UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de los Computadores

Trabajo Fin de Grado

Desarrollo de Sistema Centralizado de recolección y correlación

de eventos de red enviadas desde sondas en Opensource

Carlos Carvajal Molinero

Noviembre 2018

UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de los Computadores

Trabajo Fin de Grado

Desarrollo de Sistema Centralizado de recolección y correlación de eventos de red enviadas desde sondas en Opensource

Autor: Carlos Carvajal Molinero

Tutor: Rafael Martin Espada

Agradecimientos

A mi padre, a mi madre y a mi hermano

por estar siempre apoyándome y darme

ánimos para lograr terminar el Grado.

A mis compañeros de clase y amigos que han

estado a las duras y a las maduras

desde siempre.

A Ariadnex por permitirme trabajar y aprender

de ellos todo lo posible para lograrlo, tanto a

Juan Miguel Trejo como a Juan Luis Hoyo.

A Rafael Martin por darme la oportunidad

de realizar este proyecto.

Especial mención a David Romero por ayudarme, enseñarme

y estar pendiente de mí en todo el recorrido de la arquitectura.

ÍNDICE GENERAL DE CONTENIDOS

ÍNDICE DE FIGURAS

2. Summary

As progress is made in the development of data networks, it is becoming more difficult to defend ourselves against third-party attacks, a fact that is mainly pressing in the case of companies that need a high degree of privacy in their data. Day by day new methods of attacks appear for different purposes, but in any case they damage the operation, reputation or compromise the company in a very serious way.

As technology specialists we see the need to develop an architecture of integrated tools that help to make the information that is in the power of the organization as accessible and safe as possible.

Thanks to the configuration of elements that arise in this project we will be able to visualize and manage all the information of the events that occur in the clients and thus be able to establish the necessary defenses with the maximum possible speed. Our final goal must be to have a good level of security, both in our server, for the vital role it plays and which can be the object of attacks, as well as in our clients, the main objective of our organization, and ensure the whole environment against possible attacks by third parties.

2. Resumen

A medida que se avanza en el desarrollo de las redes de datos, se va haciendo más difícil defendernos de los ataques de terceros, hecho principalmente acuciante en el caso de empresas que necesitan una alta privacidad en sus datos. Día a día van apareciendo nuevos métodos de ataques con fines diversos, pero en todo caso dañan la operatividad, reputación o comprometen a la empresa de manera muy seria.

Como especialistas en tecnología nos vemos en la necesidad de desarrollar una arquitectura de herramientas integradas que ayuden a conseguir que la información que obra en poder de la organización sea lo más accesible y segura posible.

Gracias a la configuración de elementos que se plantean en este proyecto seremos capaces de visualizar y gestionar toda la información de los eventos que se producen en los clientes y así poder establecer las defensas necesarias con la máxima celeridad posible. Nuestro objetivo final debe ser el de disponer de un buen nivel de seguridad, tanto en nuestro servidor, por la función vital que desempeña y que puede ser objeto a su vez de ataques, como en nuestros clientes, objetivo principal de nuestra organización, y asegurar todo el entorno contra posibles ataques de terceros.

3. Introducción

Se entiende por seguridad de la información al conjunto de normas, procedimientos y herramientas que tienen como objetivo garantizar la disponibilidad, integridad, confidencialidad y buen uso de la información que reside en un sistema de información. Se diferencia de la Seguridad Informática o Ciberseguridad en que esta última se reduce a la protección de la infraestructura computacional y todo lo relacionado con esta y, especialmente, la información contenida en una computadora o circulante a través de las redes de computadoras [1].

Cada día más y más personas malintencionadas intentan tener acceso a los datos de nuestros ordenadores y eso se refleja particularmente en el número de vulnerabilidades detectadas en los sistemas.

A modo de Ejemplo, durante 2017, las vulnerabilidades en los sistemas de información han sobrepasado por mucho los registros de años previos. De acuerdo con CVE Details, en el año citado se han reportado más de 14.700 vulnerabilidades en los sistemas (hardware y software), contra las cerca de 6447 que se informaron en 2016. Por tanto el crecimiento se ha más que duplicado en un solo año, esto representa un aumento superior del 120% de un año a otro [2].

Ilustración 1 Vulnerabilidades reportadas por año

Viendo los resultados de estos registros, cabe destacar también que, en promedio, durante 2017 se reportaron 40 vulnerabilidades a diario, contra el promedio de 17 vulnerabilidades registradas por día durante 2016.

Una de las posibles consecuencias de una intrusión es la pérdida de datos. Es un hecho frecuente y ocasiona muchos trastornos, sobre todo si realizamos diariamente copias de seguridad. Y aunque se realizasen, no siempre es posible recuperar la totalidad de los datos pues podrían contener errores.

La vulnerabilidad tan solo representa un factor de la ecuación que mide los riesgos a los que se enfrenta una organización o un particular. En efecto, los riesgos, en términos de seguridad, se caracterizan por lo general mediante la siguiente ecuación:

$$Riesgo = \frac{Amenaza \times Vulnerabilidad}{Contramedida}$$

La **amenaza** representa el tipo de acción que tiende a ser dañina, mientras que la **vulnerabilidad** representa el grado de exposición a las amenazas en un contexto particular. Finalmente, la **contramedida** representa todas las acciones que se implementan para prevenir la amenaza.

Las contramedidas que deben implementarse no solo son soluciones técnicas, sino que también reflejan la capacitación y la toma en consecuencia por parte del usuario, además de establecer unas reglas claramente definidas en términos de procesos y organización.

Para que un sistema sea seguro, hasta donde el término pueda establecerse en un caso concreto, deben identificarse las posibles amenazas y por lo tanto, conocer y prever el curso de acción del agente malicioso. Por tanto, el objetivo de este proyecto es brindar una perspectiva general de las posibles motivaciones de los hackers, categorizarlas y ofrecer una perspectiva de cómo funcionan dichos ataques para conocer la mejor forma de reducir el riesgo de intrusiones.

Y es en este ámbito, el del desarrollo de herramientas y configuraciones para que nuestros sistemas sean seguros, aquellas que se encargan de detectar los posibles intrusos las que se examinaran a lo largo del proyecto. Esas herramientas, denominadas IDS (Intrusion Detection Systems o Sistemas de Detección de Intrusos) no son más que programas de detección de accesos no autorizados ya sea a organizaciones, clientes o en la misma red de una empresa de seguridad y todo ello a través de Open Source.

Los IDS se basan en sistemas que están monitoreando los eventos que ocurren en dicha red, para poder así interpretarlos y gestionarlos de la mejor forma posible en un servidor de correlación y consolidación de eventos, teniendo en cuenta que nuestros clientes podrían estar localizados en cualquier parte del mundo. Se les deberá ofrecer el mayor grado de seguridad posible, de forma que se les pueda ofrecer un entorno de red fiable y confiable.

Múltiples empresas de seguridad y redes recomiendan la instalación de sistemas IDS en las redes. Paradigmático es el caso de CISCO que lo propone como la mejor forma de proteger nuestros datos a posibles ataques que puedan surgir y poder así asegurarlos. [3]

4. Objetivos

Generalmente, en todos los sistemas de información se incluyen algunos datos privados de una compañía, material y los recursos software, que permitirían a una empresa almacenar y hacer circular estos datos por espacios indebidos. La seguridad informática consiste en garantizar que el material y los recursos de software de una organización se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto.

Es por ello que queremos realizar una arquitectura de sistemas para poder monitorizar los eventos que se producen en las redes de los clientes y así poder analizarlos y gestionarlos, en caso de ser necesarios y ofrecer un servicio de seguridad contra ataques de terceros. Dicha arquitectura constará de un servidor central el cual recibirá la información de los eventos acaecidos en la infraestructura tecnológica de los clientes (redes y sistemas) que se encuentren en cualquier parte del mundo.

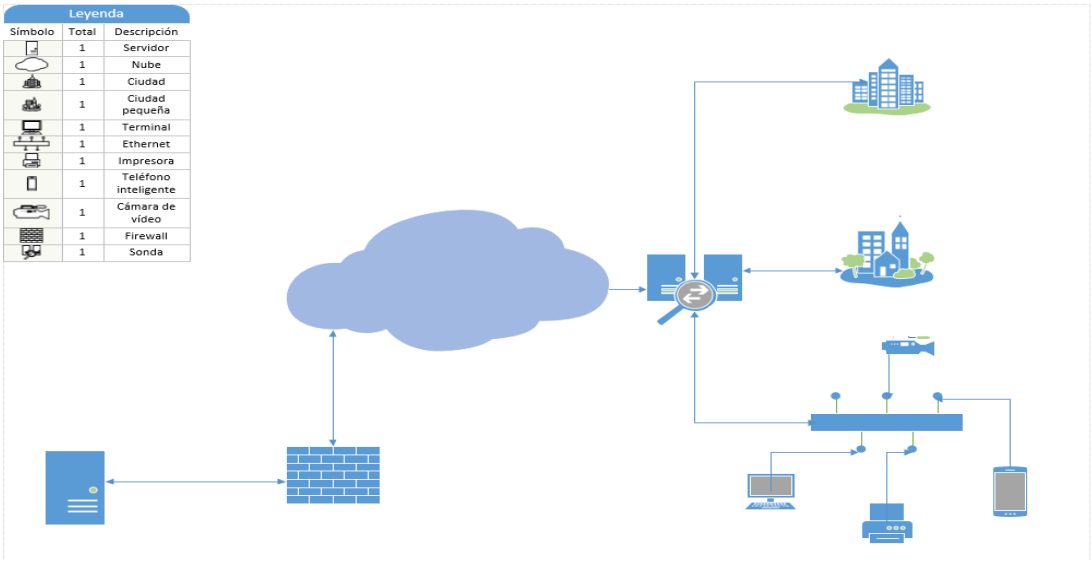


Ilustración 2 Esquema general

Los eventos que queremos recibir podrían pertenecer a las redes de una ciudad grande o pequeña, desde una red en la que se conecten múltiples equipos desde un cliente hasta una con Smartphone's. Los datos de los eventos se transmitirán a través de un túnel establecido a través de los elementos de Internet, lo que se representa de forma clásica por la nube,

para que los datos fluyan cifrados y así alcanzar el servidor de forma segura. Una vez en el servidor, estos datos se analizan para que este pueda gestionarse y ofrecer las condiciones de mantener un nivel de seguridad óptimo.

4.2 Tecnología empleada

En los siguientes apartados constará de una introducción de las tecnologías y herramientas que se utilizarán para realizar con éxito la arquitectura deseada y con ello poder detectar eventos de seguridad para visualizarlos y gestionarlos.

4.2.1 Port Mirroring o puerto espejo

Cuando tenemos una red profesional y existe algún problema, lo más fácil y rápido para poder solucionarlo es reenviar todo el tráfico de los diferentes puertos a un único puerto, y en dicho puerto conectar un ordenador con algún analizador de paquetes para posteriormente analizarlo en detalle de manera offline.

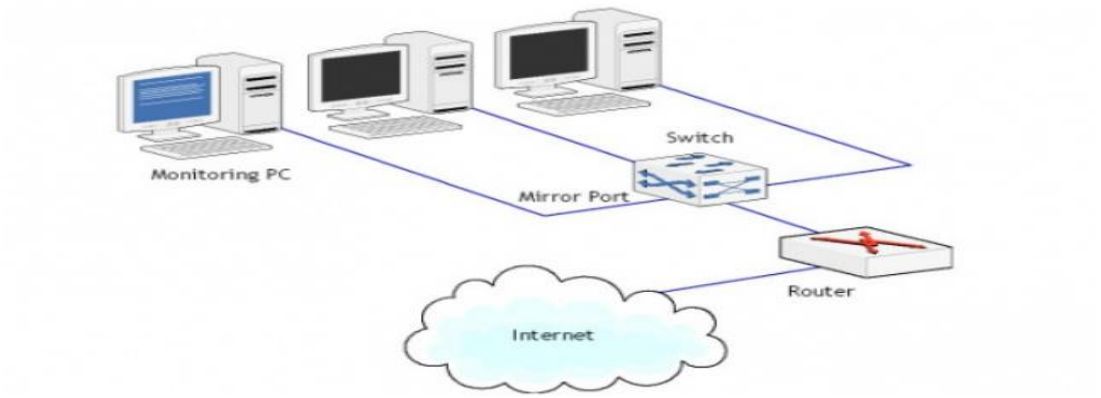


Ilustración 3 Esquema Port Mirroring

Port Mirroring es la característica de los switches que nos permite enviar una copia de todas las tramas enviadas y recibidas por el switch a un determinado puerto del mismo, ideal para investigar qué está ocurriendo con el tráfico. [4]

4.2.2 Rsyslog

Syslog es un estándar utilizado para la captura, el procesamiento y el trasporte de mensajes de registro del sistema —es decir las bitácoras del sistema. Es tanto un protocolo de red como a la aplicación o biblioteca compartida que sirve para procesar y enviar los mensajes de registro del sistema.

Los mensajes se etiquetan con un código de identificación de entre los siguientes: *auth*, *authpriv*, *daemon*, *cron*, *ftp*, *lpr*, *kern*, *mail*, *news*, *syslog*, *user*, *uucp* y *local0* hasta *local7*. La etiqueta también incluye el tipo de programa que generó los mensajes, indicando también el nivel de severidad de entre los siguientes: *Emergency*, *Alert*, *Critical*, *Error*, *Warning*, *Notice*, *Info* y *Debug*.

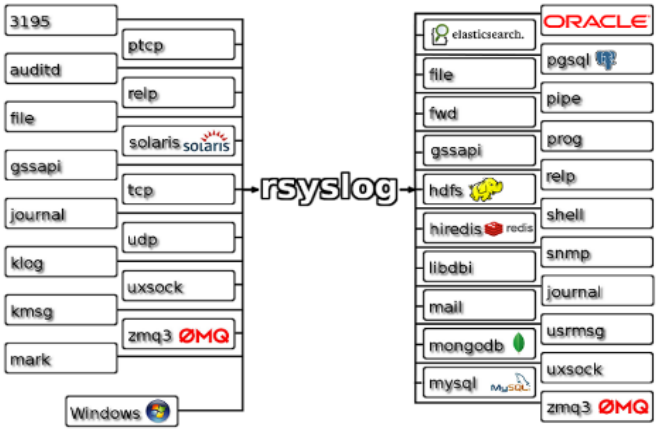


Ilustración 4 Rsyslog [5]

Rsyslog es un eficiente y rápido sistema de procesamiento de registros de sistema. Ofrece un diseño modular de alto desempeño y niveles de seguridad apropiados. A diferencia de sus predecesores —*sysklog* y *syslog*— permite ingreso de datos desde diversas fuentes, trasformación de datos y salida de resultados hacia varios destinos. Es lo suficientemente versátil y robusto para ser utilizado en entornos empresariales y tan ligero y sencillo que permite utilizarlo también en sistemas pequeños. Permite almacenar las bitácoras en archivos de texto simple o bases de datos MySQL y PostgreSQL, utilizar otros destinos en

caso de falla, transporte de *syslog* a través de tcp, control detallado de formatos, etiquetas de tiempo exactas, operaciones en cola de procesamiento y capacidades de filtrado en cualquier parte de los mensajes.[6]

4.2.3 VPN y openVPN

Una **VPN** o **Red Privada Virtual** es una tecnología que permite la extensión de una red pública como Internet a un espacio de red local. Una VPN es una red virtual que se crea dentro de otra red, como por ejemplo Internet. [7]

En la informática una Red Privada Virtual (RPV) o Virtual Private Network (VPN) supone una tecnología de red que, por razones de costo y comodidad, brinda la posibilidad de conectarse a una red pública generando una extensión a nivel de área local. Por caso, este tipo de redes se utilizan a la hora de conectar dos o más oficinas de una empresa a través de Internet. Esto facilita la conexión y el intercambio a un bajo costo económico, y permite que miembros de un mismo equipo se conecten entre sí desde locaciones remotas.

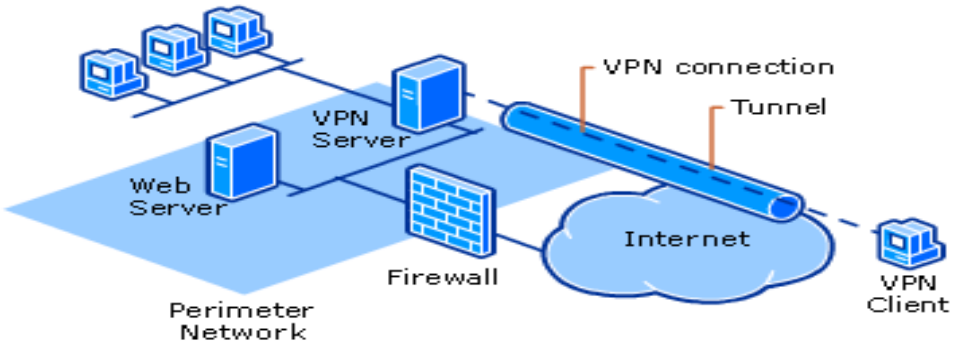


Ilustración 5 Esquema OpenVPN

Las VPN funcionan de manera tal que, si bien se utiliza una red pública como es la de conexión a Internet, los datos son transmitidos por un canal privado, de forma que no peligra la seguridad ni la integridad de la

información interna. Los datos son cifrados y descifrados alternativamente, ahorrando dinero y problemas a empresas de distinta escala.

OpenVPN es una solución para VPN que implementa conexiones de capa 2 o 3, usa los estándares de la industria SSL/TLS para cifrar y combina todas las características mencionadas anteriormente en las otras soluciones VPN. Su principal desventaja por el momento es que hay muy pocos fabricantes de hardware que lo integren en sus soluciones. Sin embargo, en sistemas basados en Linux se puede implementar sin problemas mediante software. [8]

4.2.4 SSH / PuTTy

SSH™ (o *Se*cure *Sh*ell) 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 encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. Por defecto se utiliza el puerto 22. [9]

SSH trabaja de forma similar a como se hace con telnet. La diferencia principal es que SSH usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible, evitando que terceras personas puedan descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión; aunque es posible atacar este tipo de sistemas por medio de ataques de REPLAY y manipular así la información entre destinos.

Ilustración 6 Esquema ssh y puerto utilizado

En cuanto al **PuTTy** es un cliente SSH, Telnet, rlogin, y TCP raw con licencia libre. Disponible originalmente sólo para Windows, ahora también está disponible en varias plataformas Unix, y se está desarrollando la versión para Mac OS clásico y Mac OS X. Otra gente ha contribuido con versiones no oficiales para otras plataformas, tales como Symbian para teléfonos móviles. [10]

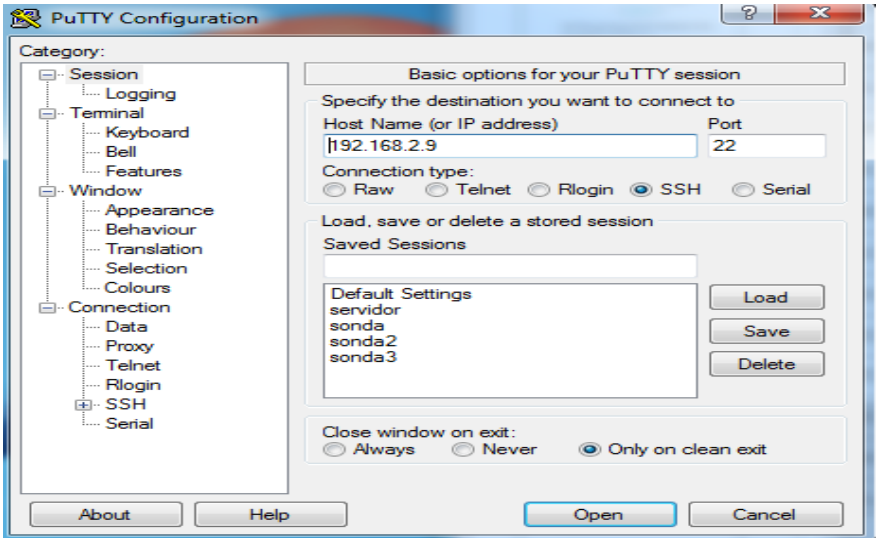


Ilustración 7 Ventana Putty

4.2.5 Elasticsearch

Elasticsearch es un motor de búsqueda y análisis RESTful distribuido de código abierto capaz de resolver un número creciente de casos de uso.

