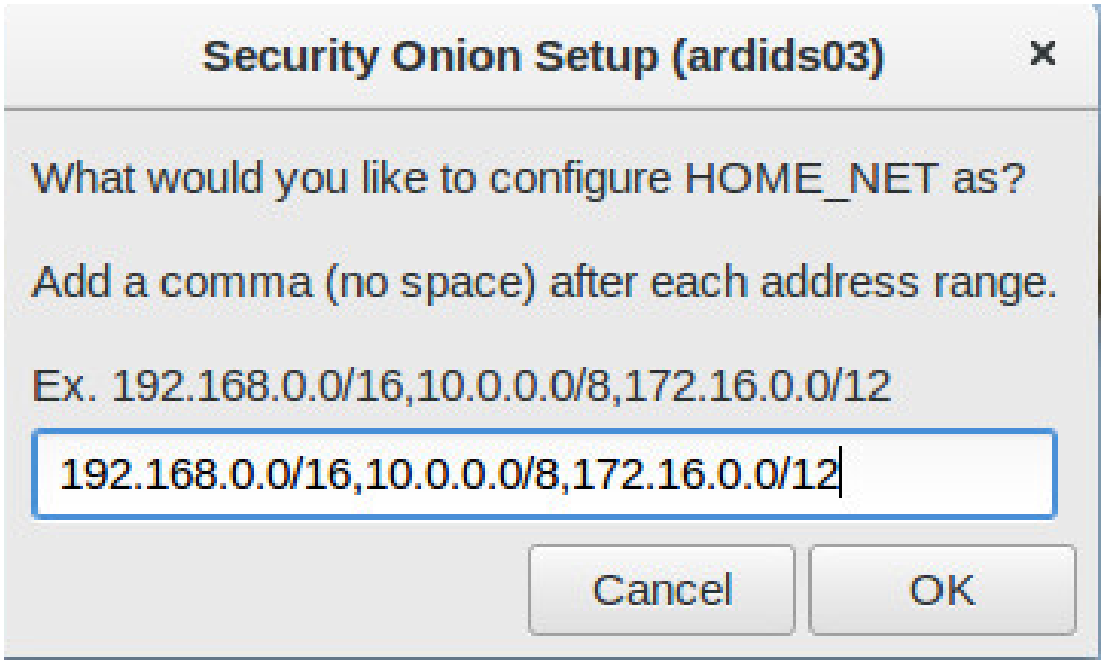


Figura 5.42: Ejemplo de firma TLS en Squert.
Fuente:Elaboración propia

NOTA: Es importante instalar primero el master y después la sonda, ya que la sonda hace el tunel SSH con el master.

5.2.3. Filtrar por sensor

Squert nos muestra las alertas que ha habido en las últimas 24 horas, pero a simple vista no podemos saber de qué sensor proviene, ya que en la interface gráfica no lo especifica en ningún momento. Para ello debemos pulsar en uno de los botones de arriba a la derecha donde está rotulada la palabra sensor. Seguidamente, nos aparecerá una ventana con todos los sensores que están instalados que es donde seleccionaremos el que necesitemos para nuestros propósitos. Entonces será cuando en el entorno SQUERT aparecerán tan solo las alertas del sensor elegido. Obviamente, se pueden seleccionar múltiples sensores y mostrar en SQUERT las alertas, simplemente marcándolos en la ventana descrita.

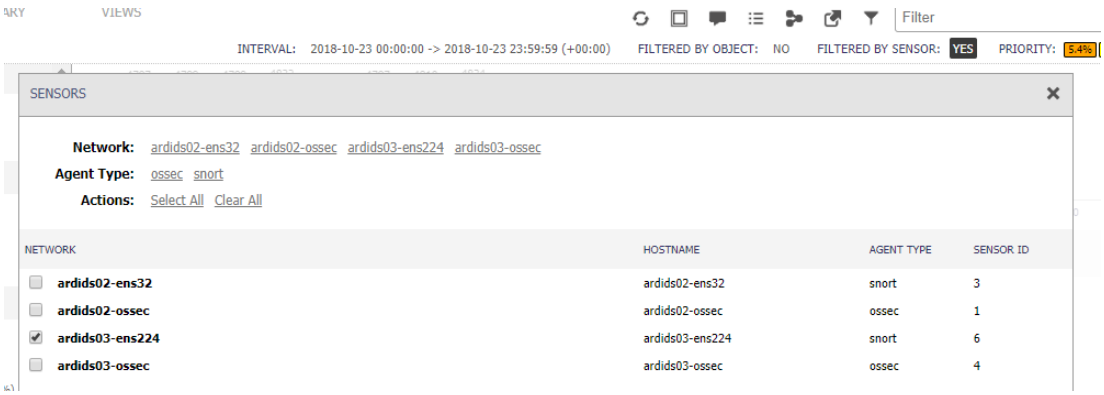


Figura 5.43: Filtro por sensor.
Fuente:Elaboración propia

En Kibana también se puede filtrar por sensores. En este caso debemos ir a device/sensor pulsar la lupa, con el símbolo + dentro, para crear un filtro. Si queremos eliminar un sensor, del mismo modo, pulsaríamos la lupa con el signo - en el interior.

El funcionamiento es el mismo para filtrar alertas o cualquier tipo de dato que necesitemos en cada momento.

5.3. Monitorización de redes sociales

Figura 5.44: Redes sociales.
Fuente:Elaboración propia

La importancia de las Redes Sociales, como ya se ha señalado, merece un estudio particularizado aparte en cuanto la seguridad y su integración en entornos empresariales seguros. Por ello, en este apartado, se verá como monitorizar el tráfico de las redes sociales, para ello nos centraremos en las principales redes sociales de España, que son las siguientes (29):

- Facebook©: es la red social más grande del mundo, con más de 1,4 mil millones de miembros e ingresos anuales superiores a los 12 mil millones de dolares. Facebook© está disponible en todo el mundo.
- Twitter©: permite a sus poco menos de 300 millones de usuarios enviar y recibir mensajes de 140 caracteres que se llaman tweets. Twitter© está disponible en todo el mundo y también ofrece funciones de comercio para hacer compras directamente a través de tweets.

- Whatsapp©: es una red social centrada de mensajería instantánea, con más de 700 millones de miembros a nivel mundial. Fue adquirida por Facebook en febrero de 2014, pero opera como una entidad separada.

- Snapchat©: es una red social de mensajería que permite compartir fotos y vídeos durante un periodo limitado de tiempo definido por el usuario. Cuenta con más de 200 millones de usuarios. El servicio está disponible en más de 15 idiomas.

- LinkedIn©: tiene cerca de 350 millones de miembros. LinkedIn© está disponible a nivel mundial en 20 idiomas diferentes y es utilizado por los profesionales de recursos humanos para la contratación de los candidatos adecuados. También es una red interesante para las marcas.

- Telegram©: es una aplicación de mensajería instantánea. El sitio cuenta con más de 50 millones de usuarios activos, y está disponible en 8 idiomas. Los usuarios pueden enviar mensajes cifrados y que se borran de manera automática con fotos y vídeos.

- Instagram©: es una red social para compartir fotos y de videos con más de 300 millones de miembros. Instagram logró un rápido crecimiento gracias a su capacidad para aplicar fácilmente múltiples filtros para una foto, que puede colocar a diferentes redes sociales, como Facebook© y Twitter©.

- Youtube©: es la red social por excelencia para compartir videos que permite a sus usuarios subir, ver y compartir vídeos. Está disponible en todo el mundo y es uno de los sitios más consultados en la web.

