Servidor dedicado de seguridad en contenedores de Linux Docker

Daniel Arjona Rivera

Resumen—Actualmente, estamos viviendo en un momento en el que la importancia de la seguridad informática está creciendo exponencialmente, en parte, debido a la gran subida del porcentaje de cibercrimen en los últimos meses. El sector empresarial y financiero como grandes afectados, han entrado en un estado de alerta en el cual la preocupación se ha hecho notar. El presente proyecto plantea una posible solución informática enfocada principalmente para aquellas personas que por su constante movildidad tengan que protegerse para acceder a Internet desde múltiples entornos, entre ellos puntos no seguros como pueden ser las redes inalámbricas públicas. También es aplicable su implantación en aquellas empresas que requieran una mejora en su nivel de seguridad y en la de sus trabajadores. Por ello, se ha creado un servidor dedicado de seguridad dotado de tecnología Docker utilizando el despliegue de contenedores en uno de los módulos y formado por distintas herramientas como VPN, Firewall, Antivirus, HIDS, Filtro-proxy Web, Log y un servicio de monitorización que permite dar al sistema un nivel de seguridad más que razonable.

Palabras clave—VPN, Firewall, Antivirus, HIDS, Log, DMZ, VM, Monitorización

Abstract— Currently, we are living in a time where the importance of computer security is growing exponentially, in part, due to the large increase in the percentage of cybercrime in recent months. The business and financial sector as major affected, have entered a state of alert in which the concern has been noted. The present project poses a possible computer solution focused mainly for those people whose constant mobility has access to the internet from multiple sites, among them unsafe points such as public wireless networks. Its implementation is also applicable in those companies that require an improvement in their level of safety and that of their workers.For this reason, a dedicated security server with Docker technology has been designed using container deployment in one of the modules and created with several tools such as VPN, Firewall, Antivirus, HIDS, and a monitoring service that give a good level of security.

Index Terms—VPN, Firewall, Antivirus, HIDS, Log, DMZ, VM, Monitoring

1 Introduction

El constante flujo de información que encontramos en Internet día a día mediante las millones de comunicaciones que se llevan a cabo, ya sean a nivel personal o empresarial, hace que alguno de los múltiples datos que circulan por la red muchas veces sean vulnerables a su obtención por terceros, ya que estos pueden hacer uso de prácticas maliciosas para sacar partido a las deficiencias de los distintos protocolos, servicios o sistemas entre otros. Por otra parte, la cantidad de malware que circula por Internet es abundante y es fácil acabar infectado por alguno de ellos de una manera u otra y más si no eres conocedor de las medidas de protección informática. Tal y como está la situación actual y cada vez más, se trata de un punto muy sensible y a tener en cuenta ya que que cada vez menos debería pasar desaparecibido. Hasta hace realmente muy poco tiempo, proteger los datos, sistemas o cualquier tipo de dispositivo era algo que se despreciaba y se le daba poco valor ya que las personas, en general, no estan concienciadas de ello y de la vital importancia que realmente tiene.

Dada la problemática mencionada, la empresa R3 Cybersecurity quiere poner a disposición de sus clientes más autónomos, o de empresas que quieran protección interna, un sistema de seguridad perimetral, formado por varios módulos, que les permita a estos, salir a Internet de una forma segura y con garantías de que su tráfico está protegido por el sistema. Por lo tanto, el proyecto ha tratado de desarrollarse y ajustarse a esta necesidad y dar una posible solución. Para ello, se han seleccionado una serie de herramientas que permiten dar una robustez y un nivel de seguridad alto conjuntamente con su correspondiente monitorización. Una de estas herramientas ha sido Docker, una tecnología muy reciente que automatiza el despliegue y paquetización de aplicaciones dentro de "contenedores virtuales" aprovechando únicamente las librerías y configuraciones del sistema operativo host. Otra de sus características es que permite el aislamiento de servicios y aplicaciones con requerimientos diferentes, con esto, además mejora la seguridad respecto a otros procedimientos de aislamiento ya que permite desplegar diferentes contenedores de forma ligera y por ello si una aplicación acaba siendo infectada o hackeada, el resto de contenedores o del

- *E-mail de contacto: Daniel.arjona@e-campus.uab.cat*
- *Mención realizada: Tecnologías de la Información*
- *Trabajo tutorizado por: Ramon Martí (Departamento de Ingeniería de la Informacion y de las Comunicaciones)*
- *Curso: 2016/17*

servidor difícilmente acabe afectándose.