Elasticsearch es un servidor de búsqueda basado en Lucene. Provee un motor de búsqueda de texto completo, distribuido y con capacidad de multi-tenencia con una interfaz web RESTful y con documentos JSON. Elasticsearch está desarrollado en Java y está publicado como código abierto bajo las condiciones de la licencia Apache. [11]

Es una base de datos distribuida. Distribuye toda la información en todos los nodos, por tanto es tolerante a fallos y tiene alta disponibilidad. Al igual que distribuye la información, distribuye el procesamiento. Cuando se realiza una consulta o búsqueda y esa información se encuentra distribuida, será

cada nodo el que procese dicha información y devuelva los resultados. Por tanto, podemos obtener mejores rendimientos.

Lo que se consigue con ELK es coger toda esta información, procesarla y almacenarla de forma distribuida. Así se va a poder escalar en Big Data y obtener buenos rendimientos con grandes cantidades de información, y transformarla en visualizaciones, con las que poder identificar anomalías, comportamientos, eventos, picos, etc., de forma gráfica y visual, como en la diapositiva mostrada. [12]

4.2.6 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 gráficos de barra, línea y dispersión, o gráficos circulares y mapas además de grandes volúmenes de datos.

Es la parte de preprocesamiento antes de guardar la información en Elasticsearch que hemos comentado, donde recogemos un input, una entrada, trabajamos los eventos y los sacamos por una salida, antes de almacenarlos en las bases de datos.[13]

Ilustración 8 Ventana Kibana

4.2.7 Logstash

Es la parte de preprocesamiento antes de guardar la información en Elasticsearch que hemos comentado, donde recogemos un input, una entrada, trabajamos los eventos y los sacamos por una salida, antes de almacenarlos en las bases de datos.

Logstash es una herramienta para recopilar, procesar y reenviar eventos y registrar mensajes. La recopilación se lleva a cabo a través de complementos de entrada configurables, incluida la comunicación bruta de socket / paquete, seguimiento de archivos y varios clientes de bus de mensajes. Una vez que un complemento de entrada ha recolectado datos, puede ser procesado por cualquier cantidad de filtros que modifiquen y anoten los datos del evento. Finalmente **Logstash** enruta los eventos a los complementos de salida que pueden reenviar los eventos a una variedad de programas externos, incluidos Elasticsearch, archivos locales y varias implementaciones de bus de mensajes [14]

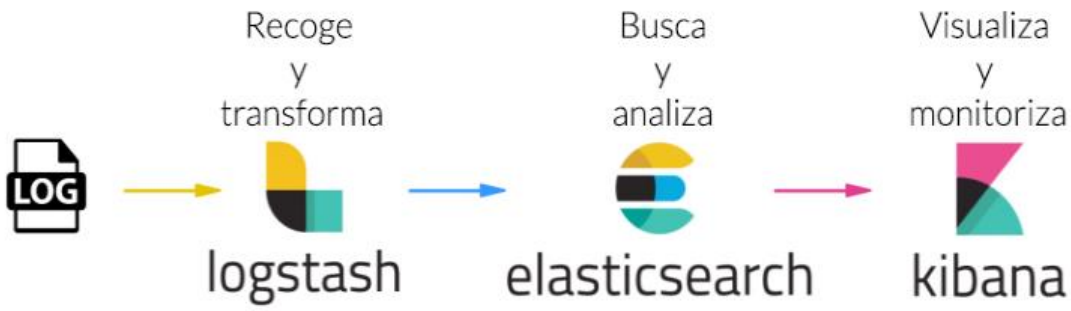


Ilustración 9. Tratamiento Logstash, Elasticsearch y kibana

4.2.8 Sistemas de detección de intrusos

En este apartado veremos los diferentes IDS (Intrusion Detection Systems o Sistemas de Detección de Intrusos) que hemos visto para lograr montar la arquitectura. Los que hemos destacado son: **Snort, Suricata y Bro.**

4.2.8.1 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. 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 los principales fabricantes de IDS/IPS lo utilizan, pudiendo utilizarse sus firmas en casi cualquier dispositivo. [15]

4.2.8.2 Suricata

Suricata es el nombre de un proyecto de software libre, desarrollado por la comunidad OISF (Open Information Security Foundation). Es un motor 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. [16]

Suricata puede funcionar como IDS de tiempo real, IPS, 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 el lenguaje LUA. Dispone de entradas y salidas estandarizadas a formatos 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 y otros recursos importantes en la industria para proporcionar las reglas más actualizadas y completas disponibles.

4.2.8.3 Bro

Bro es otra herramienta que sirve de 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. [17]

Al igual que otras herramientas, Bro también hace uso de la librería libpcap para su funcionamiento, además es capaz de funcionar en varias redes. 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. Bro está conceptualizado en dos capas: ν Motor de eventos: Analiza y guarda el tráfico de la red para generar eventos neutros, que se basan en inicios o paradas de transmisiones, detección de puertos y protocolos. ν Política de programa: Analiza los acontecimientos para crear políticas de acción. Bro registra los eventos, pero también puede ser configurado para tomar acciones como el envío de alertas, ejecuciones de comandos, actualizaciones y llamadas a otros programas. [18]

Diferencias entre **Bro, Snort** y **Suricata**.

Parámetros	Bro	Snort	Suricata
Multi-hilos	No	V3.0	Si
Soporte para ipv6	Si	Si	Si

Reputación IP	Parte	No	Si
Detección automática de protocolos	Si	V3.0	Si
Aceleración con GPU	No	No	Si
Variables Globales/flowbits	Si	No	Si
GeoIP	Si	No	Si
Análisis avanzado de http	Si	No	Si
HTTP Access Logging	Si	No	Si
SMB Access Logging	Si	No	Si
gratis	Si	Si	Si

Como se muestra en la tabla, parece que el mejor IDS es **Suricata** pues muestra todas las características de los demás con más mejoras como por ejemplo que sea multi-hilos y que tenga aceleración con GPU.

Bro no se diferencia mucho de Suricata pero sí que se asemeja bastante pues tiene gran parte de sus características.

En cuanto a Snort es la herramienta con la que cuenta con menos características a diferencia de Bro y Suricata. Por lo que en el proyecto nos centraremos en los dos primeros.

4.2.9 Filebeat

Beats es la plataforma para los cargadores de datos de propósito único. Se instalan como agentes livianos y envían datos de cientos o miles de máquinas a Logstash o Elasticsearch.

Beats puede enviar datos directamente a Elasticsearch o a través de Logstash, donde puede procesar y mejorar aún más los datos, antes de visualizarlos en Kibana.

Filebeat se usa en todos los servidores para recopilar archivos de registro. Si bien tradicionalmente se ha empleado Syslog para esta tarea y para reenviar datos, existen algunos inconvenientes en Syslog. El más importante es que algunos servicios no usan Syslog sino que simplemente escriben en un archivo. Además, **Filebeat** es mejor para configurar el tráfico, si dirige mucho tráfico de rutina pero necesita reaccionar rápidamente a los errores.

4.2.10 Security Onion

Security Onion es una distribución de Linux de fuente abierta y gratuita para detección de intrusiones, monitoreo de seguridad empresarial y administración de registros. Incluye Elasticsearch, Logstash, Kibana, Snort, Suricata, Bro, OSSEC, Sguil, Squert, NetworkMiner y muchas otras herramientas de seguridad. El asistente de configuración fácil de usar le permite construir un ejército de sensores distribuidos para su empresa en cuestión de minutos. [19]

La distribución cuenta también con un asistente muy sencillo que nos va a permitir, en cuestión de minutos, montar toda una red de sensores capaces de detectar prácticamente cualquier amenaza presente en cualquiera de los hosts de una red.

Ilustración 10 Inicio Security Onion

4.2.11 Raspberry pi

Raspberry PI es una placa computadora (SBC) de bajo coste, se podría decir que es un ordenador de tamaño reducido, del orden de una tarjeta de crédito, desarrollado en el Reino Unido por la Fundación Raspberry PI (Universidad de Cambridge) en 2011, con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012. [20]

Ilustración 11 Raspberry PI 3 Model B

El concepto es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico. Está formada por

una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal.

A la **raspberry Pi** la han definido como una maravilla en miniatura, que guarda en su interior un importante poder de cómputo en un tamaño muy reducido. Es capaz de realizar cosas extraordinarias.

4.3 Solución de propuesta

En este apartado nos introduciremos en hablar de los Sistemas Operativos que usaremos y de poner en detalle cómo será la estructura así como que protocolos de comunicación se utilizaran y que sistema de recolección de eventos realizaremos para dar así la seguridad que deseamos.

4.3.1 Sistemas operativos

- Debían

El Proyecto Debían es una asociación de personas que han hecho causa común para crear un sistema operativo (SO) libre. Fue fundado en el año 1993 por Ian Murdock.

Un sistema operativo es un conjunto de programas y utilidades básicas que hacen que su computadora funcione. El centro de un sistema operativo es el núcleo (N. del T.: kernel). El núcleo es el programa más importante en la computadora, realiza todo el trabajo básico y le permite ejecutar otros programas.

Los sistemas Debían actualmente usan el núcleo de Linux o de FreeBSD. Linux es una pieza de software creada en un principio por Linux Torvalds y desarrollada por miles de programadores a lo largo del mundo. FreeBSD es un sistema operativo que incluye un núcleo y otro software. [21]

- Raspbian

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debían (Debían 9.4) para la placa computadora (SBC) Raspberry Pi, orientado a la enseñanza de informática. El lanzamiento inicial fue en junio de 2012

Técnicamente el sistema operativo es un port no oficial de Debían armhf para el procesador (CPU) de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento en según qué casos. El port fue necesario al no haber versión Debían armhf para la CPU ARMv6 que contiene el Raspberry PI

Al ser una distribución de GNU/Linux las posibilidades son infinitas. Todo software de código abierto puede ser recompilado en la propia Raspberry Pi para arquitectura armhf que pueda ser utilizado en el propio dispositivo en caso de que el desarrollador no proporcione una versión ya compilada para esta arquitectura. Además esta distribución, como la mayoría, contiene repositorios donde el usuario puede descargar multitud de programas como si se tratase de una distribución de GNU/Linux para equipos de escritorio. Todo esto hace de Raspberry Pi un dispositivo que además de servir como placa con micro controlador clásica, tenga mucha de la funcionalidad de un ordenador personal. Lo que lo puede convertir en una alternativa a los ordenadores personales, especialmente para personas con pocos recursos, para la extensión de la informática en países subdesarrollados o para aplicaciones que no soliciten muchos requerimientos. [22]

- Security Onion

Security Onion es un Administrador de seguridad de red (NSM) Esta plataforma ofrece múltiples sistemas detección de intrusos (IDS) Incluyendo Host IDS (HIDS) y el IDS de red (NIDS). Muchos tipos de datos pueden ser adquiridos usando Security Onion para su análisis. Esto incluye los datos relativos a: host, red, Sesión, activos, alerta y protocolos. Puede ser implementado como un servidor independiente con el despliegue y el sensor incluido o con un servidor maestro y varios sensores permitiendo que el sistema sea reducido como sea necesario. Muchas interfaces y herramientas están disponibles para la gestión del sistema y el análisis de datos, tales como sguil, Snorby, Squert y Empresa de búsqueda de registros y Archivo (ELSA). Estas interfaces se pueden utilizar para el análisis de alertas y eventos capturados y luego se pueden exportar más para el análisis en red Herramientas análisis forense (NFAT), tales como NetworkMiner, CapME o Xplico.

También ofrece varios métodos de gestión tales como Secure Shell (SSH) para la gestión de servidor y cliente Web sensores y acceso remoto. Todo esto con la capacidad de reproducir y analizar el tráfico malicioso ejemplo hace que la adecuada una alternativa de bajo costo para la Gestión de la Seguridad en la red. [23]

Network Security Monitoring (NSM) es, en pocas palabras, monitoreando su red en busca de eventos relacionados con la seguridad. Puede ser proactivo, cuando se usa para identificar vulnerabilidades o certificados SSL que caducan, o puede ser reactivo, como en la respuesta a incidentes y análisis forense de la red. Ya sea que esté rastreando a un adversario o tratando de mantener a raya el malware, NSM proporciona contexto, inteligencia y conocimiento situacional de su red. Existen algunas soluciones comerciales que se acercan a lo que ofrece Security Onion, pero muy pocas contienen las enormes capacidades de Security Onion en un solo paquete. [24]

4.3.2 Esquema del sistema

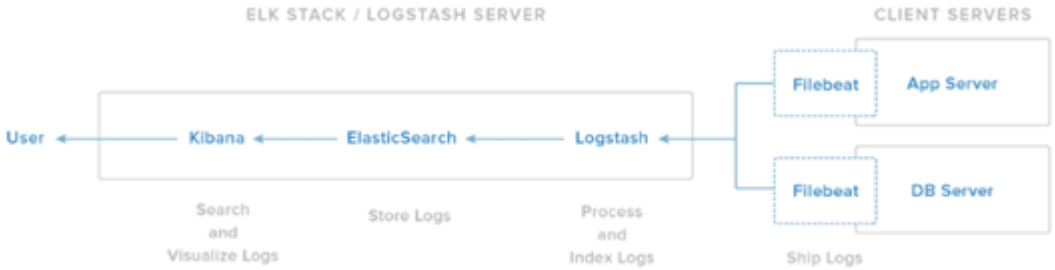


Ilustración 12 Esquema del Sistema Interno

4.3.3 Protocolo de comunicación

En este apartado explicaremos cómo estarán conectados tanto los clientes como el servidor para que así puedan comunicarse, y con ello puedan estar enviándose eventos de manera continuada.

Lo primero será lograr enviar la información de manera cifrada, con ello tendríamos una comunicación segura. Para realizarlo se utilizará una comunicación en modo túnel y lo conseguiremos a través de la configuración **OpenVPN**. Cuando todos los parámetros estén configurados, tendrán una arquitectura cliente/servidor desde una punta a otra en la que podrán lograr tener una comunicación “segura”.

En este punto tendremos la comunicación entre el cliente y el servidor, pero tendremos que configurar las demás herramientas para que logren pasarse los eventos obtenidos por parte de los clientes y poder visualizarlos en el servidor, eso se logra con Elasticsearch, kibana, Logstash y filebeat como se muestra en el punto anterior Esquema del sistema.

Lo primero es saber en qué sitio va cada herramienta explicada anteriormente, por ejemplo, por parte del cliente, lo que necesitamos es algo que nos recolecte los eventos y los mande a donde queramos, en nuestro caso nos referimos al servidor, que será el encargado de recibir y gestionar esos eventos que han sido mandado por parte del cliente y será gracias a la comunicación previa que habíamos conseguido al configurar **OpenVPN**.

4.3.4 Sistema de recolección de eventos

Una vez el sistema de comunicación que hemos explicado en el punto anterior, pasaremos a explicar en más detalle el punto 4.3.1 Esquema del sistema más en cuanto a la recolección de eventos.

La primera herramienta que tenemos que instalar en los clientes (sondas) para que nos esté recopilando la información es **Suricata**, que una vez configurado, será el encargado de recopilar todo tipo de eventos en nuestros clientes en el fichero eve.json y fast.json, de forma que se nos quede guardado dentro de nuestros clientes.

Una vez que se estén recopilando los eventos que queramos, necesitaremos de otra herramienta que sea el encargado de que toda esa información recolectada se envíe en nuestro caso al servidor. La herramienta encargada de poder enviar la información a través de las OpenVPN (sistema de comunicación que explicamos en el punto anterior) desde nuestros clientes al servidor se llama **Filebeat**.

En cuanto al sistema operativo **Security Onion** para realizar nuestra recolección de eventos utilizamos un IDS llamado **Bro** que será el encargado después de configurarlo que nos esté recolectando diversos tipos de eventos, ya sea desde una conexión http hasta una comunicación DNS o fallo de autenticación.

Como en los demás clientes, **Bro** es ayudado por **Filebeat** para el envido de los datos recolectados para enviarlo desde nuestra sonda hasta nuestro servidor a través del túnel “seguro” de manera cifrada.

4.3.5 Diseño del sistema

En cuanto al diseño del sistema explicaremos más en detalle de cómo se conectan todas las herramientas para lograr así que se registren todos los eventos que indiquemos y podamos gestionarlo lo más eficiente posible.

Como explicamos en el punto 4.3.3 Protocolo de comunicación, lo primero que se necesita es que el servidor y los clientes tengan una conexión segura que se ganaba gracias a **OpenVPN**.

Una vez esa comunicación, primero haremos que los clientes puedan registrar los eventos que tengan, esto en debían lo lograremos con la herramienta **Suricata** y en Security Onion será gracias a **Bro.**

Cuando ya estemos registrando los eventos, necesitaremos de otra herramienta para que logre enviar esos eventos al servidor a través de la conexión segura que ya teníamos, será la herramienta **Filebeat**.

Por parte del servidor lo primero que debemos realizar es analizar los eventos para poder visualizarlos lo más detalladamente posible, esto será gracias a la herramienta **Logstash** que a través de sus filtros podremos visualizarlos para luego guardarlos.

Para guardar dichos eventos para que después se puedan visualizar se necesitará la herramienta **Elasticsearch** que será el encargado de ir almacenando los eventos que se van registrando en los clientes. Se necesita de un template para que los datos se puedan guardar con los mismos campos después de que dichos eventos se parseen.

Una vez que los datos están almacenados en el servidor, la herramienta encargada de poder visualizarlo ser **Kibana**. Con este a través de la dirección que hayamos configurado podremos acceder a la visualización, con lo que podremos gestionarlos más cómodamente y lograr así realizar un Dashboard para realizar un resumen de los datos que necesitemos de forma gráfica.

5. Implementación

En este apartado nos disponemos a explicar cada una de las herramientas que hemos explicado en referencia a su instalación y a su debida configuración para que la arquitectura esté completa y pueda usarse satisfactoriamente.