- Skype©: es una red social que permite a los usuarios comunicarse mediante (a) de voz usando un micrófono, (b) de vídeo mediante el uso de una cámara web, y (c) de mensajería instantánea a través de Internet. las llamadas de Skype a Skype son gratuitas, mientras que las llamadas a teléfonos fijos y móviles (a través de redes telefónicas tradicionales) se cargan a través de un sistema de cuentas de

usuario basada en la llamada de débito de crédito de Skype. Skype© tiene más de 300 millones de usuarios activos.

- Tumblr©: es una red social de microblogging con más de 200 millones de miembros. Permite a los usuarios publicar blogs que otros pueden seguir, por lo que es una buena opción para anunciar la introducción de nuevos productos y promociones.

Vivimos en una época en la que utilizamos las redes sociales a diario. Ya sea Twitter©, Facebook©, Instagram© o cualquier otra. Queramos o no, de forma directa o indirecta estamos conectados con el resto del mundo. Sin embargo, no siempre tenemos en cuenta las medidas de seguridad necesarias para mantener nuestra privacidad.

A comienzo de 2018, Facebook© abrió sus puertas al intercambio de aplicaciones de terceros. En este tiempo, millones de sus usuarios han usado pequeñas aplicaciones para jugar o intercambiar recomendaciones de música o películas. En la medida que la popularidad de estas aplicaciones ha ido creciendo, los expertos en seguridad informática han empezado a preocuparse, ya que las redes sociales, además de ser un medio muy eficaz para distribuir aplicaciones informáticas, también lo pueden ser para distribuir código malicioso.

Ya hay en marcha varios proyectos que tratan de demostrar lo real que llega a ser este peligro. El último de ellos ha sido llevado a cabo por la Foundation for Research and Technology Hellas (FORTH). Sus investigadores han creado una aplicación que permite mostrar bonitas fotografías de National Geographic en la página del perfil del usuario de Facebook. Esta aplicación tiene otra propiedad invisible para el usuario: solicita archivos de imágenes de un servidor concreto, en este caso un servidor de pruebas del FORTH. Si muchos usuarios instalaran esta “inocente” aplicación, mandarían sin saberlo miles de peticiones a ese servidor, de tal modo que se bloquearía o sus dueños legítimos no lo podrían usar.

Los investigadores no hicieron ningún esfuerzo para promover esta aplicación entre los usuarios de Facebook y, sin embargo, en sólo unos días 1.000 usuarios ya lo habían

instalado en sus ordenadores. El ataque resultante sobre el servidor usado para el experimento no fue demasiado severo, pero sería suficiente para bloquear una pequeña web, por ejemplo (30).

Sería interesante para cualquier entorno empresarial seguro analizar las conexiones y accesos a estas redes sociales basándonos en las herramientas descritas, de forma que podamos identificar por medio de nuestros sensores posibles vulnerabilidades inmersas en la navegación por la red social en cuestión.

Por esto vamos a realizar la monitorización de estas importantes redes sociales creando las firmas DNS y TLS de cada de ellas.

Como estamos trabajando en modo producción debemos escribir las firmas en el archivo local.rules del máster. Una vez guardadas y comprobado que se han añadido al archivo sid-msg.map, accedemos a la sonda y ejecutamos sudo rule-update. Este comando posibilitará que la sonda actualice la firma copiándolo del máster, esto hará que se copien del máster a la sonda. No es válido escribir las firmas solo en la sonda ya que al hacer sudo-rule update se sobre escriben las reglas del máster en la sonda. Por lo tanto, si no hay firmas en local.rules del máster en la sonda, aparecerá que el archivo también está vacío.

Figura 5.45: Actualización de firmas en la sonda.
Fuente:Elaboración propia

En la figura 33 se ve cómo en el archivo sid-msg.map de la sonda se han generado los sid de las firmas que hemos creado en local.rules del master.

```
  GNU nano 2.5.3        File: /etc/nsm/rules/sid-msg.map

#v1
# sid-msg.map autogenerated by PulledPork - DO NOT MODIFY BY HAND!
1001 || Facebook DNS
1002 || Twitter DNS
1003 || Whatsapp DNS
1004 || Snapchat DNS
1101 || SSL Facebook Huella
1102 || SSL Twitter Huella
1103 || SSL Whatsapp Huella
1104 || SSL Snapchat Huella
```

Figura 5.46: Firmas en sid-msg.map.
Fuente:Elaboración propia

A continuación, vamos a ver un ejemplo de alertas DNS en squert de cuatro de las principales redes sociales:

Figura 5.47: Alertas DNS en squert.
Fuente:Elaboración propia

Se han recibido muchas alertas DNS cada vez que un usuario de la oficina se conectaba a una red social, pero solamente una alerta TLS ya que una vez que ingresas por primera vez en la web, ya la alerta no vuelve a saltar porque la clave pública ya está guardada en nuestro navegador, por lo que no vuelve a pedir la huella digital.

5.4. Monitorización proxys maliciosos

Un proxy, o servidor proxy, en una red informática, es un servidor —programa o dispositivo—, que hace de intermediario en las peticiones de recursos que realiza un cliente (A) a otro servidor (C). Por ejemplo, si una hipotética máquina A solicita un recurso a C, lo hará mediante una petición a B, que a su vez trasladará la petición a C; de esta forma C no sabrá que la petición procedió originalmente de A.

Esta situación estratégica de punto intermedio le permite ofrecer diversas funcionalidades: control de acceso, registro del tráfico, restricción a determinados tipos de tráfico, mejora de rendimiento, anonimato de la comunicación, caché web, etc. Dependiendo del contexto, la intermediación que realiza el proxy puede ser considerada por los usuarios, administradores o proveedores como legítima o delictiva y su uso es frecuentemente discutido. (31)

Para la monitorización de los proxys se han seguido los mismos pasos, que, para la monitorización de las redes sociales, es decir, elegir los proxys maliciosos más utilizados y realizar las firmas DNS y TLS de estos.

Proxys maliciosos (32):

- VirtualShield
- RUS.VPN

- Nord.VPN
- Shadowsocks
- HMA.Pro.VPN
- Lantern
- Setup.VPN
- Hoxx.VPN

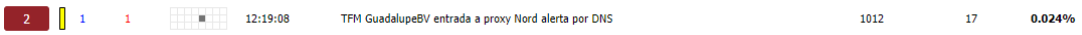


Figura 5.48: Alertas DNS en squert para proxy.
Fuente:Elaboración propia

5.5. Comunicación mediante un tunel VPN

Si bien las pruebas se han realizado hasta ahora en un entorno controlado y localizado, la realidad es que en múltiples ocasiones precisamos de interconectar los elementos de seguridad de forma remota, lo que los convierte a su vez en objetivos de los potenciales atacantes. Esta es la razón por las que se suelen disponer conexiones cifradas entre sedes, arbitrando una red casi paralela de seguridad donde circulan datos, logs, alertas, etc.. Y nada se podría conseguir sin las VPN.

VPN (Virtual Private Network) es una tecnología de red de computadoras que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet. Permite que la computadora en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada con toda la funcionalidad, seguridad y políticas de gestión de una red privada.1 Esto se realiza estableciendo una conexión virtual punto a punto mediante el uso de conexiones dedicadas, cifrado o la combinación de ambos métodos.

Ejemplos comunes son la posibilidad de conectar dos o más sucursales de una empresa utilizando como vínculo Internet, permitir a los miembros del equipo de

soporte técnico la conexión desde su casa al centro de cómputo, o que un usuario pueda acceder a su equipo doméstico desde un sitio remoto, como por ejemplo un hotel. Todo ello utilizando la infraestructura de Internet.