Durante el presente documento se podrán encontrar las siguientes secciones: **Introducción, Objetivos, Estado del arte, Metodología y desarrollo, Test, Discusión de resultados, Conclusiones y Bibliografía.**

2 Objetivos

Los objetivos que se han definido para el proyecto son:

- Realizar la investigación y estudio de la las herramientas, tecnologías y productos disponibles actualemente y de la competencia.
- Diseño del prototipo del sistema.
- Implementar un acceso seguro al sistema mediante VPN + sistema de autenticación, HIDS (*Host Intrusion Detection System*) y Firewall.
- Implementar medidas de protección interna mediante Antivirus, Filtro-Proxy web, SNMP, servidor DNS.
- Introducir la monitorización y con ello poder obtener información útil de las herramientas del sistema más importantes.
- Ajustar el sistema de la mejor manera posible mediante los datos que se obtengan de la configuración inicial.

3 Estado del arte

En este apartado encontraremos la situación actual por la que pasan las distintas tecnologías más relevantes que se utilizarán en el proyecto así como las soluciones relacionadas que existen en este momento.

3.1 Herramientas de seguridad

Actualmente disponemos de diversidad de Antivirus de tipo OnAcces, es decir, que están constantemente analizando nuestros archivos y que nos permiten detectar las amenazas más significativas que podamos encontrarnos en la red de nuestro propio ordenador, es decir, de manera local. Lo mismo ocurre con los módulos Anti-spam para el caso del correo electrónico etc. Si queremos dar un paso más en la seguridad y juntar todo en un mismo producto para utilizarlo desde cualquier lugar que nos interese, nos tenemos que acercar a algún tipo de FireWall más profesional, como podría ser algunos de los productos que ofrece por ejemplo la empresa Sophos, los cuales incluyen conexión VPN de varios tipos, módulos anti-malware, anti-spam, filtro web entre otros servicios.

También podemos encontrar algunas soluciones que ofrecen servicios VPN con algún Firewall como mecanismo de protección, sin darnos muchas más especificaciones de la seguridad que podemos alcanzar ni de las medidas de protección que disponen sus servidores VPN. Algunas de ellas son Spotflux o ExpressVPN entre otras. Una VPN es una tecnología de red que permite una extensión segura de una red de área local (LAN) sobre una red pública o no controlada como Internet, en cambio un Firewall es un software o hardware que controla el acceso a diferentes niveles de una computadora a la red y de elementos de la red a la computadora por motivos de seguridad.

Por lo tanto, en vez de tener un FireWall hardware, o una simple VPN, el proyecto se enfoca a la creación de un sistema perimetral que incluya los módulos más importantes y herramientas de seguridad necesarias para que un cliente desde cualquier punto pueda salir a internet de forma segura utilizando como punto de entrada una VPN hacia nuestro sistema y además asegurar que nuestro servidor cuenta con los módulos de seguridad adecuados y una buena monitorización de estos para así tener un sistema completo.

3.2 Monitorización

Las soluciones de monitorización que nos encontramos hoy en día son múltiples, se vió Splunk, una herramienta referente en este campo, pero dado su compleja configuración, y la gran inversión que requiere, el proceso de implementación sería demasiado tedioso, por ello después de realizar una búsqueda y selección, llamó especialmente la atención el *stack* ELK.

Dicho módulo está compuesto por tres herramientas, Logstash [1], es la encargada de procesar los logs, es decir, extraer la información útil de ellos mediante el uso de filtros que modifican sus campos, ya que los mensajes de Logstash no son en formato de texto plano, sino que son objetos con campos tipados. Para complementar este módulo se utilizó Filebeat, una herramienta que permite el forwarding de logs de una máquina a otra, ya que por ejemplo, los logs del HIDS Ossec se requiere reenviarlos a Logstash al tener dichas herramientas aisladas en máquinas virtuales distintas.

La siguiente herramienta que lo compone es Elasticsearch [2], una vez procesados los logs pertinentes, son enviados a este módulo que su función es la de permitir indexar un gran volumen de datos y además que luego se permita hacer consultas sobre ellos en formato tipo JSON.

Por último contamos con la herramienta llamada Kibana [3], la cual mediante los distintos índices que dispone Elasticsearch permite extraer los logs en tiempo real y aparte crear distintos tipos de gráficos para una visualización más sencilla e intuitiva.

Estas herramientas son muy potentes ya que permiten la lectura, manejo, personalización, escalabilidad y monitorización de enormes cantidades de logs, cosa realmente interesante ya que ofrecen infinitas posibilidades para el tratamiento de este tipo de datos, su esquema simple de flujo de datos lo podemos ver en la *figura 1.*Otra de las alternativas de monitorización que se estudió fue otro sistema de monitorización llamado Nagios, analizando así

sus debilidades y fortalezas, pero dado que Kibana formaba parte del stack ELK, es sencillo de poner en marcha, y cumplía con todos los requisitos potenciales del sistema, finalmente se optó por Kibana.