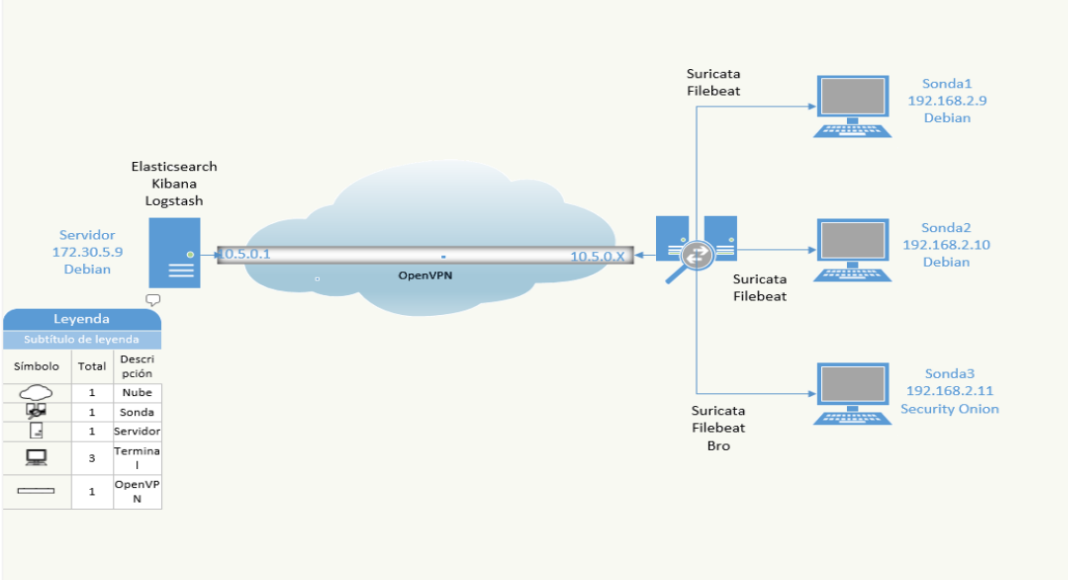


Ilustración 13 Esquema de la Implementación

5.1 Instalación y configuración de OpenVPN

Lo primero sería realizar una conexión segura entre el cliente y el servidor, para ello utilizaremos tantas VPN's como clientes tengamos. Esta configuración vale para los clientes que tenemos con el Sistema Operativo Debían y Security Onion. [25]

Para instalar OpenVPN hay que irse a la consola y poner en la línea de comandos desde root **apt-get install openvpn openssl**.

Instalaremos el servicio VPN y pasaremos a configurarlo de la siguiente forma en el servidor añadiendo o modificando dichas líneas en el archivo serv.conf: (en el anexo se encuentra el fichero de configuración del serv.conf)

Local 172.30.5.9 # Indica la IP del servidor VPN

Port 1194 # Indica el Puerto que utilizará la conexión VPN

Proto udp # Indica que utilizará UDP

Dev tun # Indica que utilizará una versión tunelizada

Ca /etc/openvpn/ssl/ca.crt # Agregación de Certificados

Cert /etc/openvpn/ssl/server.crt # Agregación de Certificados

Key /etc/openvpn/ssl/server.key # Agregación de Certificados

Dh /etc/openvpn/ssl/dh2048.pem # Longitud de la clave RSA utilizadas al generar las claves del servidor y del cliente.

Comp-lzo # Habilita la compresión LZO.

Server 10.5.0.0 255.255.255.0 # Rango de dirección que tendrá el túnel

Client-config-dir ccd # Directorio de configuración de clientes estáticos

Ifconfig-pool-persist ipp.txt # Para que las instancias múltiples no sobrescriban los archivos de salida de los demás.

Duplicate-cn # Duplicidad de certificados

Keeplive 10 120 # Tiempo de vida.

Cipher AES-256-CBC # Cifrado que utiliza

Persist-key # Al reiniciar accede al mismo certificado

Persist-tun # y privilegios que tenía antes.

Status openvpn-status.log # Genera una lista de conexiones de clientes actuales.

Verb 3 # Muestra más información en los registros.

Explicit-exit-notify 1 # Trabaja sobre UDP (si se comenta, es que se desea trabajar sobre TCP en lugar de UDP).

Una vez que se ha configurado lanzaremos el servicio mediante el comando: **openvpn –config /etc/openvpn/serv.conf** y nos dará la siguiente respuesta

```
root@ARIARX001:/etc/openvpn# openvpn /etc/openvpn/server.conf
Thu Jul 12 09:32:34 2018 OpenVPN 2.4.0 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [L
Z4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] built on Jul 18 2017
Thu Jul 12 09:32:34 2018 library versions: OpenSSL 1.0.2l  25 May 2017, LZO 2.08
Thu Jul 12 09:32:34 2018 Diffie-Hellman initialized with 2048 bit key
Thu Jul 12 09:32:34 2018 Outgoing Control Channel Authentication: Using 160 bit mes
sage hash 'SHA1' for HMAC authentication
Thu Jul 12 09:32:34 2018 Incoming Control Channel Authentication: Using 160 bit mes
sage hash 'SHA1' for HMAC authentication
Thu Jul 12 09:32:34 2018 ROUTE_GATEWAY 172.30.5.254/255.255.255.0 IFACE=ens192 HWAD
DR=00:50:56:b6:70:e5
Thu Jul 12 09:32:34 2018 TUN/TAP device tun0 opened
Thu Jul 12 09:32:34 2018 TUN/TAP TX queue length set to 100
Thu Jul 12 09:32:34 2018 do_ifconfig, tt->did_ifconfig_ipv6_setup=0
Thu Jul 12 09:32:34 2018 /sbin/ip link set dev tun0 up mtu 1500
Thu Jul 12 09:32:34 2018 /sbin/ip addr add dev tun0 local 10.5.5.1 peer 10.5.5.2
Thu Jul 12 09:32:34 2018 /sbin/ip route add 10.5.5.0/24 via 10.5.5.2
Thu Jul 12 09:32:34 2018 Could not determine IPv4/IPv6 protocol. Using AF_INET
Thu Jul 12 09:32:34 2018 Socket Buffers: R=[212992->212992] S=[212992->212992]
Thu Jul 12 09:32:34 2018 UDPv4 link local (bound): [AF_INET]172.30.5.9:1194
Thu Jul 12 09:32:34 2018 UDPv4 link remote: [AF_UNSPEC]
Thu Jul 12 09:32:34 2018 MULTI: multi_init called, r=256 v=256
Thu Jul 12 09:32:34 2018 IFCONFIG POOL: base=10.5.5.4 size=62, ipv6=0
Thu Jul 12 09:32:34 2018 IFCONFIG POOL LIST
Thu Jul 12 09:32:34 2018 Initialization Sequence Completed
^CThu Jul 12 09:32:47 2018 event_wait : Interrupted system call (code=4)
Thu Jul 12 09:32:49 2018 /sbin/ip route del 10.5.5.0/24
Thu Jul 12 09:32:49 2018 Closing TUN/TAP interface
Thu Jul 12 09:32:49 2018 /sbin/ip addr del dev tun0 local 10.5.5.1 peer 10.5.5.2
Thu Jul 12 09:32:49 2018 SIGINT[hard,] received, process exiting
```

Ilustración 14 Configuración Openvpn servidor

Ahora cuando realizamos Ifconfig para ver nuestras tarjetas de red, nos damos cuenta que nos muestra otra interfaz llamada **tun0** que se corresponde con nuestra VPN como se muestra en la figura:

```
root@ARIARX001:/etc/openvpn# ifconfig
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.30.5.9  netmask 255.255.255.0  broadcast 172.30.5.255
        inet6 fe80::250:56ff:feb6:70e5  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b6:70:e5  txqueuelen 1000  (Ethernet)
        RX packets 256265  bytes 15777079 (15.0 MiB)
        RX errors 0  dropped 53  overruns 0  frame 0
        TX packets 17175  bytes 2490851 (2.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.5.5.1  netmask 255.255.255.255  destination 10.5.5.2
        inet6 fe80::c141:6b08:b55c:797f  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UN
SPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3  bytes 144 (144.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ilustración 15. Interfaces abiertas

Podemos ver como se encuentra el proceso levantado con el puerto anteriormente configurado:

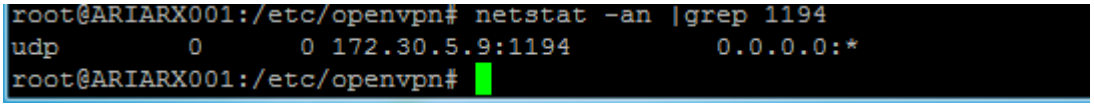


Ilustración 16. Proceso buscado activo

A continuación nos iremos a la parte del cliente en el que también instalaremos el servicio VPN. Lo primero será traer los certificados del servidor al cliente usando el comando scp y así tendríamos nuestro certificado ca.crt en el lado del cliente y nos ponemos a configurar el fichero client.conf añadiendo o modificando las líneas siguientes: (En el anexo se encuentra el fichero de configuración client.conf)

Client # Indica que será un cliente
Dev tun # Indica que utilizará una versión tunelizada
Proto udp # Indica que utilizará una versión tunelizada
remote 141.136.59.251 1194 # Indica la IP del servidor al que se conectará
Resolv-retry infinite # Si la resolución del nombre falla, lo intentara infinitamente hasta que se pueda conectar al servidor.
Nobind # No enlazar a la dirección local y el puerto
Persist-key # Al reiniciar accede al mismo certificado
Persist-tun # y privilegios que tenía antes.
Ca /etc/openvpn/ssl/ca.crt # Agregación de Certificados
Cert /etc/openvpn/ssl/server.crt # Agregación de Certificados
Key /etc/openvpn/ssl/server.key # Agregación de Certificados
Remote-cert-tls server # Obliga al certificado del otro extremo a ser un certificado de servidor
Cipher AES-256-CBC # Cifrado que utiliza
Comp-lzo # Habilita la compresión LZO.
Verb 3 # Muestra más información en los registros.

Una vez configurada la VPN, lanzaremos el servicio de la misma manera que lo hicimos en el servidor, mediante el comando: **openvpn –config /etc/openvpn/client.conf**

Ya estaría configurado el tunel en el **servidor** que tiene una direccion IP de **172.30.5.9** teniendo la direccion de tun0 **10.5.0.1** la del **cliente uno** llamado ccarvajal que tiene una dirección IP de **192.168.1.9** con la interfaz tun0 con direccion **10.5.0.18**(generalmente cuando se reinicia coge una dirección que le da el servidor de la red 10.5.0.0/24), el **cliente dos** con direccion IP **192.168.1.10** y una interface tun0 de **10.5.0.22** y asi sucesivamente para cada cliente. Pues si ahora realizamos un ping o un ssh a la direccion tun0 debería darnos conexión.

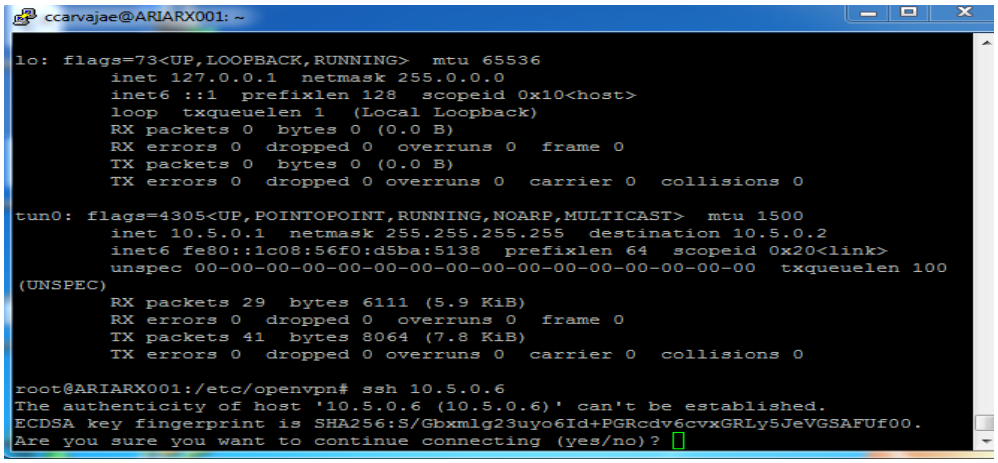


Ilustración 17 Prueba conexión ssh

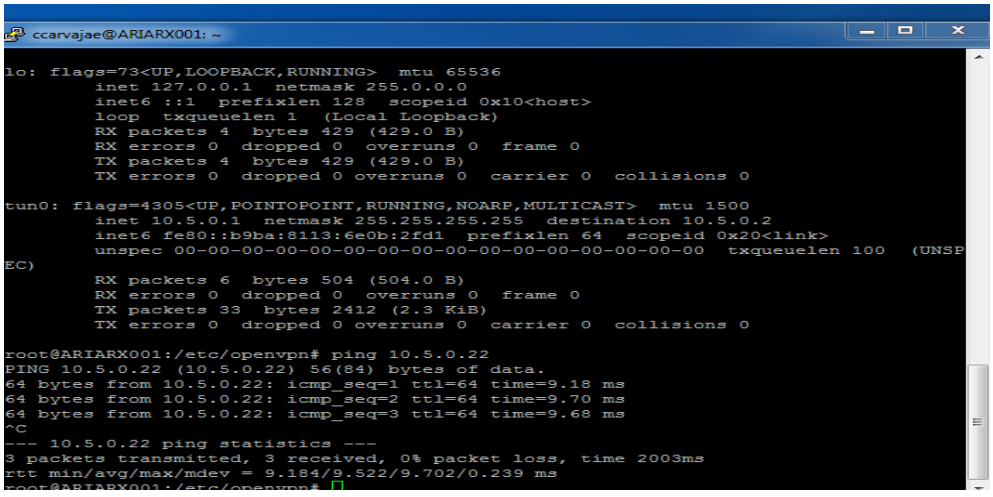


Ilustración 18 Prueba ping conexión ssh

A continuación hay que definir otra tarjeta de red que será la utilizada para el port Mirroring, la crearemos configurando el fichero /etc/network/interface como se aprecia en la siguiente imagen:

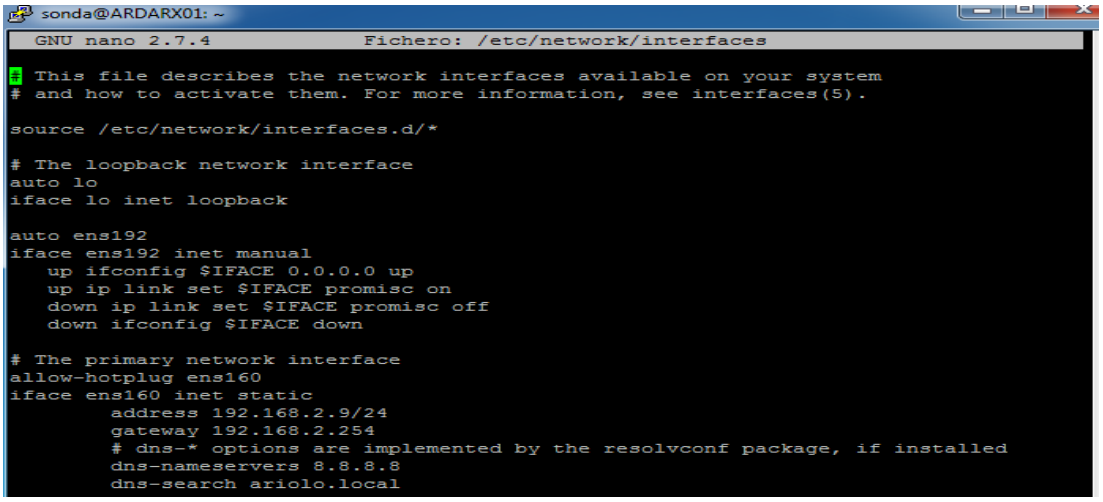


Ilustración 19 Creación Interfaz Port Mirroring

Hay que hacerlo para cada cliente que tengamos de manera que nos muestre las siguientes interfaces:

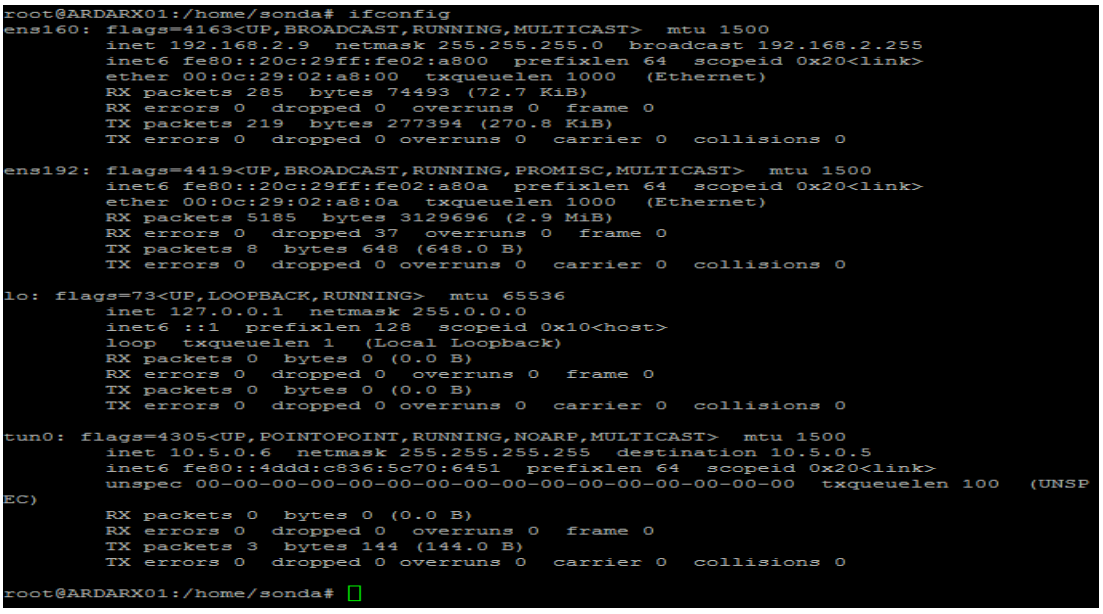


Ilustración 20 Interfaces abiertas

5.2 Instalación y configuración de Elasticsearch

Una vez instalado la comunicación segura entre clientes y servidor mediante openVPN, en el propio servidor instalaremos Elasticsearch y kibana para que podamos mostrar todas los tipos de incidencias que indiquemos más tarde a través de otras herramientas que más tarde indicaremos.

Para instalar Elasticsearch se requiere al menos de java 8 por lo que lo instalaremos en la consola a través de root con el comando **apt-get install Oracle-java8-installer** y a continuación nos disponemos a instalar Elasticsearch.

En nuestro caso instalaremos Elasticsearch con la versión 6.3.1 y para ello debemos bajarnos del repositorio el archivo de instalación .tar.gz por lo que en la línea de comandos introducimos **curl -L -O https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.3.1.tar.gz**. Una vez descargado lo descomprimimos con tar –xvf Elasticsearch-6.3.1.tar.gz, nos metemos en la carpeta de instalación y añadiremos "./Elasticsearch" para que se instale.

Ahora pasaremos a la configuración de nuestro Elasticsearch una vez instalado. Por lo que nos vamos a la ruta **/etc/Elasticsearch** y ahí configuraremos el archivo Elasticsearch.yml añadiendo o modificando las siguientes líneas: (en el anexo se encuentra la configuración de Elasticsearch.yml).

Path.data: /var/lib/Elasticsearch #Lugar donde se almacenarán datos
Path.log: /var/log/Elasticsearch #Lugar para ver los Logs.
Network.host: 10.5.0.1 # Dirección de IP para el servidor Elasticsearch
http.port: 9200 #Puerto que utilizará para conectarse.

Para que Elasticsearch se inicie nada más encender el ordenador y cargar el sistema operativo debemos poner los siguientes comandos en la consola para que se inicie el demonio.

sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service

Podemos encender el servicio introduciendo service Elasticsearch start

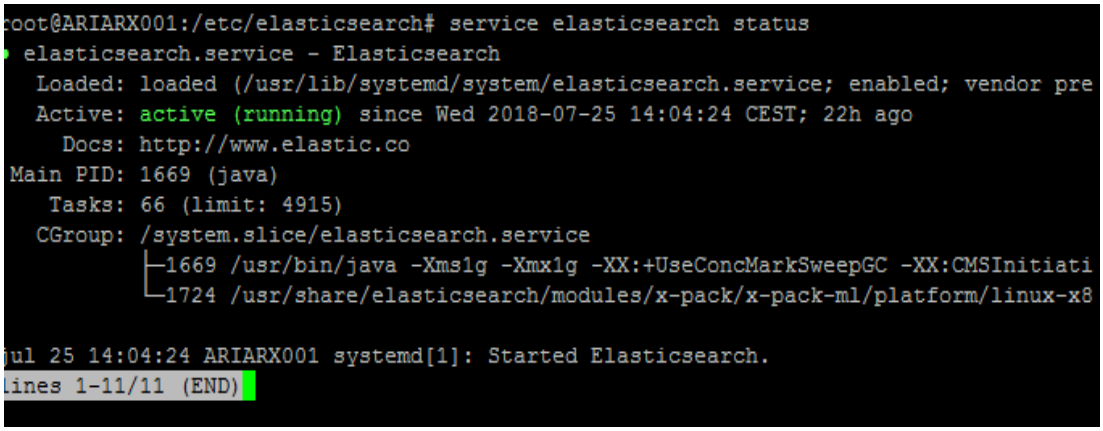


Ilustración 21 Estado Elasticsearch

5.3 Instalación y configuración de kibana

A continuación de configurar Elasticsearch nos disponemos a instalar kibana para después configurarlo. Para ello nos vamos a la línea de comandos y e introduciremos **wget https://artifacts.elastic.co/downloads/kibana/kibana-6.3.1-amd64.deb**. Una vez descargado el paquete, lo instalaremos con **dpkg –i kibana-641-amd64.deb** para a continuación realizar la configuración del fichero. [26]

Nos dirigimos al fichero **/etc/kibana/kibana.yml** y añadiremos o modificaremos las siguientes líneas: (en el anexo se encuentra la configuración de kibana.yml).

Server.port: 5601 # Puerto que utilizará kibana
Server.host: “172.30.5.9” # Dirección IP para acceder a la página de kibana
Elasticsearch.url: “http://10.5.0.1:9200” # Dirección para conectar kibana con Elasticsearch

Como hicimos con Elasticsearch, pondremos Kibana para que se inicie el servicio nada más iniciar el sistema operativo, por lo tanto pondremos en la consola:

sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable kibana.service

Al igual que en Elasticsearch podemos iniciar, parar, reiniciar o ver el estado el servicio utilizando **service kibana [start – status – restart – stop].**

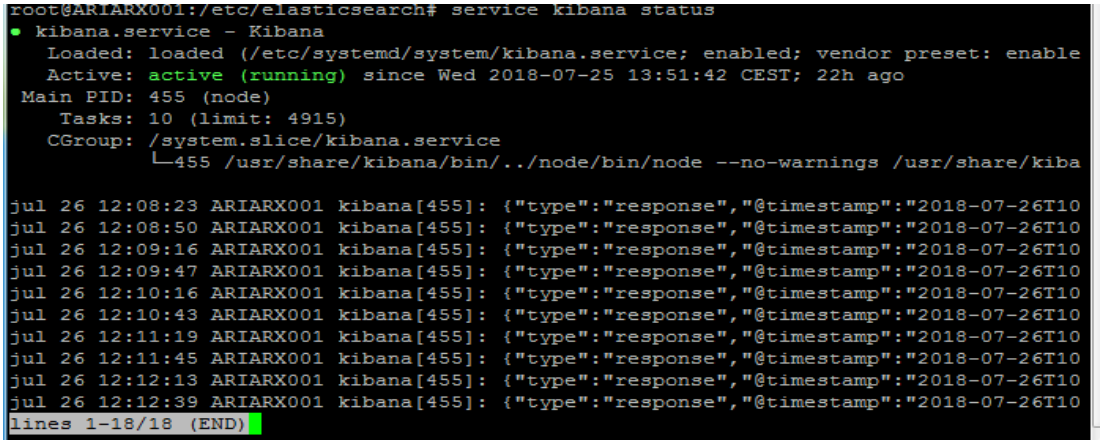


Ilustración 22 Estado Kibana

5.4 Instalación y configuración de Suricata

Una vez configurado lo anterior, en cada cliente nos instalaremos **Suricata** que será el encargado de recoger todas las intrusiones de nuestro PC en formato Json para más tarde enviarlo a nuestro servidor. Esta configuración vale para los clientes que tenemos con el Sistema Operativo Debían y Security Onion.

Instalamos el paquete de suricata en la consola añadiendo **apt-get install suricata** desde la línea de comandos.

Para configurar nuestro Suricata, tendremos que configurar el fichero /etc/suricata/suricata.yaml e indicar la interface del port Mirroring que será el que se encuentre a la escucha (en nuestro caso la asignaremos a la interfaz ens192). [27]

Ahora mismo nuestro suricata está configurado y corriendo, pero de poco sirve pues debemos ejecutar la herramienta suricata-oinkmaster-update para descargar todas las reglas y pueda detectarlas.

Un ejemplo muy sencillo para indicar una alerta es crear un fichero y meterlo en la carpeta rules de suricata y especificar qué tipo de alerta queremos detectar, en este caso añadiremos una para detectar los paquetes de ICMP poniendo en el fichero **alert icmp any any ->any any (msg: "ICMP detected").** Y como se muestra en la siguiente imagen nos mostrará todos los icmp detectados

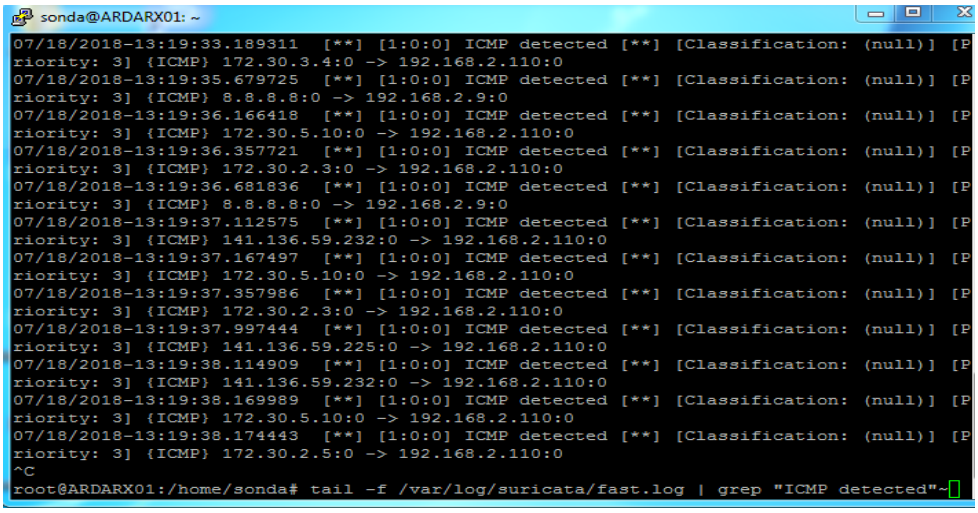


Ilustración 23 Captación de paquetes Suricata (Fast.log)

Primero de todo oinkmaster es una herramienta para descargar y administrar reglas para Snort y Suricata y para instalarlo nos vamos a la línea de comandos de la consola y añadimos **apt-get install oinkmaster.**

Para configurar **oinkmaster** y hacer que se descargen y/o actualicen todas las reglas realizaremos con el comando **crontab –e** y agregaremos la linea **0 0 * * * oinkmaster –C /etc/oinkmaster.conf –o /etc/suricata/rules**, esta linea sirve para que llame al fichero de configuracion oinkmaster.conf pues dentro de ese fichero pondremos una url desde donde se encuentran las reglas actualizadas, para ello lo indicamos poniendo en el fichero la linea **url = http://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz**

Para que nuestro suricata se inicie nada más iniciar el ordenador debemos añadir las siguientes líneas en la consola para activar el demonio que haga que se cargue el proceso al encendido del sistema operativo:

sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable suricata.service

5.5 Instalación y configuración de Filebeat

A través de suricata, ya está cogiendo los sucesos que pasan en el ordenador cliente y poniéndolo en los ficheros **eve.json** y **fast.json** por lo que ahora necesitaremos de un método para lograr pasar a través del túnel de cada cliente, los registros que queramos enviar al servidor. Para ello utilizaremos **Filebeat** que realizará todo el proceso de envío al servidor.

Esta configuración vale para los clientes que tenemos con el Sistema Operativo Debían y Security Onion.

Una manera de poder instalarlo será a través de paquetes .Dev por lo que usaremos **curl -L –O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.3.1-amd64.deb** para después instalarlo con **sudo dpkg –i filebeat-6.3.1-amd64.deb**.

Para configurar Filebeat deberemos modificar el fichero **/etc/filebeat/filebeat.yml** y añadir o modificar las siguientes líneas: (en el anexo se encuentra el fichero filebeat.yml).

Filebeat.inputs: #Lugar donde se configura la entrada de datos que quieren enviar al servidor.

- **Type: log** # Tipo de datos a enviar, en este caso tipo logs
  Enable: true # Para activar la configuración de Filebeat.inputs.
  Paths: # Sección para indicar los ficheros a enviar.
  - **/var/log/suricata/eve.json** # Ruta de fichero a enviar al servidor.

Filebeat.config.modules:

Paths: ${path.config}/modules.d/*yml # patrón para la carga de configuración.

Reload.enabled: true #habilita la carga de la configuración

Setup.template.settings: #configuración de template.

Index.number_of_shards: 3 # Cantidad de fragmentos de template.

Setup.dashboards.enabled: true #controla la carga del template (tablero) que muestra al índice kibana.

Setup.kinana: #configuración para conectar a servidor kibana

Host: "172.30.5.9:5601" #host donde se encuentra el servidor de kibana con el puerto a la escucha.

#Output.elasticsearch: #configuración para conectar al servidor Elasticsearch

#Hosts: [10.5.0.1:9200"] #host donde se encuentra el servidor Elasticsearch con el puerto a la escucha.

Output.logstash: #configuración para conectar al servidor Logstash

Hosts: ["10.5.0.1:5044"] #host donde se encuentra el servidor con el Puerto la escucha.

En cuanto a **#output.elasticsearch** y **Output.logstash** uno de los dos tiene que estar comentado ("**#**") pues uno es para mandar la salida del archivo descrito anteriormente en Filebeat.inputs directamente a Elasticsearch o pasando por Logstash.

5.6 Instalación y configuración de Logstash

A continuación vamos a configurar para que filebeat pueda enviar los registros indicados a **Logstash** en el servidor. Para la instalación basta con **apt-get install Logstash** con lo que ahora nos dispondremos a la configuración.

En cuanto a la instalación de **Logstash** nos dirigimos a la carpeta **/etc/Logstash** y allí nos encontramos una serie de archivos en los que solo configuraremos **Logstash.yml** y también se encuentra una carpeta llamada

conf.d en la que se encuentra la verdadera configuración de **Logstash** en nuestro caso sería el archivo **Logstash.conf.** [28

En el archivo **Logstash.yml** modificaremos o añadiremos las siguientes líneas: (en el anexo se encuentra el archivo Logstash.yml)

Path.data /var/lib/Logstash #Lugar donde se almacenarán datos
Path.logs: /var/log/Logstash #Lugar donde se guardaran y se verán los logs.

En el archivo **Logstash.conf** es, como comentamos anteriormente, donde se realizará la verdadera configuración. Entrará desde la entrada de datos, la salida y un filtrado de los datos para que se puedan parsear y podamos analizar los datos más individualmente.