La conexión VPN a través de Internet es técnicamente una unión wide area network (WAN) entre los sitios, pero al usuario le parece como si fuera un enlace privado de allí la designación ”virtual private network”.

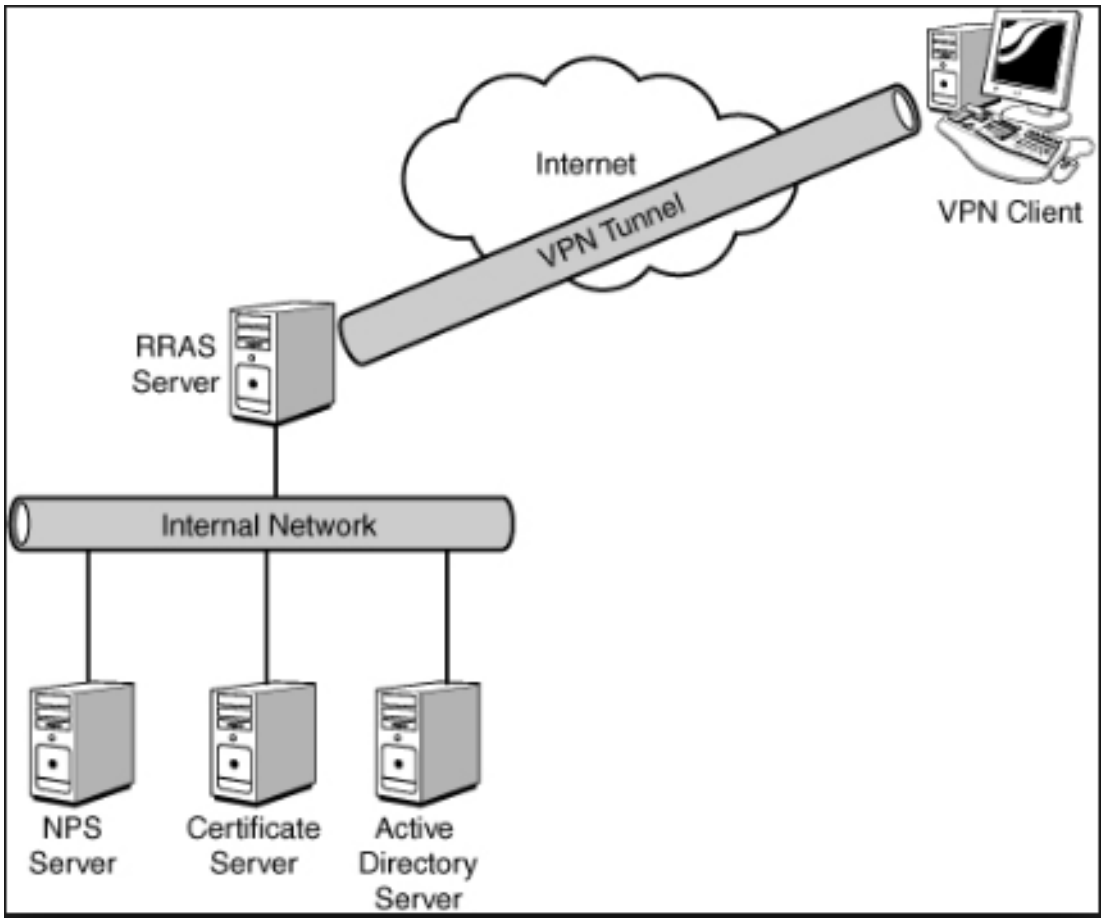


Figura 5.49: VPN.
Fuente:Elaboración propia

OpenVPN ofrece soluciones de VPN flexibles para proteger sus comunicaciones de datos, ya sea para la privacidad de Internet, el acceso remoto para los empleados, la seguridad de IoT o la red de centros de datos en la nube. La seguridad que aportará el servidor OpenVPN está compuesta por 3 capas: (https://geekland.eu/crear-y-configurar-servidor-openvpn/) Capa 1 “Autentificación

TLS": Con la autentificación TLS estamos introduciendo una firma digital HMAC a los paquetes antes de empezar la autentificación reciproca entre cliente y servidor. Si no se pasa el test de la firma HMAC, no se llegará ni a iniciar el proceso de autenticación entre cliente y servidor. Capa 2 "SSL/TLS": Mediante las herramientas de seguridad proporcionadas SSL/TLS se realiza el proceso de autentificación bidireccional entre el cliente y el servidor OpenVPN mediante claves criptográficas. Capa 3 "Cifrado": Dispone de varios tipos de cifrado disponibles en la transmisión de datos entre el cliente y el servidor. Además, se pueden aplicar medidas para los privilegios del servidor de OpenVPN sean los mínimos para poder realizar la función que tiene que realizar.

OpenSSL es un kit de herramientas robusto, de grado comercial y con todas las funciones para los protocolos de Seguridad de la capa de transporte (TLS) y de Capa de sockets seguros (SSL). También es una biblioteca de criptografía de propósito general.

OpenSSL tiene licencia bajo una licencia de estilo Apache, lo que básicamente significa que se puede obtener libre y usarla con fines comerciales y no comerciales, sujeto a algunas condiciones de licencia simples.

5.5.1. Instalación server VPN

Para la instalación del server se seguirán los siguientes pasos: Instalar OpenVPN versión 2.3.10 y OpenSSL versión 1.0.2 mediante el comando:

- sudo apt-get install openvpn openssl.

Crear archivo server.conf en /etc/openvpn/ de la siguiente manera:

- sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
- sudo gzip -d /etc/openvpn/server.conf.gz Antes de crear los certificados hay que estar en el directorio /etc/openvpn para que se creen en esa carpeta y así tener todos los certificados juntos y localizados.

Crear certificado ca.crt:

- sudo openssl genrsa -des3 -out ca.key 2048
- openssl req -x509 -new -nodes -key ca.key -days 2048 -out ca.pem
- sudo openssl req -new -key ca.key -out ca.csr
- sudo openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt

Crear server.crt y server.key:

- sudo openssl genrsa -des3 -out server.key 2048
- sudo openssl req -new -key server.key -out server.csr y completar los datos que pide.
- openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 500

Crear dh2048.pem:

- sudo openssl dhparam -out dh2048.pem 2048

Modificar el archivo server.conf para que tengan los siguientes parámetros:

- port 1194 – Puerto de escucha del servicio. El puerto de escucha se puede modificar.
- proto tcp – Protocolo de la conexión VPN. También podríamos usar el udp.
- dev tun – Dispositivo virtual en el cual se creara el túnel.
- server 10.5.0.0 255.255.255.0 – Indica que los clientes del VPN se les asignará IP del tipo 10.5.0.0/24
- ca /etc/openvpn/ca.crt – Certificado de la autoridad certificadora
- cert /etc/openvpn/server.crt –Certificado del servidor

- key /etc/openvpn/server.key –Clave privada del servidor
- dh /etc/openvpn/dh2048.pem –Carga de los parámetro de Diffie Hellman.
- user nobody –Para limitar los privilegios del demonio de VPN hacemos que funcione con el usuario nobody.
- group nogroup – Para limitar los privilegios del demonio de VPN hacemos que funcione con el grupo nogroup.
- verb 3 – Grado de detalle del estado del túnel en los logs.
- keepalive 10 120 – El servidor VPN enviará un ping cada 10 segundos y como máximo esperará 120 segundos para que el cliente de una contestación.
- client-config-dir /etc/openvpn/ccd
- comp-lzo – Activar compresión LZO para la transmisión de datos.
- persist-key – En caso que el servidor OpenVPN se caiga las claves no tendrán que ser analizadas de nuevo.
- persist-tun –El dispositivo tun0 no tendrá que ser reabierto ni cerrado en el caso que tengamos que reiniciar el servidor.
- client-to-client –los clientes que estan conectados al servidor VPN puedan comunicarse entre ellos

Lanzar el servicio mediante:

- sudo openvpn /etc/openvpn/server.conf

Después de todos estos pasos si se hace un ifconfig aparecerá tun0, si no aparece hacer sudo so-restart y volver a hacer ifconfig.