Figura 1. Esquema básico del flujo de datos

5 Metodología y Desarrollo

En este apartado se verá la metodología que se ha utilizado a lo largo del presente proyecto y la explicación de las fases de desarrollo por las cuales ha pasado para llegar a cumplir los objetivos establecidos.

5.1 Metodología

El proyecto se ha realizado siguiendo una metodología de desarrollo Agile [7], una de sus características más importantes es la adaptabilidad a cualquier cambio por su proceso iterativo, cosa que importante teniendo en cuenta el proyecto que se va a realizado, ya que este, ha necesitado constantes iteraciones y así conseguir detectar/corregir errores lo antes posible antes de pasar a fases más avanzadas. Otra de sus características que se tuvieron en cuenta es su incrementalidad, cosa que ayuda a evolucionar el proyecto en base a los resultados completados en las anteriores iteraciones (Análisis, Diseño, Implementación, Test).

Al ser un proyecto realizado por una única persona, en algunas facetas se utilizó la metodología SCRUM [8] pero no al detalle, ya que la misma persona tendrá el rol tanto de Scrum Master como de desarrollador del proyecto.
El primer paso fué realizar una entrevista con el Product Owner para definir correctamente los objetivos fundamentales del proyecto. Cada iteración o sprint durará unas 4 semanas coincidiendo así con las entregas de los documentos pertinentes. Las reuniones con el Product Owner se realizarán semanalmente para tener un control de las entregas y revisar los posibles errores o modificaciones que sean necesarias.

5.2 Análisis

El primer paso fue el correspondiente análisis de requisitos realizado con la ayuda del Product Owner para seguidamente definir los requerimientos técnicos y funcionales del proyecto. Con ello, se pudo pasar a seleccionar las distintas herramientas de una forma más idónea acotando las opciones respecto a lo definido previamente.

5.2.2 Selección de herramientas

Una vez en este punto, se empezó a analizar las diferentes opciones que se disponían actualmente de cada herramienta, viendo así sus puntos positivos y negativos para ir haciendo una primera valoración e ir seleccionando las que más se ajustaban al proyecto.

En un primer momento se consiguió hacer un estudio del abanico de posibilidades que se disponía para elegir las distintas herramientas y posterior aprendizaje de las que han acabado siendo elegidas.

Después de seguir este procedimiento y una posterior prueba individual localmente de cada una de ellas, se obtubo así una idea global de estas para facilitar un punto más su elección. Con lo cual, se pudo tomar la decisión de seleccionar definitivamente las que iban a ser utilizadas en el proyecto. Por lo tanto, las herramientas seleccionadas y que han sido utilizadas son: VirtualBox, Docker, Ossec, SNMP, Diladele, OpenVPN, Iptables, DNSMasq, Filebeat, Elasticsearch, Logstash, Kibana, Rsyslog, y un S.O. Debian.

5.2.1 Firewall

En el entorno de trabajo donde se realiza el proyecto, se dispone de un dispositivo hardware Firewall del cual se tuvo que planificar la realización de una certificación *Online* oficial para la obtención de las nociones básicas y técnicas que permitieran su uso de manera segura. Tras este dispositivo corre gran parte de las configuraciones de red que se llevaron a cabo para la puesta a punto de la infraestructura y del proyecto en concreto, es decir, todo el diseño de la red (NAT's, reglas FW, interfaces, rutas, etc.) la cual cosa quiere decir que dominando su uso facilitó en las posteriores fases todo tipo de cambios y adaptaciones que se requirieron.

5.2.3 HIDS

En el sistema a diseñar, se ha visto necesario el uso de un *Host-based Intrusion Detection System*(HIDS) [9], para este módulo se ha seleccionado *Ossec* [10] ya que un punto a su favor es que es *Open Source* y además se pudo ver que dispone de grandes funcionalidades y características como el control de integridad de ficheros, el detector de rootkits, el control de integridad de registros, su constante actualización de las reglas implantadas y sus correspondientes alertas, que se adaptan a la perfección a las necesidades y requerimientos del proyecto. Esta herramienta ha sido probada integrándola en un contenedor Docker [11] y el resultado ha sido satisfactorio pero a la vez limitado, ya que se vió que solo se podía ejecutar en el caso que el agente estuviera situado en la interfaz donde estaba instalado Docker.