```
input {        # Especifica la entrada de los datos, por el puerto que vienen.
  beats {
    port => "5044"
    ssl  => false
  }
}
```

filter { # campo donde se especifican los filtros de los eventos que han sido recibidos a traves de filebeat.

json{ #Realiza el parseo de los datos para que se descompongan en subcampos

```
      source => "message"
    }
    geoip { # Realiza la geolocalización
      source => "src_ip"
      target => "geoip"
     add_field => [ "[geoip][coordenates]", "%{[geoip][longitude]}" ]
     add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
    }
```

mutate { #Convierte los datos de las localizaciones en float para poder utilizarlos

```
      convert => [ "[geoip][coordinates]", "float"]
    }
    geoip {# Realiza la geolocalización
      source => "dest_ip"
      target => "geoip"
      add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
      add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
     }
     mutate {#Convierte los datos de las localizaciones en float para poder utilizarlos
    convert => [ "[geoip][coordinates]", "float"]
     }
}
output { #Realiza la salida de los datos, (a quien se lo mandamos)
  elasticsearch {   #Realiza el envío de los eventos registrados a elasticsearch
    hosts => ["http://10.5.0.1:9200"]
    sniffing => true
    manage_template => false
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    document_type => "%{[@metadata][type]}"
   }
  stdout {
    codec => "rubydebug"
  }
}
```

5.7 Instalación y configuración de Bro

En cuanto al Security Onion hemos probado con otra herramienta para recopilar información de este cliente, se llama **Bro.**

Es una herramienta que Security Onion la trae instalada y en cuanto a la configuración, dentro de la consola ejecutaremos el comando **Broctl** y nos saldrá un una consola. Lo primero que debemos realizar es un chequeo a la configuración de Bro y ese comando será **deploy.** Para ver las estadísticas de trafico de red lo realizaremos con el comando **captats.**

Hay que configurar 2 ficheros llamados **local.bro, node.cfg.** En cuanto a local bro es para definir las redes internas por lo que se configurará como

Redef Site::local_nets = {
10.5.0.0/8
192.168.2.0/24
};

El fichero de node.cfg es para definir la interfaz de captura y para configurarlo lo realizamos de la siguiente forma:

[bro]
Type=standalone
Host=192.168.2.11
Interface=ens160

Una vez configurado estos ficheros Bro nos va a dar información ya sea de resumen de conexiones, conexiones, DNS, SSH, SSL, HTTP entre otras.

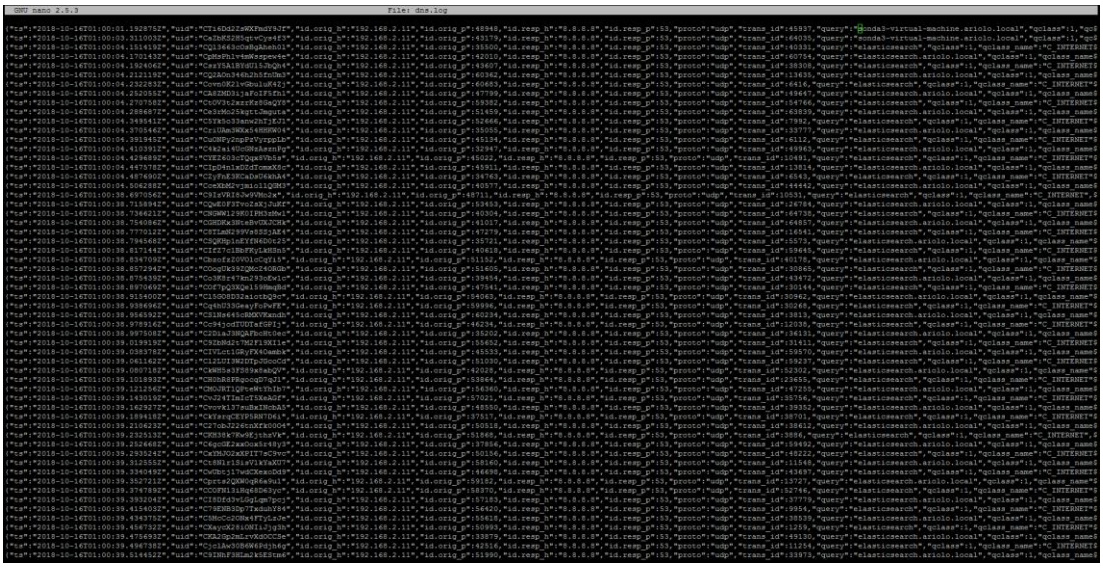


Ilustración 24 Archivo dns.log de Bro

Por ejemplo en el primer archivo que es dns.log la primera línea seria esta:

{"ts":"2018-10-16T01:00:01.192875Z","uid":"CTi6Dd2ZsWXFmdY9Jf","id.orig_h":"192.168.2.11","id.orig_p":48948,"id.resp_h":"8.8.8.8","id.resp_p":53,"proto":"udp","trans_id":45937,"query":"sonda3-virtual-machine.ariolo.local","qclass":1,"qclass_name":"C_INTERNET","qtype":28,"qtype_name":"AAAA","rcode":3,"rcode_name":"NXDOMAIN","AA":false,"TC":false,"RD":true,"RA":false,"Z":0,"rejected":false}

Esto nos indica que el día 16 de octubre a la 1:00 ha habido una conexión dns con la IP 192.16.2.11 con una id 48948 y le ha respondido el dns 8.8.8.8 por el puerto 53 por udp al cliente sonda3-virtual-mmachine.ariolo.local para conectarse a internet.

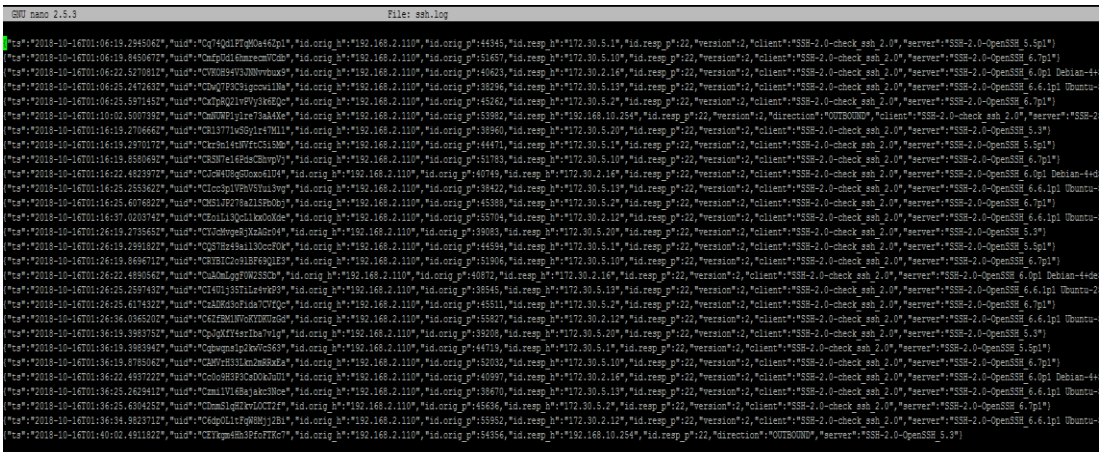


Ilustración 25 Archivo ssh.log de Bro

Por ejemplo en el primer archivo que es dns.log la primera línea seria esta:

{"ts":"2018-10-16T01:06:19.294506Z","uid":"Cq74Qd1PTqMOa46Zp1","id.orig_h":"192.168.2.110","id.orig_p":44345,"id.resp_h":"172.30.5.1","id.resp_p":22,"version":2,"client":"SSH-2.0-check_ssh_2.0","server":"SSH-2.0-OpenSSH_5.5p1"}

Esto indica que ha habido una conexión ssh de la IP 192.168.2.110 y a la dirección 172.30.5.1.

Cuando se encuentra con algún tipo de error, ya sea de la propia configuración o de algo que ha encontrado nos recibe un Email en la fichero **/var/spool/mail/root**

Una vez instalado y configurado todas las cosas explicadas con anterioridad (openvpn, Elasticsearch, kibana, filebeat, Logstash) nos disponemos a introducir en el buscador la dirección que configuramos para entrarnos en kibana, en nuestro caso corresponderá con la dirección **http://172.30.5.9:5601** y nos dirigirá a la siguiente ventana de kibana:

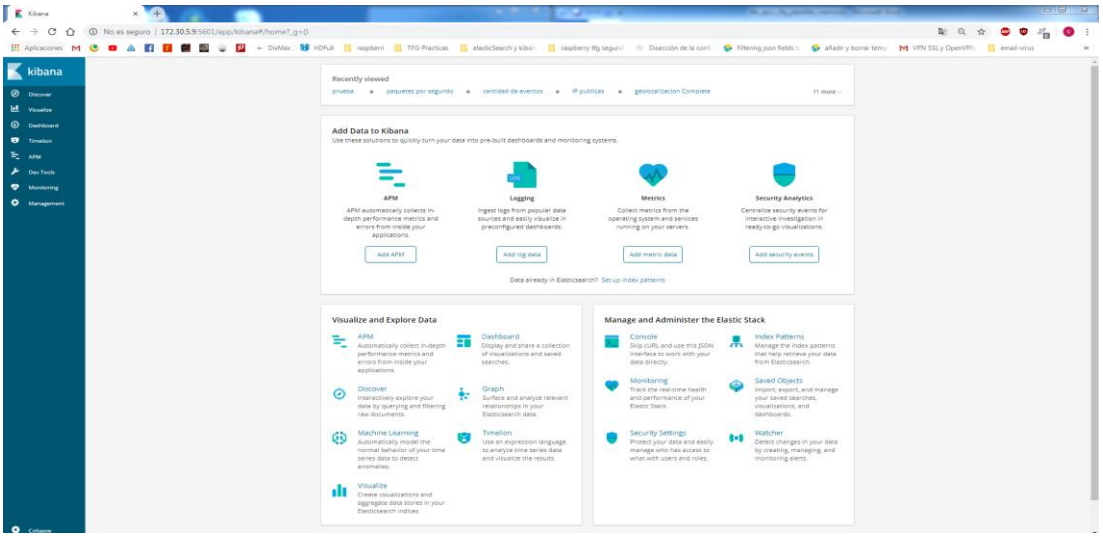


Ilustración 26 Página principal de Kibana

Dentro de **kibana**, si nos dirigimos a la pestaña **discover** nos mostrará todos los datos que han sido capturados por suricata y enviados por filebeat a nuestro servidor de nuestros clientes:

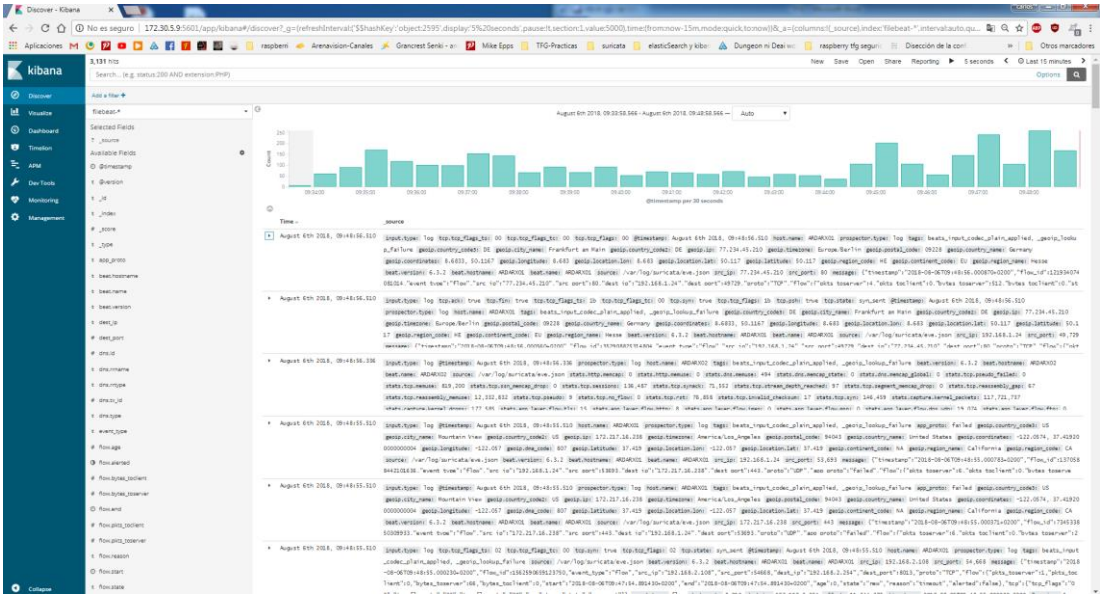


Ilustración 27 Pestaña de muestra de recolección de eventos de kibana

Como se muestra en la imagen anterior, nos permite mostrar la cantidad de eventos recogidos por hora, además de toda la información del evento.

Con la configuración de filtro que realizamos en Logstash a través de **Logstash.conf** nos permite ver cada evento de manera individual, para que

podamos ver por ejemplo, que IP destino tiene, que puerto, de donde es esa Dirección IP sin tener que buscar los datos uno a uno imaginando que dato corresponde con qué campo.

En cuanto a la zona geográfica solo nos la pueden dar si provienen de una IP pública ya que si viene de una IP privada, no logrará sacar la zona geográfica de donde se encuentran.

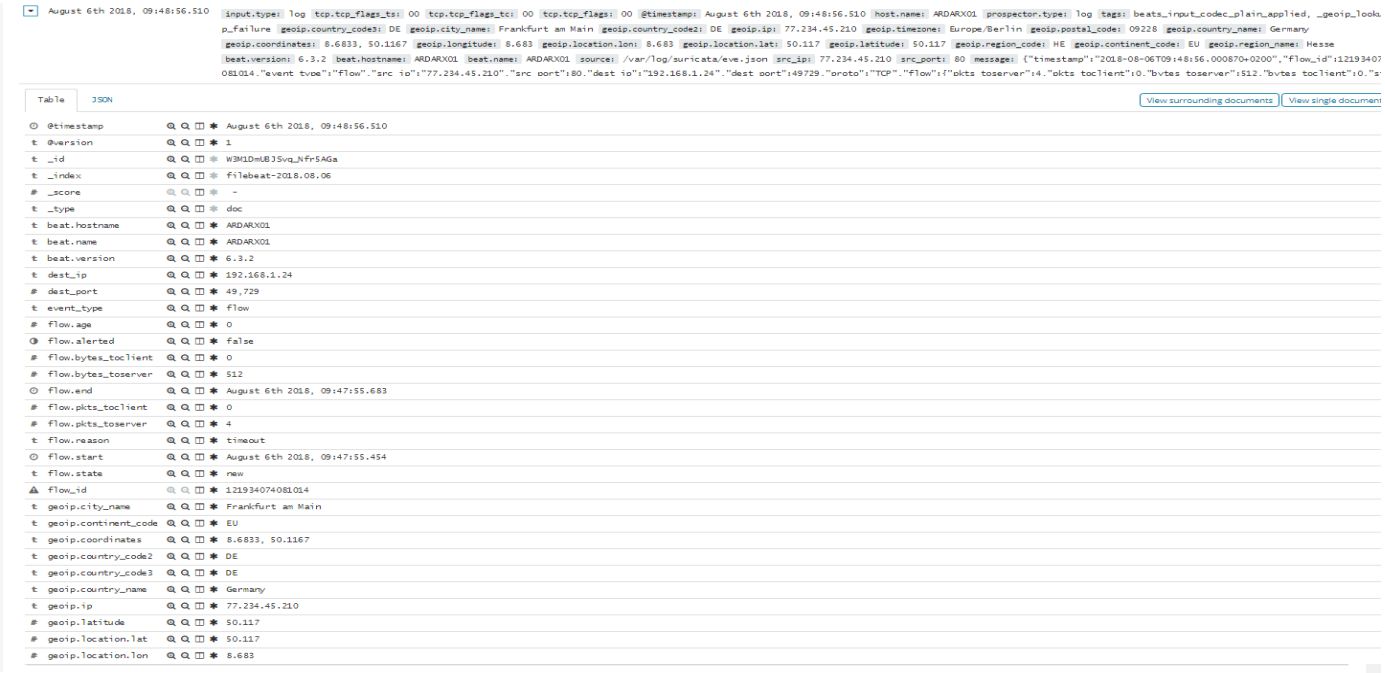


Ilustración 28. Muestra de un evento

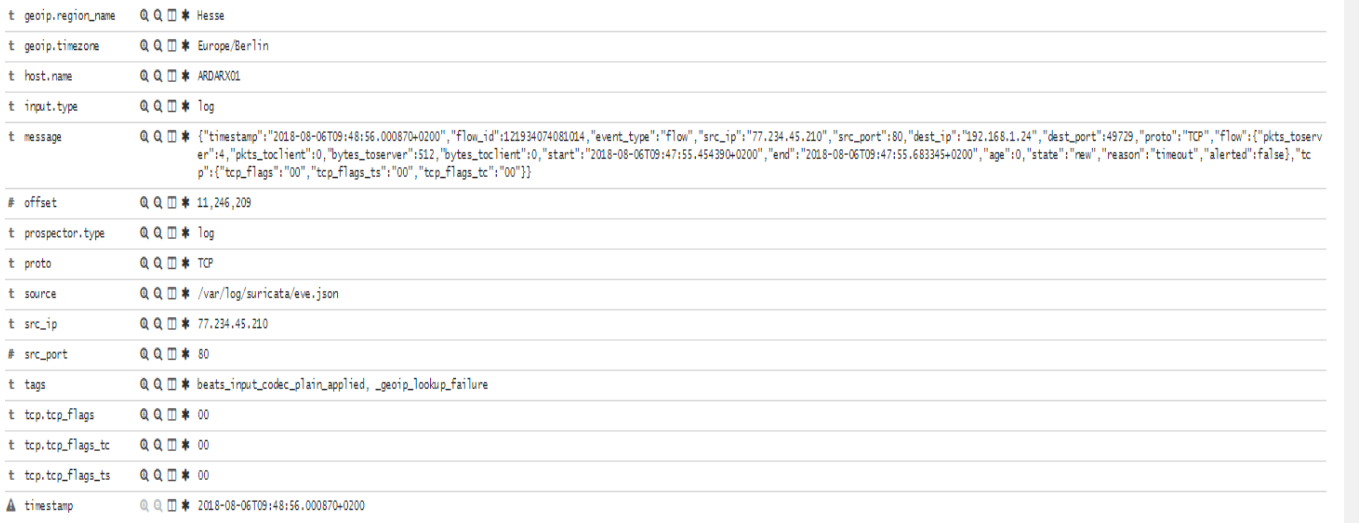


Ilustración 29 Continuación muestra de evento

Una vez mostrado los datos nos dispondremos a crear un **Dashboard** que no es más que una representación gráfica de los principales indicadores que intervienen en cada uno de los eventos que hemos estado recibiendo por parte de los clientes.

En nuestro **Dashboard** hemos incluido en primer lugar la zona geográfica de donde nos vienen los eventos que comunican con nuestros clientes además de una ruleta más específica en la que aparte de la ciudad de donde vienen también nos muestra quienes son los que más mandan datos.

También hemos incluido la cantidad de datos que nos llegan en cada una de las sondas (clientes) que tenemos y la cantidad de paquetes que recibimos en total, además de las IP y DNS de tales paquetes para lograr así un control más exhaustivo.

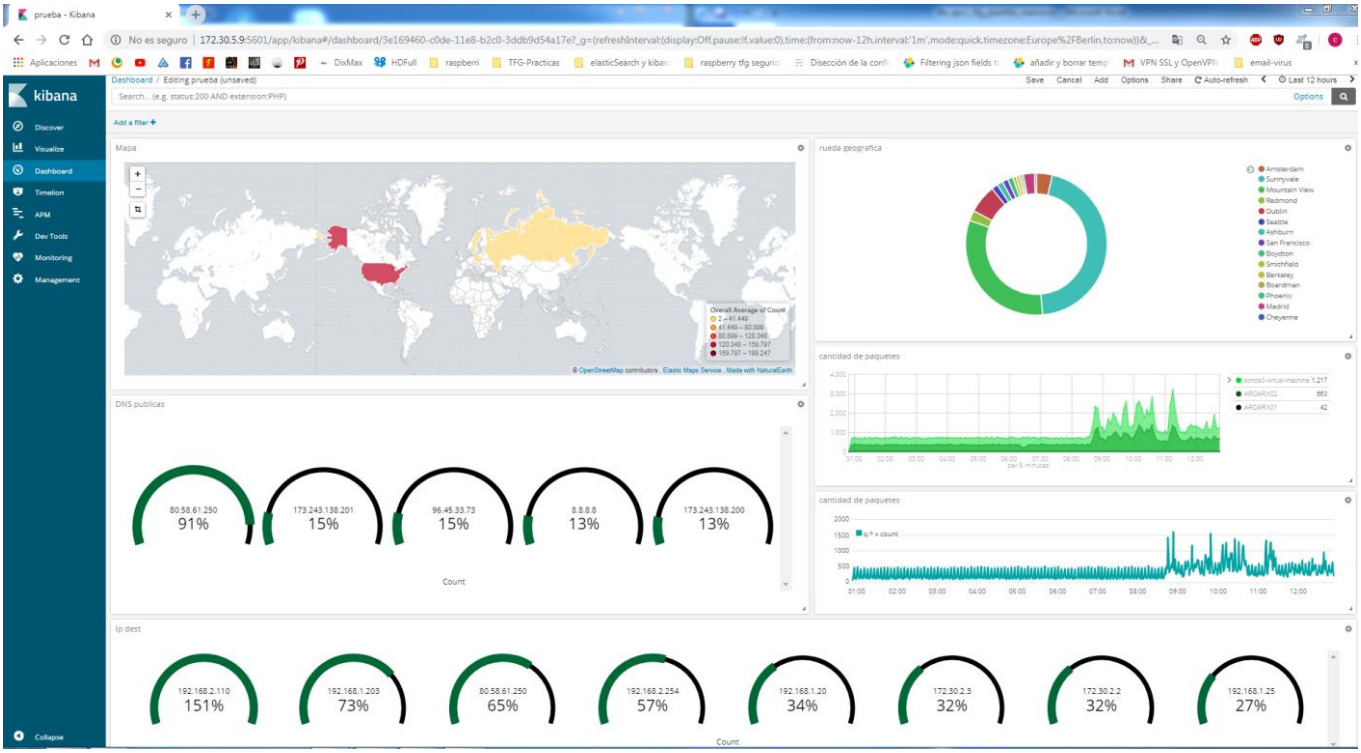


Ilustración 30. Dashboard de los eventos recogidos

6. Planificación temporal

Esta planificación del proyecto es dentro de las horas de TFG aunque dentro de ese horario también realicé horas fuera de la empresa, por búsqueda de información, fallos imprevistos o continuación de la arquitectura.

Fecha	Horas	Detalle de trabajo en Practicas
9/07-13/07	28	Practicas - Búsqueda de información
16/07-20/07	27	Practicas – instalación de servidores y Clientes
23/07-27/07	29	Practicas – instalación y configuración de OPENVPN en todos los clientes.
30/07-3/08	26	Practicas – instalación y configuración de Elasticsearch y kibana en clientes Debían
6/08-10/08	30	Practicas - fin de interconexión de Elasticsearch con kibana e instalación de Suricata en clientes Debían.
13/08-17/08	29	Practicas – instalación y configuración de Filebeat con envío de datos a Elasticsearch para muestreo por kibana en clientes Debían.
	Total:	169

Fecha	Horas	Detalle de trabajo en Practicas
20/08	8	Instalación de Logstash y configuración del archivo Logstash.conf para su recepción a través de filebeat
21/08	7	Carga de template de Logstash y configuración para parsear datos.
24/08	5	Instalación de Security Onion y configuración de OpenVPN
27/08	6	Configuración de suricata y Filebeat en Security Onion
28/08	7	Configuración Bro y Filebeat en Security Onion
6/09	8	Arreglo de fallos en muestreo de datos
7/09	7	Configuración de Dashboard con su muestreo de datos.
10/09	6	Prueba de instalación con Raspberry Pi
12/09	7	Empiece Documentación de TFG índice y titulo
14/09	8	Documentación TFG- introducción, objetivos y resumen
18/09	6	Documentación TFG – Solución propuesta y búsqueda de información más exhaustiva
19/09	8	Documentación TFG – punto Implementación
25/09	7	Documentación TFG- punto tecnología empleada y continuación Implementación TFG
27/09	7	Documentación TFG – manuales y conclusiones
1/10-3/10	9	Documentación TFG – Anexos y revisiones y modificación de TFG
5/09	7	Documentación TFG – Cambios propuestos por tutor
8/10	8	Documentación TFG- Realización de imágenes de arquitectura.
9/10	8	Documentación TFG- Planificación Temporal y más cambios propuestos por tutor
15/10	3	Realización y preparación de Power Point
	Total:	132

7. Problemas encontrados

- No tenia permiso en el servidor con la IP 192.168.2.9 y por lo tanto no se tenia comunicación.

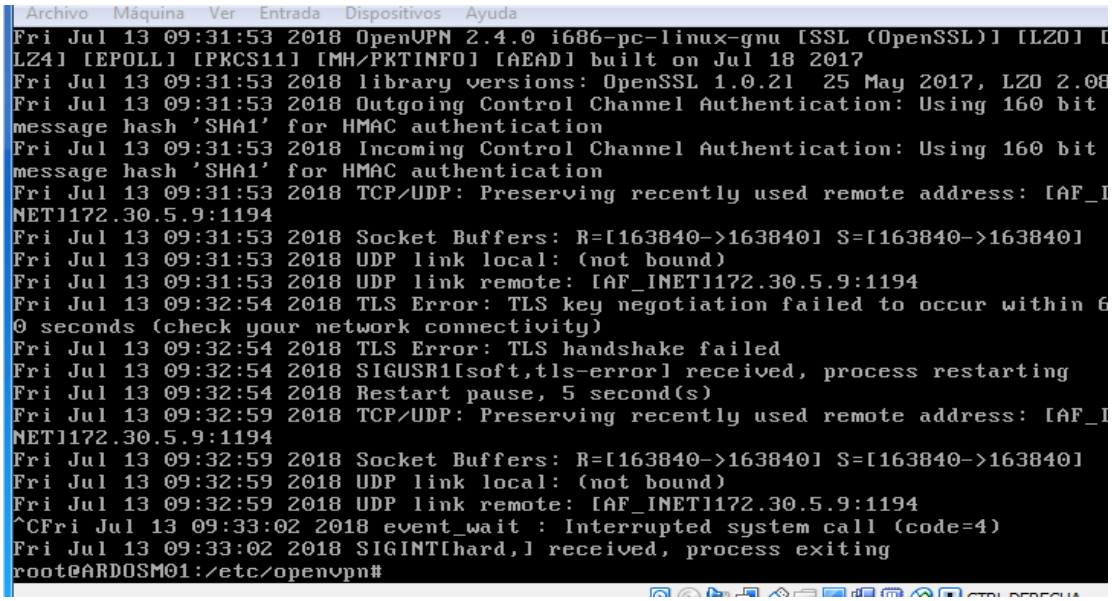


Ilustración 31. Error recogido de OpenVPN

- Options error: --tls-auth fails with 'ta.key': No such file or directory Options error: Please correct these errors. (se solucionó generando la clave con el comando openvpn—genkey—secret ta.key

- Error al instalar Logstash-> resuelto instalando java 8 ya que lo necesitaba.

- Fallo al iniciar Elasticsearch- > versión antigua, solucionado cambiando de versión 2.3 a 6.3.1