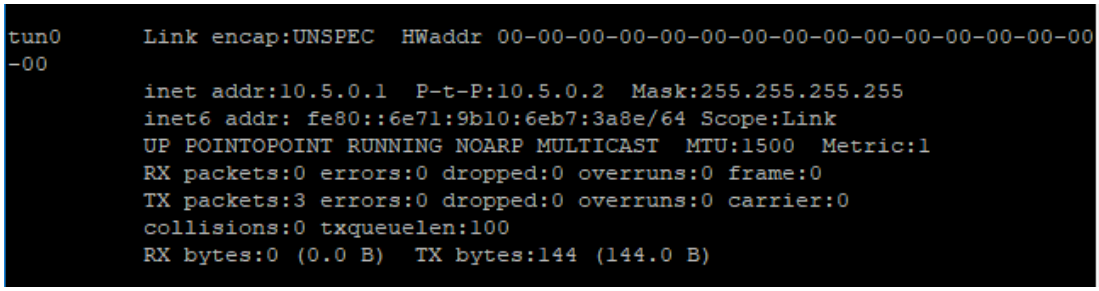


Figura 5.50: TUN 0.
Fuente:Elaboración propia

5.5.2. Intalación cliente VPN

Para la instalación del cliente en la sonda se seguirán los siguientes pasos:

1. Instalar OpenVPN versión 2.3.10 y OpenSSL versión 1.0.2 mediante el comando:

- sudo apt-get install openvpn openssl.

2. Crear archivo client.conf en /etc/openvpn/ de la siguiente manera:

- sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf.gz /etc/openvpn/
- sudo gzip -d /etc/openvpn/client.conf.gz

3. Modificar el archivo client.conf para que tenga los siguientes parámetros:

- Client –indica que este archivo es el del cliente
- Dev tun –Dispositivo virtual en el cual se creará el túnel.
- Proto tcp – Dispositivo virtual en el cual se creara el túnel.
- Remote 141.136.59.252 –IP pública del server
- Resolv-retry infinite – El cliente intentará de forma indefinida resolver la dirección o nombre de host indicado por la directiva remote

- Nobind –A los clientes se les asignará puertos dinámicos (no privilegiados) cuando haya retorno de paquetes del servidor al cliente.
- Persist-key –En caso que el servidor OpenVPN sea reiniciado no se tendrán que volver a leer las claves.
- Persist-tun – El dispositivo tun0 no tendrá que ser reabierto ni cerrado en el caso que tengamos que reiniciar el cliente Vpn.
- Ca /etc/openvpn/ca.crt – Certificado de la autoridad certificadora
- Cert /etc/openvpn/client.crt –Certificado del cliente
- Key /etc/openvpn/client.key – Clave privada del cliente
- Cipher AES-256-CBC – Por defecto el algoritmo de cifrado de OpenVPN es Blowfish con un tamaño de clave de 128 bits. Quien crea que no es suficiente puede añadir esta línea para cambiar el algoritmo de cifrado a AES con un clave de cifrado de 256 bits.
- Comp-lzo – Activar compresión LZO para la transmisión de datos.
- Verb 3 – Grado de detalle del estado del túnel

4. Llevar los certificados del server al cliente mediante scp:

- scp /etc/openvpn/ca.crt ariadnex@192.168.2.112: /
- scp /etc/openvpn/client.crt ariadnex@192.168.2.112: /
- scp /etc/openvpn/client.key ariadnex@192.168.2.112: /

Hay que poner que guarde el archivo en /home/ariadnex/ porque es en la única carpeta que se puede escribir con scp, las demás carpetas necesitan permisos especiales, otra opción es ponerlo como en la figura.

```
root@ARIIDS005:/# sudo scp /etc/openvpn/ca.crt ariadnex@192.168.2.112:~/
ariadnex@192.168.2.112's password:
ca.crt                                        100%  920     0.9KB/s   00:00
root@ARIIDS005:/# sudo scp /etc/openvpn/server.crt ariadnex@192.168.2.112:~/
ariadnex@192.168.2.112's password:
server.crt                                    100%  920     0.9KB/s   00:00
root@ARIIDS005:/# sudo scp /etc/openvpn/server.key ariadnex@192.168.2.112:~/
ariadnex@192.168.2.112's password:
server.key                                    100%  963     0.9KB/s   00:00
```

Figura 5.51: Paso de certificados.
Fuente:Elaboración propia

5. Lanzar el cliente:

- openvpn /etc/openvpn/client.conf

```
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
-00
          inet addr:10.5.0.6  P-t-P:10.5.0.5  Mask:255.255.255.255
          inet6 addr: fe80::ebfc:ee4e:696c:1e07/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:48 (48.0 B)  TX bytes:288 (288.0 B)
```

Figura 5.52: Interfaz TUN0 en el cliente.
Fuente:Elaboración propia

Para que la sonda y el máster se puedan comunicar mediante ping y ssh hay que permitir las conexiones de la interfaz TUN0 en el máster. Para ello se escribirá en la consola del máster: iptables -A INPUT -i tun0 -j ACCEPT y después reiniciar el máster mediante el comando sudo reboot.

Se puede comprobar que el túnel VPN funciona haciendo ping desde la sonda a la ip del túnel en el máster, que en este caso es la 10.5.0.1.

```
ariadnex@ARDIDS03:~$ ping 10.5.0.1
PING 10.5.0.1 (10.5.0.1) 56(84) bytes of data.
64 bytes from 10.5.0.1: icmp_seq=1 ttl=64 time=22.5 ms
64 bytes from 10.5.0.1: icmp_seq=2 ttl=64 time=98.7 ms
64 bytes from 10.5.0.1: icmp_seq=3 ttl=64 time=23.0 ms
64 bytes from 10.5.0.1: icmp_seq=4 ttl=64 time=23.2 ms
64 bytes from 10.5.0.1: icmp_seq=5 ttl=64 time=23.6 ms
^Z
[1]+  Stopped                 ping 10.5.0.1
ariadnex@ARDIDS03:~$
```

Figura 5.53: Ping.
Fuente:Elaboración propia

El último paso para tener el túnel implementado en Security Onion es configurar la sonda para que envíe todos los datos a través del túnel de forma que sea más seguro el envío de datos, ya que no sólo envía por el puerto de ssh, sino que también utiliza los puertos de salt que no van cifrados. Para ello configuramos la sonda en modo producción como se ha explicado anteriormente, con la única diferencia que cuando pida la dirección IP del máster se debe poner la dirección IP del túnel del máster en vez de la dirección IP pública del máster.

Figura 5.54: IP del túnel.
Fuente:Elaboración propia

Una vez configurado, se podrán visualizar en squert los sensores que están enviando datos al máster y entonces se podrá comprobar que efectivamente la VPN funciona correctamente, puesto que deben aparecer los sensores del máster y de las sondas.