- tags: beats_input_codec_plain_applied, _geoip_lookup_failure sale fallo pues no puede encontrar la geolocalización de una IP privada. Pero una IP pública si la encuentra

- ERROR instance/beat.go:691 Exiting: Error importing Kibana dashboards: fail to import the dashboards in Kibana: Error importing directory /usr/share/filebeat/kibana: Failed to import index-pattern:

Failed to load directory /usr/share/filebeat/kibana/6/index-pattern: error loading /usr/share/filebeat/kibana/6/index-pattern/filebeat.json: blocked by: [FORBIDDEN/12/index read-only / allow delete (api)]. Response: {"objects":[{"id":"filebeat-*","type":"index-pattern","error":{"message":"blocked by: [FORBIDDEN/12/index read-only / allow delete (api)];"}}]} -→

- solución: curl -XPUT -H "Content-Type: application/json" http://10.5.0.1:9200/_all/_settings -d '{"index.blocks.read_only_allow_delete" : null}'

- Problema espacio de disco duro pues los logs ocupan demasiado. Solución: creación de unos ficheros en logrotate.d para que pueda rotarlos y comprimirlos cada hora y así no tener problema de espacio. (ver ficheros de configuración en /etc/logrotate.d/daemon y syslog

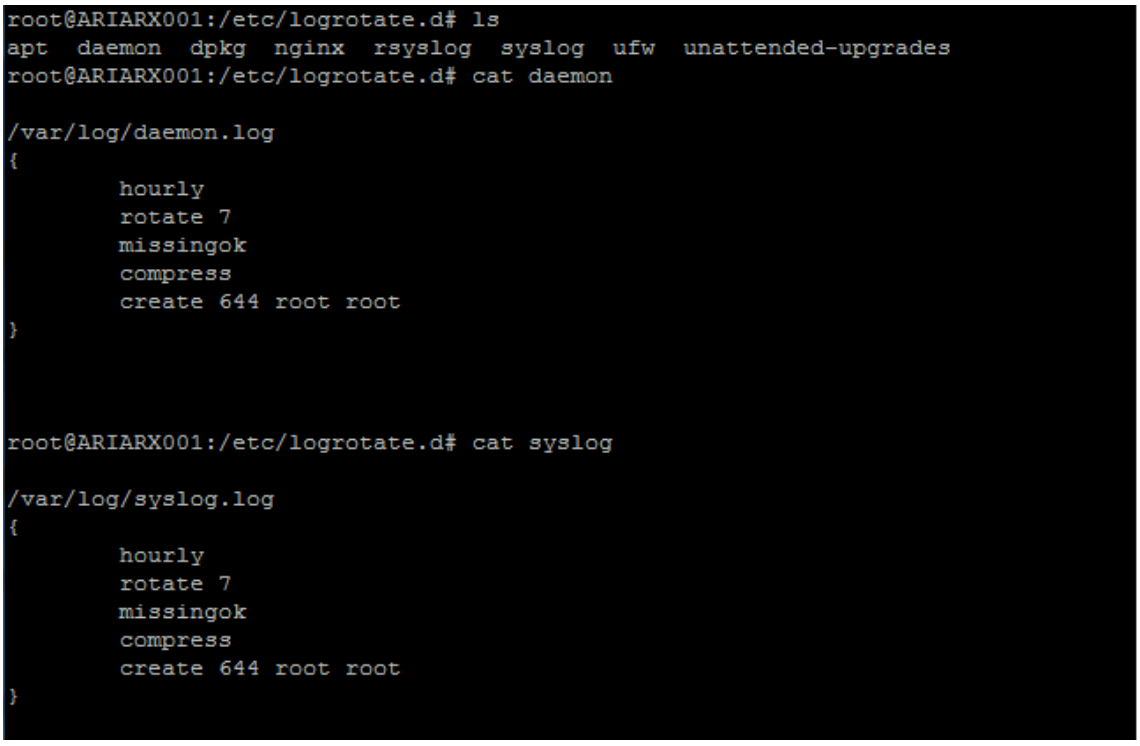


Ilustración 32. Script para borrar archivos de mucha memoria

8. Conclusiones

En cuanto a este proyecto, me ha resultado difícil tener que ver tecnologías y herramientas nunca vistas en lo que se estudia en el Grado. No obstante me ha gustado el tema que se ha escogido y he podido aprender bastante cosas.

Hemos tenido que ver qué herramientas funcionarían bien juntas, para poder lograr la estructura deseada aparte de tener que arreglar fallos imprevistos durante el proceso de la arquitectura.

No obstante se ha logrado realizar una arquitectura con la que muchas empresas deberían contar, logrando así una visualización y una gestión de todos los eventos que puedan recogerse y realizar así una intervención lo más rápido posible.

El hecho de que esta nueva herramienta mejore considerablemente la seguridad en las empresas, no implica que no puedan producirse fallos o intrusiones pues cada vez resulta más difícil darse cuenta a tiempo de ataques, aunque ayuda en gran parte a que no se produzcan.

9. Futuros Proyectos

En cuanto a proyectos futuros, hemos pensado en incorporar entornos embebidos como por ejemplo la raspberry que aunque lo hayamos probado, no lo incluimos al final en nuestra arquitectura. Esto demostraría que no se necesita una maquina con muchos recursos en el sistema para poder montar dicha arquitectura.

También pensamos en incorporar un sistema de aviso tanto por correo como por smartphone de infracciones severas que puedan producirse para poder gestionarla en el mismo momento en el que sucede.

Otra ampliación que se podría hacer es la de conectar a parte de clientes, dispositivos móviles, cámaras de seguridad y demás sistemas para prevenir que atacantes puedan lucrarse de ello.

Esto nos ayudará a ampliar el rango de clientes para que estén repartidos por el mundo y puedan tener una seguridad aunque usen el sistema para lo que fuera y lograr así que se sientan más seguros.

9. REFERENCIAS BIBLIOGRÁFICAS (Según norma ISO690)

[1] M. Bishop, "What is computer security?," URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1176998&isnumber=26429

[2] **Miguel Ángel Mendoza** 4 Jan 2018 - 12:02AM URL: https://www.welivesecurity.com/la-es/2018/01/04/maximo-historico-vulnerabilidades-2017/

[3] **Eutimio Fernández,** director de Ciberseguridad en Cisco España URL: https://www.itdigitalsecurity.es/actualidad/2017/10/cisco-el-modelo-de-seguridad-de-red-actual-no-fue-disenado-para-el-iot

[4]https://www.redeszone.net/2017/09/27/como-configurar-el-port-mirroring-en-el-switch-d-link-dxs-1100-10ts/

[5]http://www.alcancelibre.org/staticpages/index.php/configuracion-rsyslog

[6]https://www.rsyslog.com/

[7] **M.N González** (2002) URL: https://www.monografias.com/docs111/redes-privadas-virtuales-vpn/redes-privadas-virtuales-vpn.shtml

[8]https://www.definicionabc.com/tecnologia/vpn.php

[9]https://es.wikipedia.org/wiki/Secure_Shell

[10]https://es.wikipedia.org/wiki/PuTTY

[11]https://en.wikipedia.org/wiki/Elasticsearch

[12]https://openwebinars.net/blog/que-es-elk-elasticsearch-logstash-y-kibana/

[13]https://en.wikipedia.org/wiki/Kibana

[14]https://wikitech.wikimedia.org/wiki/Logstash

[15]https://es.wikipedia.org/wiki/Snort

[16]https://www.securityartwork.es/2015/07/28/bro-ids-el-ojo-que-todo-lo-ve/

[17]]https://web.nsrc.org/workshops/2015/pacnog17-ws/raw-attachment/wiki/Track2Agenda/ex-suricata-rules.htm

[18]https://en.wikipedia.org/wiki/Bro_(software)

[19]https://securityonion.net/

[20]https://es.wikipedia.org/wiki/Raspberry_Pi

[21]https://www.debian.org/intro/about.es.html

[22]https://es.wikipedia.org/wiki/Raspbian

[23]https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

[24]https://github.com/security-onion-solutions/security-onion/wiki/IntroductionToSecurityOnion

[25]https://www.digitalocean.com/community/tutorials/como-configurar-un-servidor-openvpn-en-ubuntu-16-04-es
https://fututel.com/es/tutoriales-guias-manuales-videotutoriales/2706-configurar-un-servidor-openvpn-en-debian-8

[26]https://www.elastic.co/guide/en/kibana/current/setup.html

[27]https://www.elastic.co/guide/en/logstash/current/configuration.html

[28]https://www.fwhibbit.es/suricata-ids-instalacion-puesta-en-marcha-y-primera-prueba

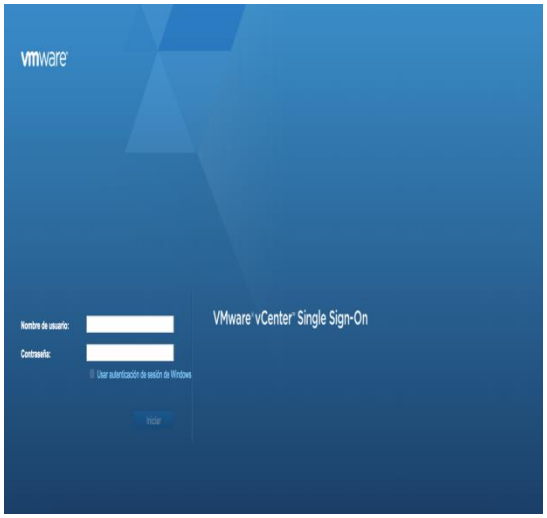
10. ANEXOS

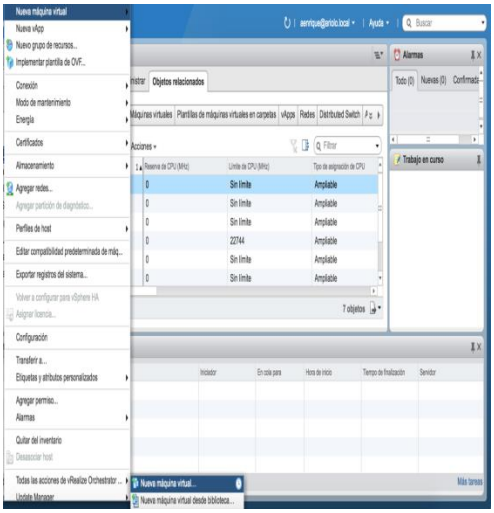
1. Configuración de Máquina Virtual

En esta parte del anexo explicaremos tanto la creación de una máquina virtual como la instalación del Sistema Operativo para poder trabajar con ellas en el proyecto.

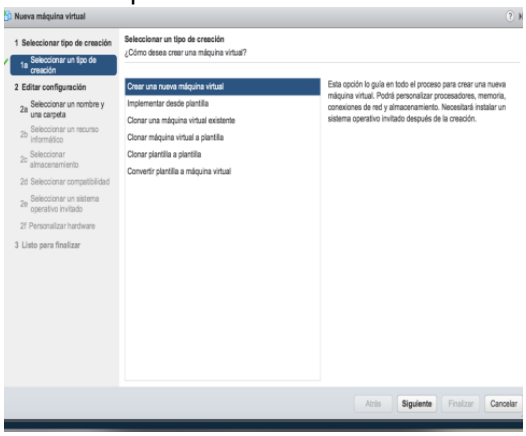
1.1 Creación de máquina virtual

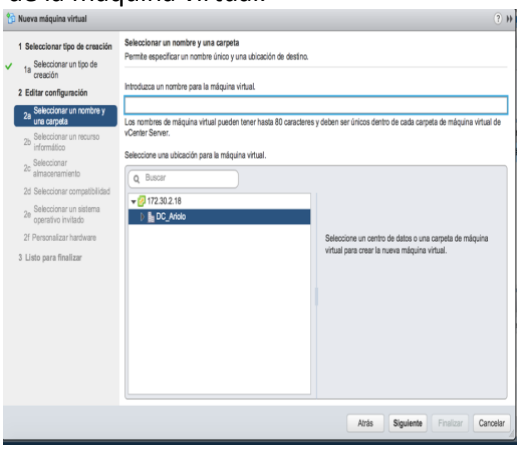
Para acceder al servidor, nos dirigimos a un navegador y escribimos en la barra de direcciones su dirección IP.



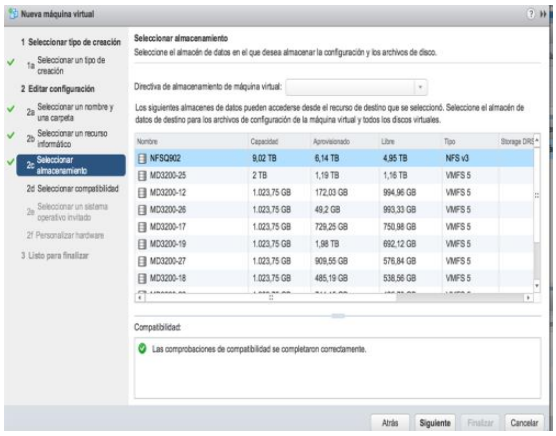
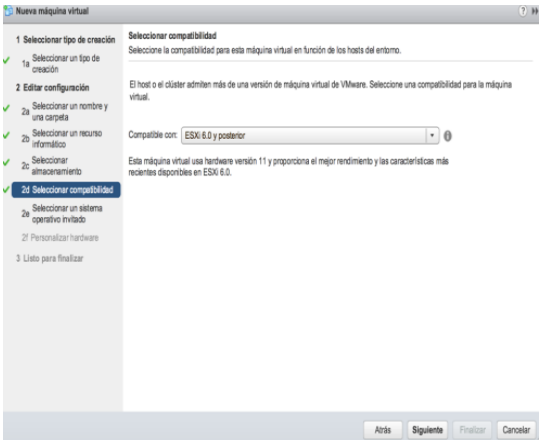


Para desplegar la máquina virtual, nos dirigimos encima de la plantilla y pulsamos sobre él con el botón derecho y seleccionamos “Nueva máquina virtual”. Seleccionamos “Crear una nueva máquina virtual” e introducimos el nombre de la máquina virtual.

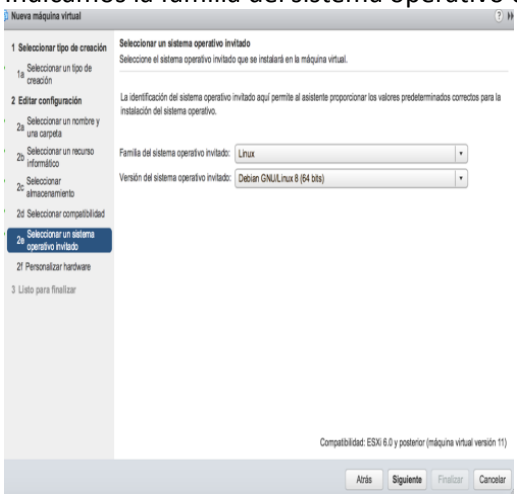
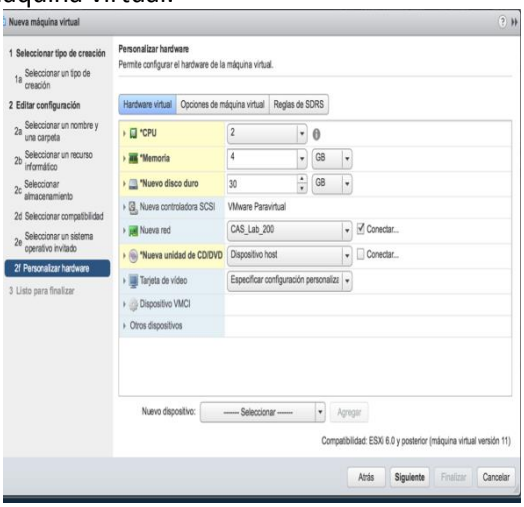




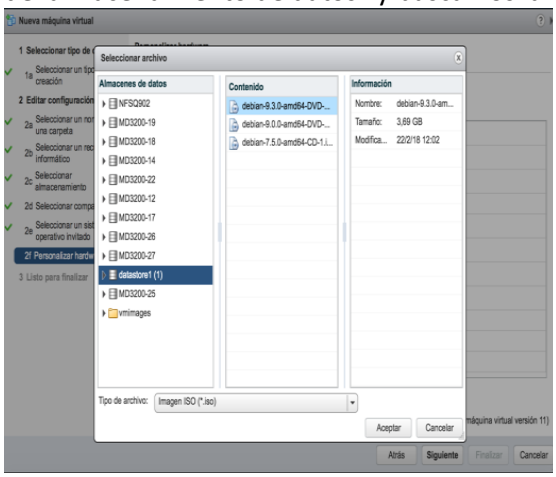
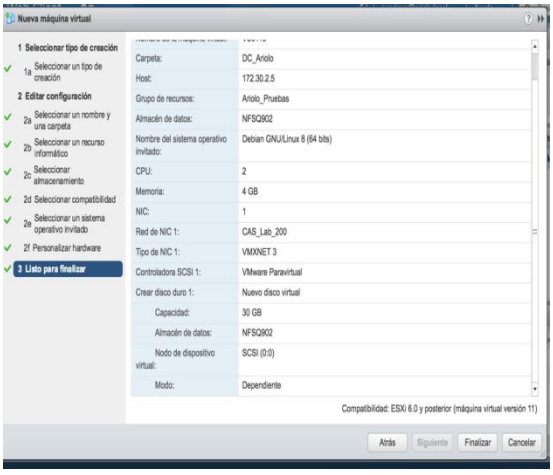
Seleccionamos el recurso informático para ejecutar la máquina virtual y el almacenamiento para el destino del almacenamiento de datos de la máquina virtual.

Seleccionamos la compatibilidad de la máquina virtual en función del host del entorno e Indicamos la familia del sistema operativo de la máquina virtual.

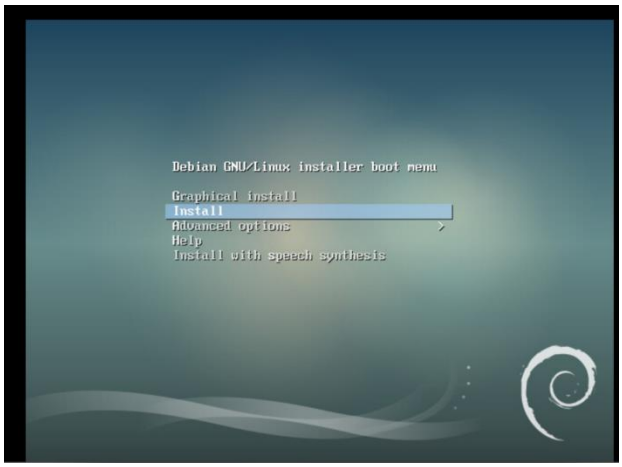
Ajustamos el hardware virtual. Si queremos poner una iso que esté guardada en el disco de almacenamiento seleccionado anteriormente, elegimos en la unidad CD/DVD “Archivo ISO del almacenamiento de datos” y buscamos la iso almacenada.

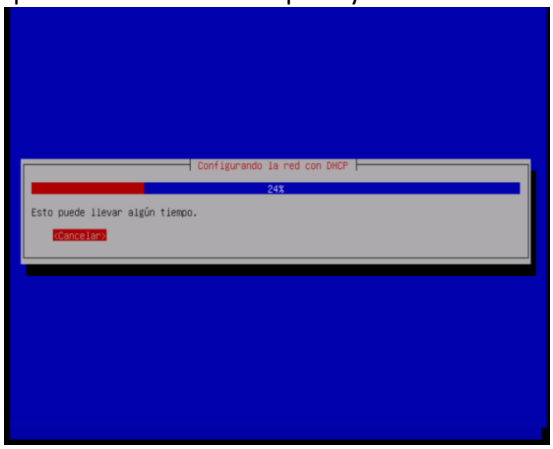
Y antes de finalizar, nos muestra un resumen de la configuración de la máquina.

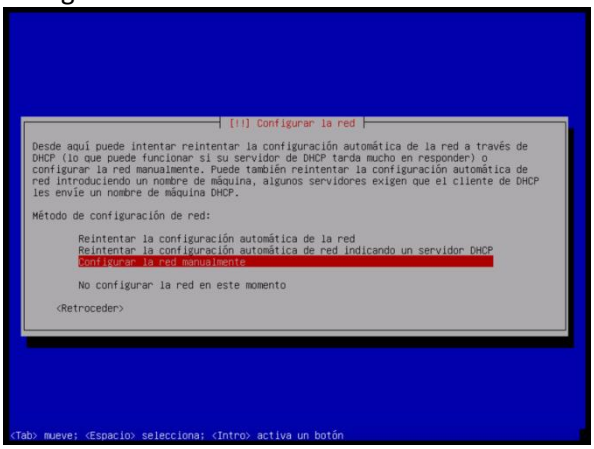
1.2 Instalación de SO Debían

Arrancamos la máquina virtual y nos aparece el siguiente menú y seleccionamos “Install”. Seleccionamos el lenguaje, país e idioma del teclado.

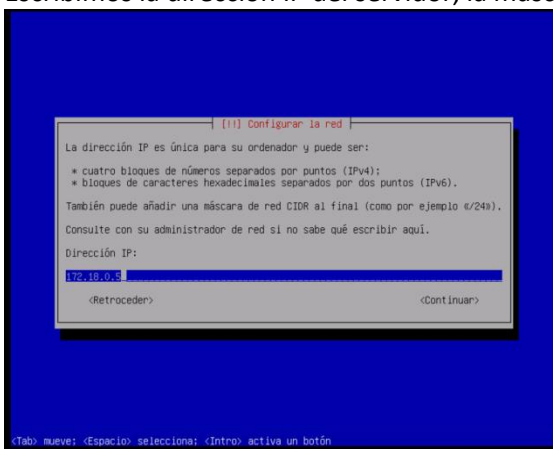


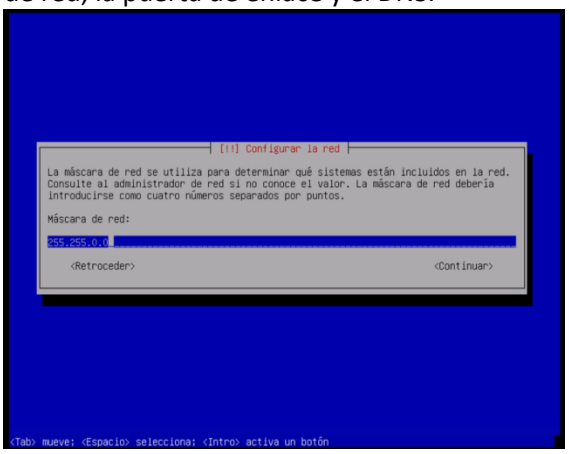
La red no la queremos por detección por DHCP, queremos escribirla de forma manual, así que cancelaremos este paso y Seleccionaremos “Configurar la red manualmente”.

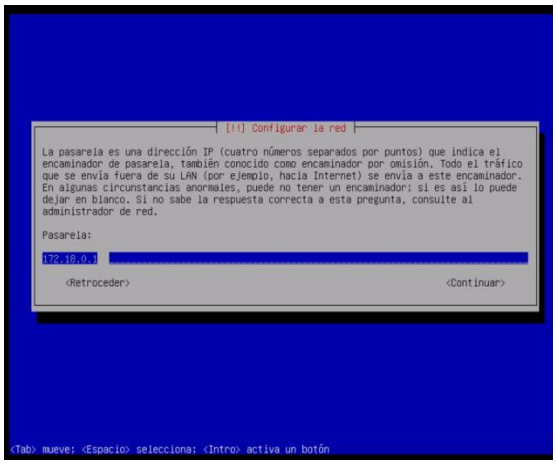


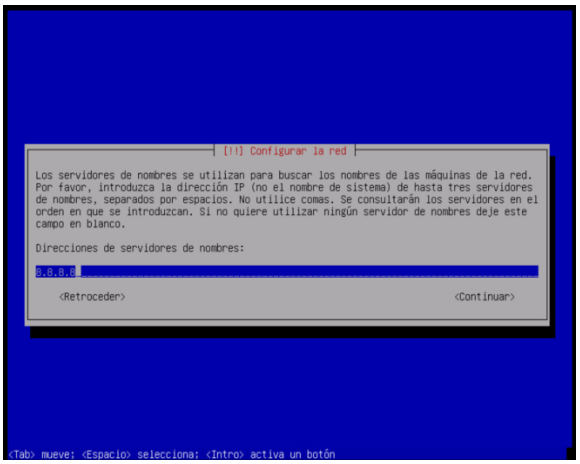


Escribimos la dirección IP del servidor, la máscara de red, la puerta de enlace y el DNS.

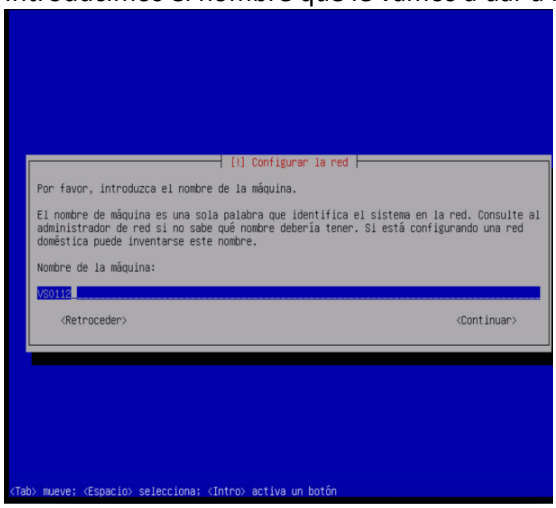


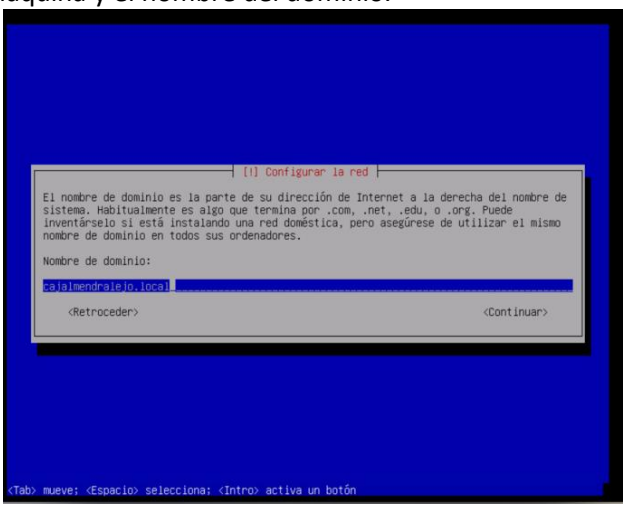




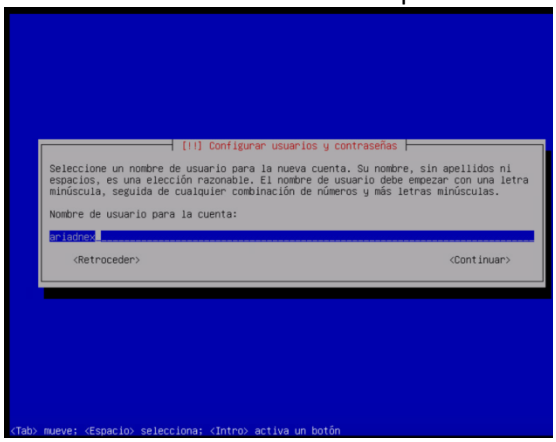


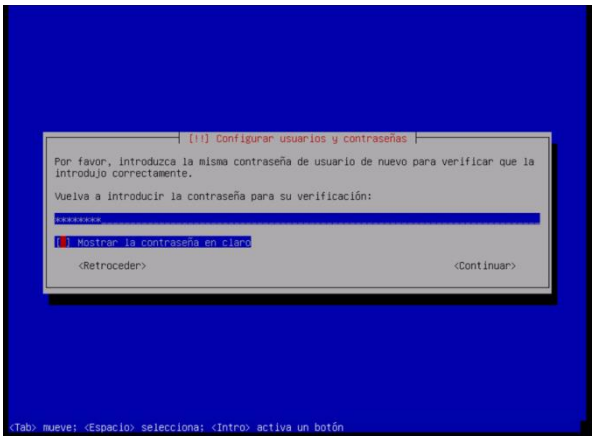
Introducimos el nombre que le vamos a dar a la máquina y el nombre del dominio.



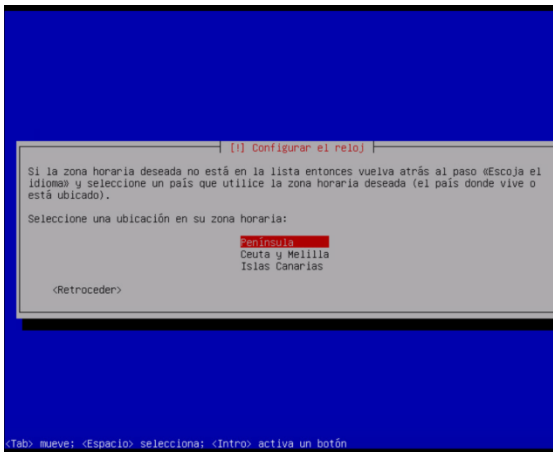


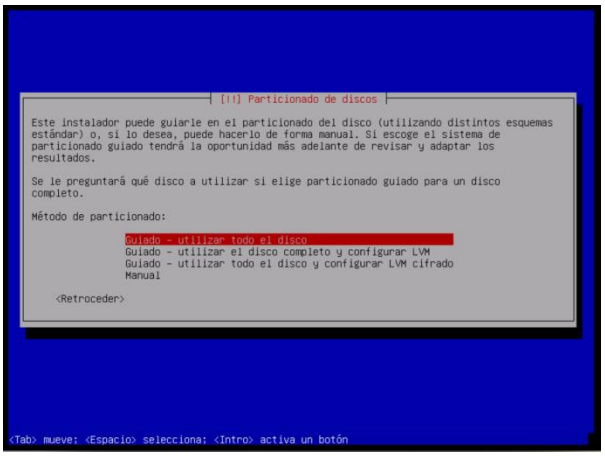
Escribimos el nombre de usuario para la cuenta y la contraseña.



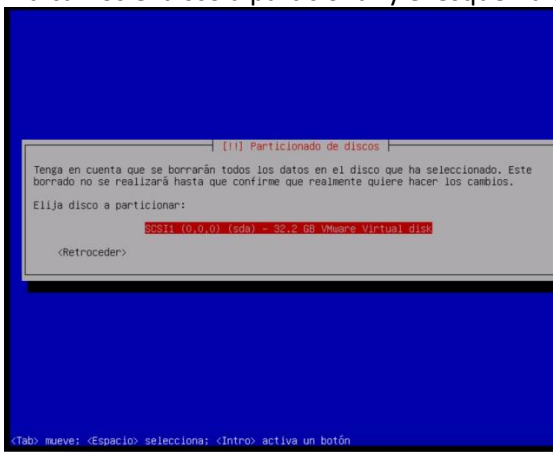


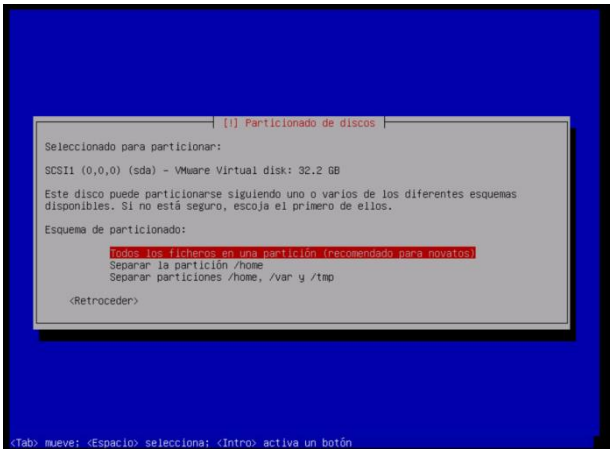
Seleccionamos la zona horaria y el método de particionado.



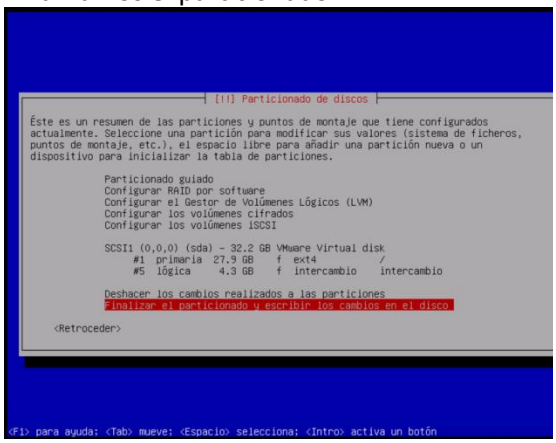


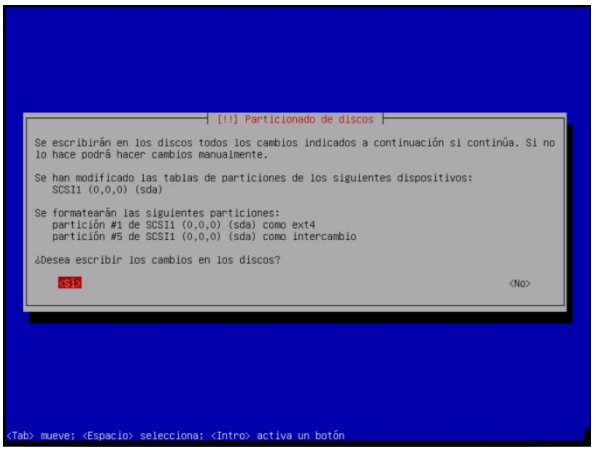
Indicamos el disco a particionar y el esquema de particionado.



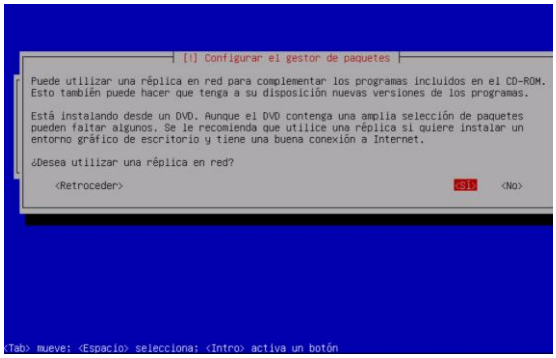


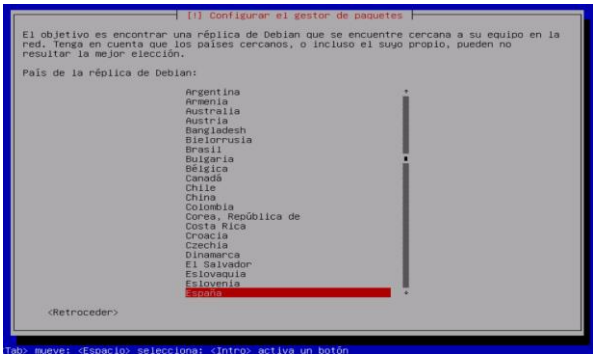
Finalizamos el particionado.



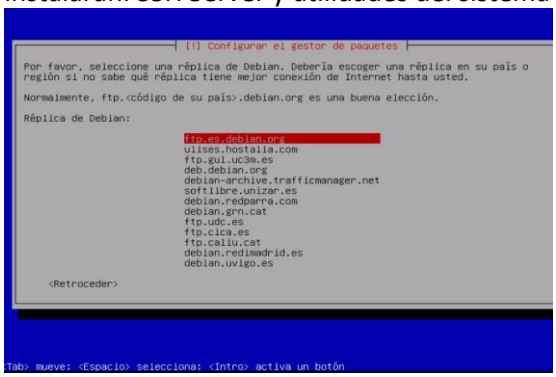


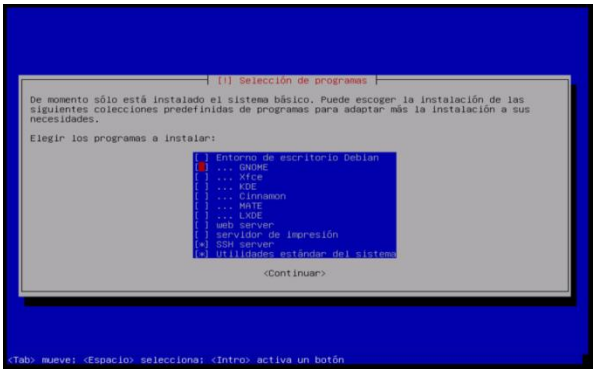
En la selección de una réplica en la red le indicamos que sí. Una vez instalado el sistema operativo, lo configuraremos manualmente. Primero seleccionamos el país de la réplica de debían.



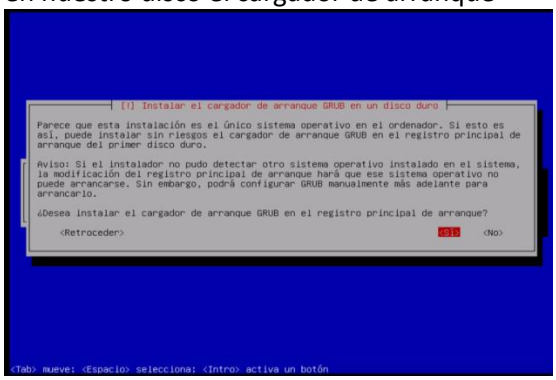


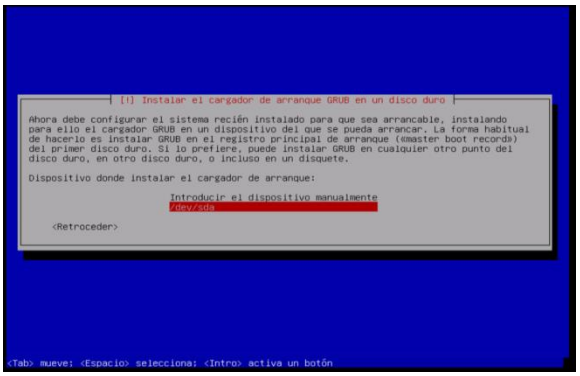
Escogemos una réplica de debían y seleccionamos los programas a instalar. En este caso se instalarán: SSH Server y utilidades del sistema



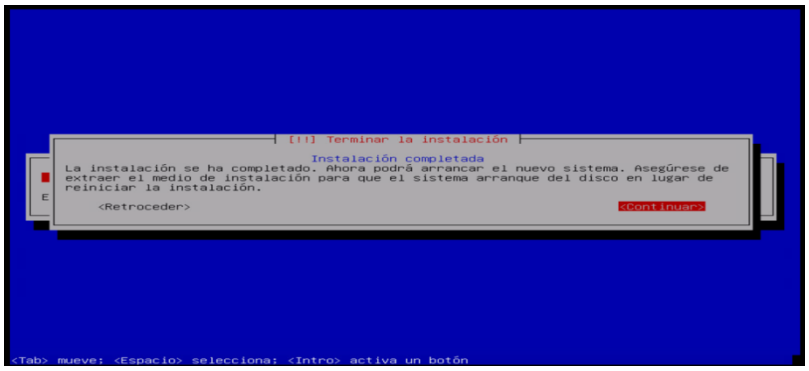


Instalamos el cargador de arranque en el registro principal del arranque y luego instalamos en nuestro disco el cargador de arranque





Por último, finalizamos la instalación.



2. Openvpn

2.1 Archivo de configuración serv.conf

```
#################################################
# Sample OpenVPN 2.0 config file for multi-client server.
# This file is for the server side of a many-clients <-> one-server OpenVPN configuration.
# OpenVPN also supports single-machine <-> single-machine configurations (See the Examples #page on the web site for more info). #
# This config should work on Windows or Linux/BSD systems. Remember on Windows to quote #pathnames and use double backslashes, e.g.: "C:\\Program Files\\OpenVPN\\config\\foo.key"
# Comments are preceded with '#' or ';'
#################################################
# Which local IP address should OpenVPN listen on? (Optional)
```

local 172.30.5.9
Which TCP/UDP port should OpenVPN listen on? If you want to run multiple OpenVPN #instances on the same machine, use a different port number for each one. You will need #to open up this port on your firewall.
port 1194
TCP or UDP server?
;proto tcp
proto udp
"dev tun" will create a routed IP tunnel,
"dev tap" will create an ethernet tunnel.
Use "dev tap0" if you are ethernet bridging
and have precreated a tap0 virtual interface
and bridged it with your ethernet interface.
If you want to control access policies
over the VPN, you must create firewall
rules for the the TUN/TAP interface.
On non-Windows systems, you can give
an explicit unit number, such as tun0.
On Windows, use "dev-node" for this.
On most systems, the VPN will not function
unless you partially or fully disable
the firewall for the TUN/TAP interface.
;dev tap
dev tun
Windows needs the TAP-Win32 adapter name
from the Network Connections panel if you
have more than one. On XP SP2 or higher,
from the Network Connections panel if you
have more than one. On XP SP2 or higher,
you may need to selectively disable the
Windows firewall for the TAP adapter.
Non-Windows systems usually don't need this.
;dev-node MyTap
SSL/TLS root certificate (ca), certificate
(cert), and private key (key). Each client
and the server must have their own cert and
key file. The server and all clients will
use the same ca file.
#
See the "easy-rsa" directory for a series
of scripts for generating RSA certificates
and private keys. Remember to use
a unique Common Name for the server
and each of the client certificates.
#
Any X509 key management system can be used.
OpenVPN can also use a PKCS #12 formatted key file
(see "pkcs12" directive in man page).
ca /etc/openvpn/ssl/ca.crt
cert /etc/openvpn/ssl/server.crt
key /etc/openvpn/ssl/server.key # This file should be kept secret

Diffie hellman parameters.
Generate your own with:
openssl dhparam -out dh2048.pem 2048
dh /etc/openvpn/ssl/dh2048.pem
Network topology
Should be subnet (addressing via IP)
unless Windows clients v2.0.9 and lower have to
be supported (then net30, i.e. a /30 per client)
Defaults to net30 (not recommended)
;topology subnet
Configure server mode and supply a VPN subnet
for OpenVPN to draw client addresses from.
The server will take 10.8.0.1 for itself,
the rest will be made available to clients.
Each client will be able to reach the server
on 10.8.0.1. Comment this line out if you are
ethernet bridging. See the man page for more info.
server 10.5.0.0 255.255.255.0
Static Client
client-config-dir ccd
Maintain a record of client <-> virtual IP address
associations in this file. If OpenVPN goes down or
is restarted, reconnecting clients can be assigned
the same virtual IP address from the pool that was
previously assigned.
ifconfig-pool-persist ipp.txt
Configure server mode for ethernet bridging.
You must first use your OS's bridging capability
to bridge the TAP interface with the ethernet
NIC interface. Then you must manually set the
IP/netmask on the bridge interface, here we
assume 10.8.0.4/255.255.255.0. Finally we
must set aside an IP range in this subnet
(start=10.8.0.50 end=10.8.0.100) to allocate
to connecting clients. Leave this line commented
out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100
Configure server mode for ethernet bridging
using a DHCP-proxy, where clients talk
to the OpenVPN server-side DHCP server
to receive their IP address allocation
and DNS server addresses. You must first use
your OS's bridging capability to bridge the TAP
interface with the ethernet NIC interface.
Note: this mode only works on clients (such as
Windows), where the client-side TAP adapter is
bound to a DHCP client.
;server-bridge
Push routes to the client to allow it
to reach other private subnets behind
the server. Remember that these

private subnets will also need
to know to route the OpenVPN client
address pool (10.8.0.0/255.255.255.0)
back to the OpenVPN server.
#push "route 192.168.1.0 255.255.255.0"
#push "route 192.168.2.0 255.255.255.0"
To assign specific IP addresses to specific
clients or if a connecting client has a private
subnet behind it that should also have VPN access,
use the subdirectory "ccd" for client-specific
configuration files (see man page for more info).
EXAMPLE: Suppose the client
having the certificate common name "Thelonious"
also has a small subnet behind his connecting
machine, such as 192.168.40.128/255.255.255.248.
First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
Then create a file ccd/Thelonious with this line:
iroute 192.168.40.128 255.255.255.248
This will allow Thelonious' private subnet to
access the VPN. This example will only work
if you are routing, not bridging, i.e. you are
using "dev tun" and "server" directives.
EXAMPLE: Suppose you want to give
Thelonious a fixed VPN IP address of 10.9.0.1.
First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
Then add this line to ccd/Thelonious:
ifconfig-push 10.9.0.1 10.9.0.2
Suppose that you want to enable different
firewall access policies for different groups
of clients. There are two methods:
(1) Run multiple OpenVPN daemons, one for each
group, and firewall the TUN/TAP interface
for each group/daemon appropriately.
(2) (Advanced) Create a script to dynamically
modify the firewall in response to access
from different clients. See man
page for more info on learn-address script.
;learn-address ./script
If enabled, this directive will configure
all clients to redirect their default
network gateway through the VPN, causing
all IP traffic such as web browsing and
and DNS lookups to go through the VPN
(The OpenVPN server machine may need to NAT
or bridge the TUN/TAP interface to the internet
in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