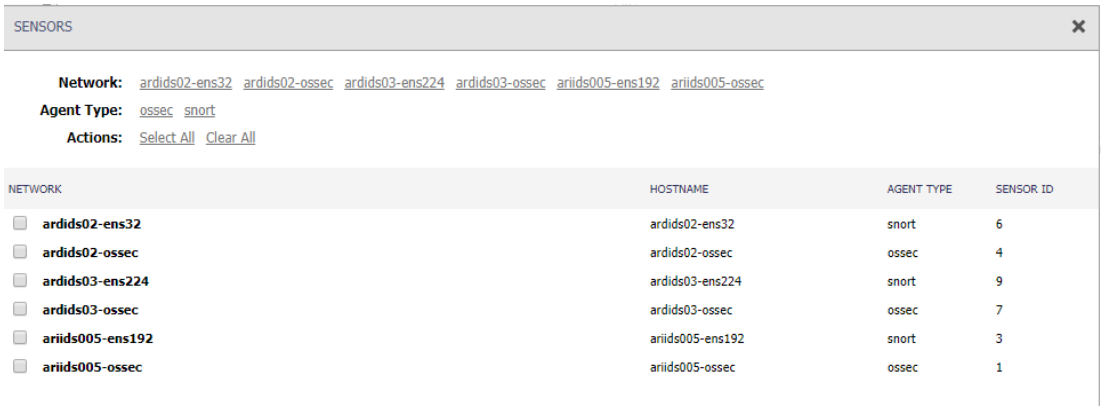


Figura 5.55: Sensores en squert.
Fuente:Elaboración propia

La última comprobación que se debe hacer es que efectivamente reciba las alertas.

Figura 5.56: Alertas en el master de las sondas.
Fuente:Elaboración propia

5.6. Agente OSSEC

El Agente OSSEC es un HIDS que permite complementar nuestra infraestructura de seguridad aportando información a nuestros sistemas recolectores de los comportamientos internos de los sistemas (O ALGO ASÍ QUE TÚ CREAS). A continuación detallaremos la instalación del agente OSSEC y su integración con Security Onion.

Instalación de un agente OSSEC en un portátil para la monitorización de las alertas OSSEC en el máster.

Primero se instalará en el portátil el agente de Windows versión 3.1.0.

Segundo añadir agente en la sonda para ello, ir a /var/ossec/bin e inicializar manage-agents, esto dará varias opciones, Para agregar un tipo de agente debe pulsar

A en la pantalla de inicio, a continuación, se le solicitará que proporcione un nombre para el nuevo agente. Este puede ser el nombre de host u otra cadena para identificar el sistema. Después de eso tienes que especificar la dirección IP para el agente. Puede ser una dirección IP única (por ejemplo, 192.168.1.25), un rango de direcciones IP (por ejemplo, 192.168.1.0/24), o any. El uso de un rango de red o any es preferible cuando la IP del agente puede cambiar con frecuencia (DHCP).La última información que se le pedirá es la identificación que desea asignar al agente. manage-agents sugerirá un valor para el ID. Este valor debe ser el número positivo más bajo que aún no esté asignado a otro agente. El ID 000 está asignado al servidor OSSEC. Para aceptar la sugerencia, simplemente presione ENTER. Para elegir otro valor, escríbalo y presione ENTER.

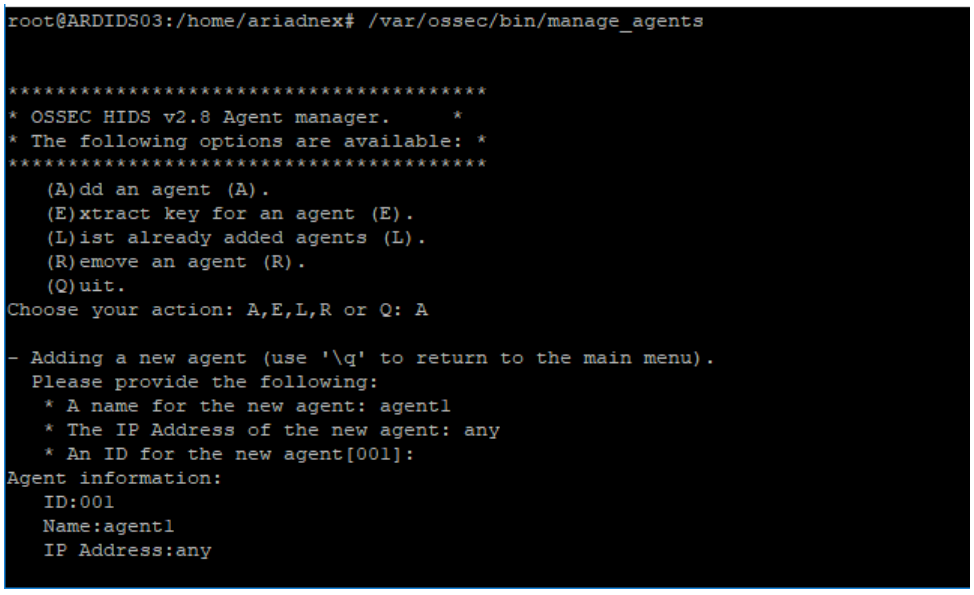


Figura 5.57: manageagents opción A

Después de agregar un agente, se crea una clave. Esta clave debe ser copiada al agente. Para extraer la clave, use la opción en la pantalla de inicio de manage-agents. Se le dará una lista de todos los agentes en el servidor. Para extraer la clave de un agente, simplemente escriba la ID del agente. Es importante tener en cuenta que debe ingresar todos los dígitos de la ID.

Una vez creado el agente y copiada la clave, ya se puede acceder al agente de Windows.

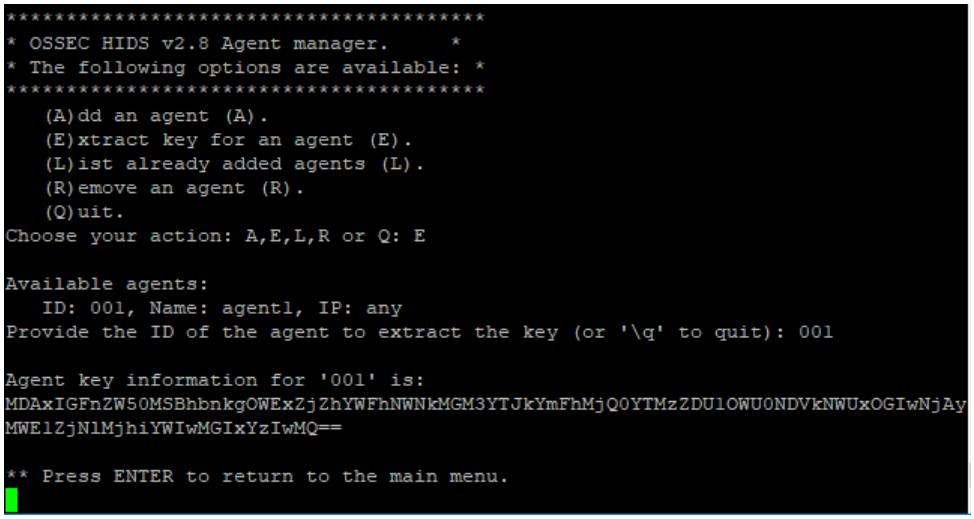


Figura 5.58: Clave de agente

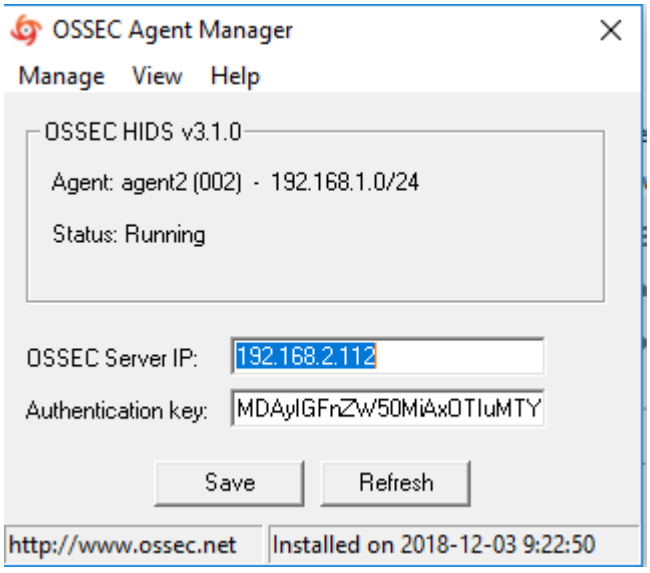


Figura 5.59: Agente ossec en Windows

Es importante abrir en la sonda el puerto de OSSEC (1514), mediante sudo: ufw allow 1514.

Ejemplo de alerta detectada por squert de ossec.

Ahora vamos a comprobar que esa alerta está en los archivos de la sonda y que pertenece al agente creado en el portátil.

Como se puede observar en las figuras a las 12:21:42 hay archivos del agent instalado en el portátil. Esto quiere decir que el agent ha mandado información a

Figura 5.60: Alerta es squert

```
 GNU nano 2.5.3     File: /var/ossec/logs/archives/archives.log

2018 Dec 03 12:21:06 (agent3) 192.168.1.0->WinEvtLog 2018 Dec 03 13:21:01 WinEv$
2018 Dec 03 12:21:06 (agent3) 192.168.1.0->WinEvtLog 2018 Dec 03 13:21:01 WinEv$
2018 Dec 03 12:21:09 ARIIDS005->/var/log/syslog Dec  3 12:21:09 ARIIDS005 ntpd[$
2018 Dec 03 12:21:29 ARIIDS005->/var/log/apache2/access.log 192.168.1.20 - aria$
2018 Dec 03 12:21:42 (agent3) 192.168.1.0->WinEvtLog 2018 Dec 03 13:21:36 WinEv$
2018 Dec 03 12:21:42 (agent3) 192.168.1.0->WinEvtLog 2018 Dec 03 13:21:37 WinEv$
2018 Dec 03 12:21:42 (agent3) 192.168.1.0->WinEvtLog 2018 Dec 03 13:21:37 WinEv$
2018 Dec 03 12:21:59 ARIIDS005->/var/log/apache2/access.log 192.168.1.20 - aria$
```

Figura 5.61: Archives.log

la sonda, la sonda lo ha detectado como alerta y a las 12:21:42 esta alerta de Ossec aparece en el log de alertas y en el squert.

```
 GNU nano 2.5.3      File: /var/ossec/logs/alerts/alerts.log

Rule: 5501 (level 3) -> 'Login session opened.'
Dec  3 12:20:01 ARIIDS005 su[20351]: pam_unix(su:session): session opened for u$

** Alert 1543839609.168073: - pam,syslog,
2018 Dec 03 12:20:09 ARIIDS005->/var/log/auth.log
Rule: 5502 (level 3) -> 'Login session closed.'
Dec  3 12:20:08 ARIIDS005 su[20351]: pam_unix(su:session): session closed for u$

** Alert 1543839702.168303: - windows,system_error,
2018 Dec 03 12:21:42 (agent3) 192.168.1.0->WinEvtLog
Rule: 18103 (level 5) -> 'Windows error event.'
User: guada
2018 Dec 03 13:21:36 WinEvtLog: System: ERROR(10016): DCOM: guada: DESKTOP-1UCS$
```

Figura 5.62: Alerts.log

Como se ha podido comprobar, en la instalación de los IDS, si bien son herramientas muy completas, siempre adolecen de una perspectiva global que nos obliga a integrar herramientas que nos permitan ver otros tipos de alertas.

5.7. Diagrama de Gantt del proyecto

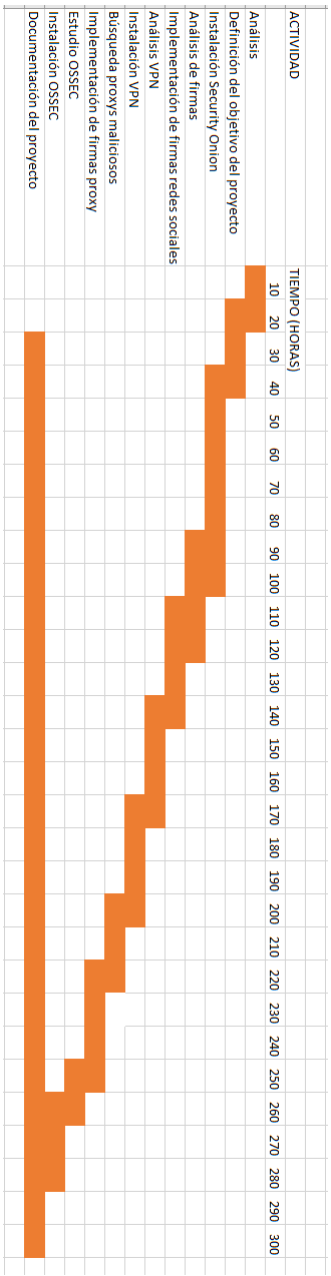


Figura 5.63: Diagrama de Gantt

Capítulo 6

Conclusiones y trabajos futuros

- Suricata es más completo que Snort ya que nos deja crear firmas TLS.
- Squert tiene una interfaz gráfica más intuitiva para ver las alertas que Kibana, en la que es más complejo encontrar la alerta que deseamos.
- Las alertas DNS son más efectivas que las TLS al monitorizar una web, ya que con DNS te avisa cada vez que entran a la web y TLS solamente te notifica cuando es la primera vez que ingresan en la web, las siguientes veces ya no notificaría porque el navegador ya tiene el certificado SSL y no hace esa petición.
- La integración de una VPN para la comunicación entre master y sonda es mucho más efectivo, ya que nos garantiza que los datos no se puedan perder ni interceptar.
- El agente OSSEC nos informa de todos los eventos que ocurran en cualquier ordenador de la oficina, por ejemplo si alguien se instala un software sin licencia sabremos a ciencia cierta quien ha sido.
- Ntop es más sencillo de implementar que Trisul en Security Onion, ya que no necesita realizar cambios en archivos de configuración y además se configura automáticamente, no como en el caso de Trisul que hay que cambiar el sensor que viene por defecto.

Como trabajos futuros se pueden considerar los siguientes:

- Instalación en modo sonda en todas las máquinas de la empresa.
- Instalación del agente OSSEC en todos los ordenadores de la empresa.
- Implementar más herramientas en security onion, para tener una monitorización más completa.