```
# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"
# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client
# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
duplicate-cn
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
#tls-auth ta.key 0 # This file is secret
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that 2.4 client/server will automatically
```

negotiate AES-256-GCM in TLS mode.
See also the ncp-cipher option in the manpage
cipher AES-256-CBC
Enable compression on the VPN link and push the
option to the client (2.4+ only, for earlier
versions see below)
;compress lz4-v2
;push "compress lz4-v2"
For compression compatible with older clients use comp-lzo
If you enable it here, you must also
enable it in the client config file.
comp-lzo
The maximum number of concurrently connected
clients we want to allow.
;max-clients 100
It's a good idea to reduce the OpenVPN
daemon's privileges after initialization.
#
You can uncomment this out on
non-Windows systems.
;user nobody
;group nogroup
The persist options will try to avoid
accessing certain resources on restart
that may no longer be accessible because
of the privilege downgrade.
persist-key
persist-tun
Output a short status file showing
current connections, truncated
and rewritten every minute.
status openvpn-status.log
By default, log messages will go to the syslog (or
on Windows, if running as a service, they will go to
the "\Program Files\OpenVPN\log" directory).
Use log or log-append to override this default.
"log" will truncate the log file on OpenVPN startup,
while "log-append" will append to it. Use one
or the other (but not both).
;log openvpn.log
;log-append openvpn.log
Set the appropriate level of log
file verbosity.
#
0 is silent, except for fatal errors
4 is reasonable for general usage
5 and 6 can help to debug connection problems
9 is extremely verbose
verb 3
Silence repeating messages. At most 20
sequential messages of the same message

category will be output to the log.
;mute 20
Notify the client that when the server restarts so it
can automatically reconnect.
explicit-exit-notify 1

2.2 Archivo de configuración client.conf