Anexos

Apéndice A

Anexo1. Firmas TLS y DNS utilizadas

alert tls EXTERNAL-NET any -¿HOME-NET any (tls.fingerprint:”bd:25:8c:1f:62:a4:a6:d9:cf:7d:98:12:d2:2e:2f:f5:7e:84:fb:36”; msg:”SSL Facebook Huella TFM GUADALUPEBV”; sid:1101; rev:1;)

alert tls EXTERNAL-NET any -¿HOME-NET any (tls.fingerprint:”d1:d1:93:3e:21:98:81:20:2f:69:fa:fc:a8:98:bc:eb:3c:61:20:39”; msg:”SSL Twitter Huella TFM GUADALUPEBV ”; sid:1102; rev:1;)

alert tls EXTERNAL-NET any -¿HOME-NET any (tls.fingerprint:”dc:cc:93:52:be:93:bf:20:ea:15:44:a6:42:28:47:d2:26:fc:a5:80”; msg:”SSL Whatsapp Huella TFM GUADALUPEBV ”; sid:1103; rev:1;)

alert tls EXTERNAL-NET any -¿HOME-NET any (tls.fingerprint:”9b:86:90:ea:00:5e:8f:12:8f:15:cb:89:02:3b:ae:3d:d4:f1:d6:8e”; msg:”SSL Snapchat Huella TFM GUADALUPEBV ”; sid:1104; rev:1;)

alert tls EXTERNAL-NET any -¿HOME-NET any (tls.fingerprint:”be:69:da:00:a2:38:6d:80:38:2b:d3:e3:17:0a:68:09:e1:ef:ad:ae”; msg:”SSL Tumblr Huella TFM GUADALUPEBV ”; sid:1105; rev:1;)

alert tls EXTERNAL-NET any -¿HOME-NET any (tls.fingerprint:”37:21:36:77:50:57:f3:c9:28:d0:f7:fa:4c:05:35:7f:60:c1:20:44”; msg:”SSL Telegram Huella TFM GUADALUPEBV ”; sid:1106; rev:1;)

alert udp any any -¿any 53 (msg:”Facebook DNS TFM GUADALUPEBV ”;

byte-test:1,!,0xF8,2; content:”—08—facebook—03—com—00—”; fast-pattern: only; sid:1001; rev:1;)

alert udp any any -¿any 53 (msg:”Twitter DNS TFM GUADALUPEBV ”; byte-test:1,!,0xF8,2; content:”—07—twitter—03—com—00—”; fast-pattern: only; sid:1002; rev:1;)

alert udp any any -¿any 53 (msg:”Whatsapp DNS TFM GUADALUPEBV ”; byte-test:1,!,0xF8,2; content:”—03—web—08—whatsapp—03—com—00—”; fast-pattern: only; sid:1003; rev:1;)

alert udp any any -¿any 53 (msg:”Snapchat DNS TFM GUADALUPEBV ”; byte-test:1,!,0xF8,2; content:”—08—snapchat—03—com—00—”; fast-pattern: only; sid:1004; rev:1;)

alert udp any any -¿any 53 (msg:”Tumblr DNS TFM GUADALUPEBV ”; byte-test:1,!,0xF8,2; content:”—06—tumblr—03—com—00—”; fast-pattern: only; sid:1005; rev:1;)

alert udp any any -¿any 53 (msg:”Telegram DNS TFM GUADALUPEBV ”; byte-test:1,!,0xF8,2; content:”—03—web—08—telegram—03—org—00—”; fast-pattern: only; sid:1006; rev:1;)

————————————————-PROXYS————————————————

VirtualShield alert udp any any -¿any 53 (msg:”TFM GuadalupeBV entrada a proxy VirtualShield alerta por DNS”; byte-test:1,!,0xF8,2; content:”—13—virtualshield—03—com—00—”; fast-pattern: only; sid:1011; rev:1;)

Nord.VPN alert udp any any -¿any 53 (msg:”TFM GuadalupeBV entrada a proxy Nord alerta por DNS”; byte-test:1,!,0xF8,2; content:”—07—nordvpn—03—com—00—”; fast-pattern: only; sid:1012; rev:1;)

Shadowsocks alert udp any any -¿any 53 (msg:”TFM GuadalupeBV entrada a proxy Shadowsocks alerta por DNS”; byte-test:1,!,0xF8,2; content:”—11—shadowsocks—03—com—00—”; fast-pattern: only; sid:1013; rev:1;)

Apéndice B

Anexo2. Crear certificados con easy-rsa 2.0

Para poder emitir y revocar la claves necesitamos crear nuestra propia autoridad certificadora y disponer de nuestro certificado raíz ca.ctr y de nuestra clave ca.key para poder crear y firmar las claves de los clientes y del servidor.

Para realizar este paso, y el resto de pasos, ejecutoriaremos los scripts que OpenVPN trae incorporados de serie. Para ello tenemos que crear una carpeta con nombre easy-rsa dentro de la ubicación /etc/openvpn. Para ello abrimos una terminal y tecleamos el siguiente comando:

cd /etc/openvpn

mkdir easy-rsa

Seguidamente tenemos que copiar los scripts de configuración de OpenVPN, que se hallan en la ubicación /usr/share/doc/openvpn/examples/easy-rsa/2.0/, dentro de la carpeta easy-rsa que acabamos de crear. Para ello en la terminal tecleamos el siguiente comando:

cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/* easy-rsa

En el caso que que vuestra distro trabaje con la versión 3 de easy-rsa, en el momento de introducir el último comando, obtendréis un error parecido al siguiente error:

cp: no se puede efectuar ‘stat’ sobre «/usr/share/doc/openvpn/examples/easy-rsa/2.0/*»: No existe el archivo o el directorio.

Los pasos a realizar para solucionar este error son los siguientes. En la terminal escriben el siguiente comando para instalar el paquete easy-rsa.

apt-get install easy-rsa

Seguidamente borran la carpeta easy-rsa que habíamos creado inicialmente introduciendo el siguiente comando en la terminal:

rm -R /etc/openvpn/easy-rsa

Finalmente para obtener los scripts para la creación de claves en la terminal introducimos el siguiente comando:

make-cadir /etc/openvpn/easy-rsa

Para ejecutar los scripts que acabamos de copiar o de obtener, tenemos que ir a la ubicación donde los guardamos. Para ello ingresamos el siguiente comando en la terminal:

cd /etc/openvpn/easy-rsa

Antes de ejecutar los scripts editaremos el fichero vars para modificar una serie de parámetros. Para modificar el fichero vars se tiene que introducir el siguiente comando en la terminal:

nano vars

Tamaño de las claves.

Una vez abierto el editor de texto tenemos que localizar y modificar la siguiente línea:

export-KEY-SIZE=1024

Una vez encontrada la sustituyen por la siguiente linea:

export-KEY-SIZE=2048

Nota: Con esta modificación estamos incrementando el tamaño de la claves privadas (.key) que vamos a generar y también del parámetro de Diffie Hellman. Con esta modificación incrementarios del tamaño de las claves de 1024 bits a 2048 bits. También seria posible usar 4096 bits. Este parámetro no tiene porqué penalizar en

exceso el rendimiento del servidor. Únicamente penalizará el proceso autentificación Handshake de SSL/TLS.

Datos de la entidad emisora de los certificados

Seguidamente tenemos que introducir los datos de la entidad emisora de los certificados que seremos nosotros mismos Para ello tenemos que localizar las siguientes lineas:

export KEY-COUNTRY=”US”

export KEY-PROVINCE=”CA”

export KEY-CITY=”SanFrancisco”

export KEY-ORG=”Fort-Funston”

export KEY-EMAIL=”me(arroba)myhost.mydomain”

export KEY-EMAIL=mail(arroba)host.domain

export KEY-CN=Changeme

export KEY-CN=Changeme

export KEY-OU=Changeme

Una vez localizadas las lineas tan solo se tienen reemplezar el contenido por defecto por nuestros datos reales. En mi caso los datos a rellenar podrían ser:

export KEY-COUNTRY=”ES” “Poner las 2 iniciales de tu país”

export KEY-PROVINCE=”CA” “Poner las 2 iniciales de tu provincia”

export KEY-CITY=”s*******a” “Poner el nombre de tu ciudad”

export KEY-ORG=”geekland” “Poner el nombre de la organización”

export KEY-EMAIL=”xxxxxxx(arroba)gmail.com” “Usar vuestra dirección de email”

export KEY-EMAIL=xxxxxxx(arroba)gmail.com “Usar vuestra dirección de email”

export KEY-CN= wheezy “Usar el nombre del host del servidor”

export KEY-NAME=vpnkey “Designa el nombre de la entidad certificadora que se creará”

export KEY-OI=IT “Departamento de la empresa”

Nota: Dentro de este fichero también podemos configurar el tiempo de validez que tendrá nuestra entidad certificadora y el tiempo de validez que tendrán los certificados y claves que crearemos. El valor estándar de validez son 3650 días que no voy a tocar.

Una vez modificado el archivo vars guardamos los cambios y lo cerramos. Ahora tendremos que exportar sus variables. Para exportar sus variables tenemos que teclear el siguiente comando en la terminal:

source ./vars

Seguidamente ejecutaremos el script clean-all. El script clean-all borrará la totalidad de claves que podrían existir en la ubicación /etc/openvpn/easy-rsa/keys. Para ejecutar el script tenemos que teclear el siguiente comando en la terminal:

./clean-all

El siguiente paso es generar los parámetros de Diffie Hellman. Los parámetros de Diffie Hellman se utilizarán para poder intercambiar las claves ente cliente y servidor de forma segura. Para poder realizar este paso tenemos que teclear el siguiente comando en la terminal:

./build-dh

Al terminar el proceso dentro de la ubicación /etc/openvpn/easy-rsa/keys se habrá creado el archivo dh2048.pem que contiene los parámetros Diffie Hellman.

Nota: Para quien requiera información adicional de los parámetros de Diffie Hellman puede consultar el siguiente enlace. Este parámetro se usará poder un intercambio de claves entre 2 participantes de forma segura.

En la siguiente captura de pantalla podrán ver una muestra de los pasos realizados hasta el momento:

Finalmente vamos a a crear el certificado y la clave privada de nuestra propia autoridad certificadora. Para ello tenemos que teclear el siguiente comando en la terminal:

./build-ca

Durante el proceso de creación se les hará una serie de preguntas para incorporar información dentro del certificado que se creará. Como anteriormente hemos editado

el fichero vars ahora solo nos tenemos que limitar a aceptar el valor por defecto de las preguntas que nos hacen.

Al terminar el proceso dentro de la ubicación /etc/openvpn/easy-rsa/keys se ha creado ca.crt y ca.key:

ca.crt: Es el certificado raíz público de la autoridad de certificación (CA)

ca.key: Este fichero contiene la clave privada de la autoridad de certificación (CA). Este archivo debe mantenerse protegido y no debe estar al alcance de terceros.

CREAR EL CERTIFICADOS Y LA CLAVE DEL SERVIDOR OPENVPN

A estas alturas ya lo tenemos todo listo para poder crear el certificado y clave de nuestro servidor. Para ello introducimos el siguiente comando en la terminal:

./build-key-server server

Una vez introducido este comando se nos hará una serie de preguntas. Simplemente tienen que contestar el valor por defecto ya que anteriormente hemos modificado el archivo vars. Al terminar el proceso dentro de la ubicación /etc/openvpn/easy-rsa/keys se habrán creado los siguientes archivos:

whezzyVPN.key: Este fichero contiene la clave privada del servidor. Este archivo no debe estar al alcance de nadie.

whezzyVPN.crt: Este fichero corresponde al certificado público del servidor.

whezzyVPN.csr: Este archivo es la petición de certificado que se envía a la autoridad de certificación. Mediante la información que contiene el archivo .csr, la autoridad de certificación podrá realizar el certificado del servidor una vez hayan realizado las comprobaciones de seguridad pertinentes.

Apéndice C

Bibliografía

1 https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos /2835-ccn-cert-ia-09-18-ciberamenzas-y-tendencias-edicion-2018-1/file.htm

2 Exploding the pone, Author: Phil Lapsley Editor: Barnes Noble.

3 http://www.editorialuoc.cat/siete-fracasos-que-han-cambiado-el-mundo-1

4 https://history-computer.com/Internet/Maturing/Thomas.html

5 https://www.abc.es/tecnologia/redes/abci-hackeos-mas-sonados-2018-saldan-alrededor-65-millones-victimas-mundo-201810031947-noticia.html

6 https://www.statista.com/statistics/494961/web-attacks-blocked-per-day-worldwide/

7 https://www.lavanguardia.com/tecnologia/20181003/452150484515/facebook-hackeo-seguridad-ue-redes-sociales-tecnologia-portada.html

8 https://iabspain.es/wp-content/uploads/estudio-redes-sociales-2018-vreducida.pdf

9 https://elpais.com/diario/2011/11/20/negocio/1321800447-850215.html

10 https://www.independent.co.uk/life-style/gadgets-and-tech/news/digmine-facebook -messenger-cryptocurrency-mining-malware-monero-bitcoin-a8125021.html

11 https://nvlpubs.nist.gov/nistpubs/ir/2013/nist.ir.7298r2.pdf

12 http://www.channelpartner.es/seguridad/noticias/1107063002502/gasto-mundial-de-seguridad-informatica-crecera-12-ano.1.html

13 https://www.researchgate.net/publication/242363259-Building-a-Secure-Computer-System

14 Procedings ti conferences, Approaches To Privacy and Security in Computer Systems, Departamento de comercio de U.S., 1974, pp 55

15 https://www.universidadviu.es/conceptos-seguridad-logica-informatica/

16 https://news.sophos.com/es-es/2012/09/14/seguridad-activa-y-seguridad-pasiva-en-equipos-informaticos/

17 https://news.sophos.com/es-es/2012/09/14/seguridad-activa-y-seguridad-pasiva-en-equipos-informaticos/

18 https://ruc.udc.es/dspace/bitstream/handle/2183/13116/CC-116-art-6.pdf

19 https://www.universidadviu.es/conceptos-seguridad-logica-informatica/

20 https://iiemd.com/malware/que-es-malware

21 https://ws680.nist.gov/publication/get-pdf.cfm?pub-id=50951, pag 15

22 https://www.incibe-cert.es

23 https://searchsecurity.techtarget.com/definition/intrusion-detection-system

24 https://es.wikipedia.org/wiki/Sistema-de-deteccion-de-intrusos

25 https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi-diseno-configuracion-ips-ids-siem-en-sci.pdf

26 https://www.ecured.cu/IDS

27 http://www.cs.colostate.edu/ cs656/reading/ieee-se-13-2.pdf

28 James P. Anderson: An Information Security Pioneer February 2008 IEEE Security and Privacy Magazine 6(1):9-9 DOI: 10.1109/MSP.2008.15 Source IEEE Xplore

29 https://pdfs.semanticscholar.org/5653/f38f614db4f51905e7146ec1147d41f8c6ba.pdf

30 https://static1.squarespace.com/static/510d93d8e4b060f86e6fdf2d/t /5b3a66282b6a28753522d374/1530553916538/DIDS-demo-1991.pdf

31 https://www.researchgate.net/publication /255061208-NADIR-Network-Anomaly-Detection-ar

32 http://rediris.es/cert/doc/pdf/ids-uv.pdf

33 https://www.incibe-cert.es/

34 https://www.securityartwork.es/2017/01/27/actualizacion-automatica-reglas-snort-pulledpork/

35 https://educacionadistancia.juntadeandalucia.es/profesorado/pluginfile.php/84356

36 https://www.incibe-cert.es

37 https://blog.securityonion.net/

38 https://www.unleashnetworks.com/blog/?p=322

39 https://trisul.org/download/

40 https://trisul.org