```
################################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.    #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.               #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension          #
################################################
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client
# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap
# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 141.136.59.251 1194
;remote my-server-2 1194
# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random
# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
```

resolv-retry infinite
Most clients don't need to bind to
a specific local port number.
nobind
Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nogroup
Try to preserve some state across restarts.
persist-key
persist-tun
If you are connecting through an
HTTP proxy to reach the actual OpenVPN
server, put the proxy server/IP and
port number here. See the man page
if your proxy server requires
authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]
Wireless networks often produce a lot
of duplicate packets. Set this flag
to silence duplicate packet warnings.
;mute-replay-warnings
SSL/TLS parms.
See the server config file for more
description. It's best to use
a separate .crt/.key file pair
for each client. A single ca
file can be used for all clients.
ca /etc/openvpn/ssl/ca.crt
cert /etc/openvpn/ssl/client.crt
key /etc/openvpn/ssl/client.key
Verify server certificate by checking that the
certicate has the correct key usage set.
This is an important precaution to protect against
a potential attack discussed here:
http://openvpn.net/howto.html#mitm
#
To use this feature, you will need to generate
your server certificates with the keyUsage set to
digitalSignature, keyEncipherment
and the extendedKeyUsage to
serverAuth
EasyRSA can do this for you.
remote-cert-tls server
If a tls-auth key is used on the server
then every client must also have the key.
#tls-auth ta.key 1
Select a cryptographic cipher.
If the cipher option is used on the server
then you must also specify it here.
Note that 2.4 client/server will automatically

negotiate AES-256-GCM in TLS mode.
See also the ncp-cipher option in the manpage
cipher AES-256-CBC
Enable compression on the VPN link.
Don't enable this unless it is also
enabled in the server config file.
comp-lzo
Set log file verbosity.
verb 3
Silence repeating messages
;mute 20

3. Archivo de configuración Elasticsearch

======================== Elasticsearch Configuration =========================
NOTE: Elasticsearch comes with reasonable defaults for most settings.
Before you set out to tweak and tune the configuration, make sure you
understand what are you trying to accomplish and the consequences.
The primary way of configuring a node is via this file. This template lists
the most important settings you may want to configure for a production cluster.
Please consult the documentation for further information on configuration options:
https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
---------------------------------- Cluster -----------------------------------
Use a descriptive name for your cluster:
#cluster.name: my-application
------------------------------------ Node ------------------------------------
Use a descriptive name for the node:
#node.name: node-1
Add custom attributes to the node:
#node.attr.rack: r1
---------------------------------- Paths ------------------------------------
Path to directory where to store the data (separate multiple locations by comma):
path.data: /var/lib/elasticsearch
Path to log files:
path.logs: /var/log/elasticsearch
---------------------------------- Memory -----------------------------------
Lock the memory on startup:
#bootstrap.memory_lock: true
Make sure that the heap size is set to about half the memory available
on the system and that the owner of the process is allowed to use this limit.
Elasticsearch performs poorly when the system is swapping the memory.
---------------------------------- Network -----------------------------------
Set the bind address to a specific IP (IPv4 or IPv6):
network.host: 10.5.0.1
Set a custom port for HTTP:
http.port: 9200
For more information, consult the network module documentation.
--------------------------------- Discovery ----------------------------------
Pass an initial list of hosts to perform discovery when new node is started:
The default list of hosts is ["127.0.0.1", "[::1]"]
#discovery.zen.ping.unicast.hosts: ["host1", "host2"]

Prevent the "split brain" by configuring the majority of nodes (total number of master-
#eligible nodes / 2 + 1):
#discovery.zen.minimum_master_nodes:
For more information, consult the zen discovery module documentation.
---------------------------------- Gateway -----------------------------------
Block initial recovery after a full cluster restart until N nodes are started:
#gateway.recover_after_nodes: 3
For more information, consult the gateway module documentation.
---------------------------------- Various -----------------------------------
Require explicit names when deleting indices:
#action.destructive_requires_name: true

4. Archivo de configuración de kibana.

Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601
Specifies the address to which the Kibana server will bind. IP addresses and host names
#are both valid values.
The default is 'localhost', which usually means remote machines will not be able to
#connect.
To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "172.30.5.9"
#server.host: "10.5.0.1"
Enables you to specify a path to mount Kibana at if you are running behind a proxy.
Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
from requests it receives, and to prevent a deprecation warning at startup.
This setting cannot end in a slash.
#server.basePath: ""
Specifies whether Kibana should rewrite requests that are prefixed with
`server.basePath` or require that they are rewritten by your reverse proxy.
This setting was effectively always `false` before Kibana 6.3 and will
default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false
The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576
The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"
The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://10.5.0.1:9200"
When this setting's value is true Kibana uses the hostname specified in the server.host
setting. When the value of this setting is false, Kibana uses the hostname of the host
that connects to this Kibana instance.
#elasticsearch.preserveHost: true
Kibana uses an index in Elasticsearch to store saved searches, visualizations and
dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"
The default application to load.
#kibana.defaultAppId: "home"
If your Elasticsearch is protected with basic authentication, these settings provide
the username and password that the Kibana server uses to perform maintenance on the
Kibana
index at startup. Your Kibana users still need to authenticate with Elasticsearch, which

```
# is proxied through the Kibana server.
#elasticsearch.username: "user"
#elasticsearch.password: "pass"
# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key
# Optional settings that provide the paths to the PEM-format SSL certificate and key files.
# These files validate that your Elasticsearch backend uses the same key files.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key
# Optional setting that enables you to specify a path to the PEM file for the certificate
# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]
# To disregard the validity of SSL certificates, change this setting's value to 'none'.
#elasticsearch.ssl.verificationMode: full
# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500

# Time in milliseconds to wait for responses from the back end or Elasticsearch. This value
# must be a positive integer.
#elasticsearch.requestTimeout: 30000
# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]
# Header names and values that are sent to Elasticsearch. Any custom headers cannot be
#overwritten
# by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist
configuration.
#elasticsearch.customHeaders: {}
# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to
#disable.
#elasticsearch.shardTimeout: 30000
# Time in milliseconds to wait for Elasticsearch at Kibana startup before retrying.
#elasticsearch.startupTimeout: 5000
# Logs queries sent to Elasticsearch. Requires logging.verbose set to true.
#elasticsearch.logQueries: false
# Specifies the path where Kibana creates the process ID file.
#pid.file: /var/run/kibana.pid
# Enables you specify a file where Kibana stores log output.
#logging.dest: stdout
# Set the value of this setting to true to suppress all logging output.
#logging.silent: false
# Set the value of this setting to true to suppress all logging output other than error
#messages.
#logging.quiet: false
# Set the value of this setting to true to log all events, including system usage information
# and all requests.
#logging.verbose: false
```

Set the interval in milliseconds to sample system and process performance
metrics. Minimum is 100ms. Defaults to 5000.
#ops.interval: 5000
The default locale. This locale can be used in certain circumstances to substitute any
#missing translations.
#i18n.defaultLocale: "en"

5. archivo de configuración filebeat

###################### Filebeat Configuration Example #########################
This file is an example configuration file highlighting only the most common
options. The filebeat.reference.yml file from the same directory contains all the
supported options with more comments. You can use it as a reference.
You can find the full configuration reference here:
https://www.elastic.co/guide/en/beats/filebeat/index.html
For more available modules and options, please see the filebeat.reference.yml sample
configuration file.

#========================== Filebeat inputs ============================
filebeat.inputs:
Each - is an input. Most options can be set at the input level, so
you can use different inputs for various configurations.
Below are the input specific configurations.
- type: log
  # Change to true to enable this input configuration.
  enabled: true
  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/suricata/eve.json
    #- c:\programdata\elasticsearch\logs*
Exclude lines. A list of regular expressions to match. It drops the lines that are
matching any regular expression from the list. exclude_lines: ['^DBG']
Include lines. A list of regular expressions to match. It exports the lines that are
matching any regular expression from the list.
#include_lines: ['^ERR', '^WARN']
Exclude files. A list of regular expressions to match. Filebeat drops the files that
are matching any regular expression from the list. By default, no files are dropped.
#exclude_files: ['.gz$']
Optional additional fields. These fields can be freely picked
to add additional information to the crawled log files for filtering fields:
level: debug
review: 1
Multiline options
Mutiline can be used for log messages spanning multiple lines. This is common
for Java Stack Traces or C-Line Continuation
The regexp Pattern that has to be matched. The example pattern matches all lines
#starting with [
#multiline.pattern: ^\[
Defines if the pattern set under pattern should be negated or not. Default is false.
#multiline.negate: false

Match can be set to "after" or "before". It is used to define if lines should be append to a #pattern
that was (not) matched before or after or as long as a pattern is not matched based on #negate.
#Note:After is the equivalent to previous and before is the equivalent to to next in Logstash
#multiline.match: after
#============================= Filebeat modules =======================
filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml
  # Set to true to enable config reloading
  reload.enabled: true
  # Period on which files under path should be checked for changes
  #reload.period: 10s
#==================== Elasticsearch template setting ==========================
setup.template.settings:
  index.number_of_shards: 3
  #index.codec: best_compression
#_source.enabled: false
#================================ General =============================
The name of the shipper that publishes the network data. It can be used to group
all the transactions sent by a single shipper in the web interface.
The tags of the shipper are included in their own field with each
transaction published.
#tags: ["service-X", "web-tier"]
Optional fields that you can specify to add additional information to the output.
#fields:
env: staging
#============================== Dashboards ============================
These settings control loading the sample dashboards to the Kibana index. Loading
the dashboards is disabled by default and can be enabled either by setting the
options here, or by using the `-setup` CLI flag or the `setup` command.
setup.dashboards.enabled: true
The URL from where to download the dashboards archive. By default this URL
has a value which is computed based on the Beat name and version. For released
versions, this URL points to the dashboard archive on the artifacts.elastic.co
website.
#setup.dashboards.url:
#============================== Kibana ====================================
Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
This requires a Kibana endpoint configuration.
setup.kibana:
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: **http://localhost:5601/path**
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "172.30.5.9:5601"
#============================ Elastic Cloud ==========================
These settings simplify using filebeat with the Elastic Cloud (https://cloud.elastic.co/).
The cloud.id setting overwrites the `output.elasticsearch.hosts` and

```
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:
# The cloud.auth setting overwrites the `output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth:
#================================ Outputs =============================
# Configure what output to use when sending the data collected by the beat.
#-------------------------- Elasticsearch output ------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
#  hosts: ["10.5.0.1:9200"]
#---------------------------- Logstash output --------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["10.5.0.1:5044"]

#================================ Logging ===================
# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
#logging.level: debug
# At debug level, you can selectively enable logging only for some components.
# To enable all selectors use ["*"]. Examples of other selectors are "beat",
# "publish", "service".
#logging.selectors: ["*"]

#============================== Xpack Monitoring ===================
# filebeat can export internal metrics to a central Elasticsearch monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.  The
# reporting is disabled by default.
# Set to true to enable the monitoring reporter.
#xpack.monitoring.enabled: false
# Uncomment to send the metrics to Elasticsearch. Most settings from the
# Elasticsearch output are accepted here as well. Any setting that is not set is
# automatically inherited from the Elasticsearch output configuration, so if you
# have the Elasticsearch output configured, you can simply uncomment the
# following line.
#xpack.monitoring.elasticsearch:
```

6. Archivo de configuración de Logstash.conf

```
input {
  beats {
    port => "5044"
    ssl  => false
  }
}
filter {

     json{
       source => "message"
     }
```

```
    geoip {
     source => "src_ip"
     target => "geoip"
     add_field => [ "[geoip][coordenates]", "%{[geoip][longitude]}" ]
     add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
    }
    mutate {
     convert => [ "[geoip][coordinates]", "float"]
    }
    geoip {
     source => "dest_ip"
     target => "geoip"
     add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
     add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
     }
     mutate {
    convert => [ "[geoip][coordinates]", "float"]
     }
}
output {
  elasticsearch {
    hosts => ["http://10.5.0.1:9200"]
    sniffing => true
    manage_template => false
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    document_type => "%{[@metadata][type]}"
   }
  stdout {
    codec => "rubydebug"
  }
}
```