5.2.4 Filtro-Proxy web

El filtro-proxy web Diladele ha sido otra de las herramientas seleccionadas, también integrada en un contenedor Docker. Se trata de un filtro-proxy web que permite que las solicitudes y las respuestas de los distintos sitios remotos por los que navegue el cliente se autentiquen y redirijan al filtro ICAP [12] que tiene incorporado este módulo que se ejecutará de forma local o remota.

Esta herramienta también incorpora una interfaz web

para el administrador del entorno, con ella podremos filtrar el trafico *HTTPS* y *SSL*, bloquear sitios para adultos, categorizar diferentes sitios, bloquear archivos y Ads, Monitorizar el tráfico y generar informes. Estas son algunas de las características más significativas que serán vitales para dotar de una robustez y fiabilidad al entorno. Como punto no tan positivo encontramos que para utilizar Diladele disponemos de una versión de prueba de 2 meses, después hay que abonar la cantidad correspondiente a la licencia, que cabe decir que su precio no es elevado y es viable para el proyecto.

5.2.5 Docker

El uso de Docker, una tecnología de virtualización de código abierto basada en la automatización del despliegue de aplicaciones dentro de contenedores, lo cual permite la creación, las pruebas y la configuración de estas aplicaciones de una forma rápida, ligera y portable, es añadirle al sistema un punto de innovación dándole así una vuelta más intentando utilizar alguno de los módulos necesarios dentro de esta tecnología. En el caso que el sistema crezca, con Docker, es realmente fácil migrar sus imagenes previamente configuradas. Como se ha comentado, el despliegue de aplicaciones en Docker es realmente ligero, por ello es una buena opción en nuestro sistema, un caso práctico sería cuando el proxy se sature de peticiones se podría realizar un balanceo de carga y desplegar un segundo contenedor a través de la misma imagen sin afectar en el rendimiento del sistema utilizando un mínimo de recursos extra. Integrar alguno de los módulos en una tecnología tan reciente como Docker, suponen un nuevo reto y ha tenido por contra una mayor inversión de tiempo en conocer su funcionamiento y sus características principales.

5.2.6 Antivirus

Como Antivirus/Antimalware se ha seleccionado *Sophos*, ya que también se trata de una herramienta OpenSource para entornos Linux. Su modo de configuración ha sido *OnAcces*, es decir, que esté analizando constantemente todo el tráfico que pasa por la máquina virtual donde está instalado, así como los archivos que se manejan en todo momento. Se ha probado tratando diferentes tipos de archivos “maliciosos” y ha reaccionado de una manera satisfactoria.

5.2.7 SNMP

El protocolo SNMP (Simple Network Manager Protocol), es basado en un conjunto de normas enfocadas a la gestión de la red y que utiliza servicios ofrecidos por TCP/IP. Está compuesto por dos elementos principales, el agente, que es un programa que ha de ejecutarse en los nodos de red que se deseen gestionar y monitorizar, y el manager, que suele ejecutarse en la estación donde se monitoriza la red y su faena es la de realizar consultas sobre los agentes instalados para obtener datos sobre ellos.En este caso, se ha utilizado un SNMP Manager y un agente SNMP.

Un punto importante a mencionar es el mensaje llamado *trap* de este protocolo, que permite a un agente enviar datos al Manager que no han sido solicitados directamente por el, asi como informar de eventos relacionados con errores, caídas del servicio, sobre carga en la CPU, memoria, etc.

Para el manejo de datos en este protocolo se utiliza un estándar de información llamado MIB (Management Information Base). Es el conjunto de información organizada jerárquicamente y que mantiene los datos de un dispositivo de red gestionado agrupados por una serie de categorías, así como las operaciones que están permitidas.

Categoría	Información
system	Información del host del sistema de encaminamiento
interfaces	Información de los interfaces de red
addr-translation	Información de traducción de direcciones
ip	Información sobre el protocolo IP
icmp	Información sobre el protocolo ICMP
tcp	Información sobre el protocolo TCP
udp	Información sobre el protocolo UDP
egp	Información sobre el protocolo (Exterior Gateway)

Figura 2. Categorias disponibles de la MIB.

5.2.8 Rsyslog

Para recibir los logs aprovechando que se disponía de un S.O Debian, se decidió implementar un servidor centralizado Rsyslog ya que es open-source y utiliza el estándar Syslog. Se trata de un sistema de procesamiento de registros de sistema rápido y eficiente que permite diversas configuraciones, la introducción de templates personalizados y obtenerlos de tantos órigenes como sea necesario, además, una vez aprendes a manejarlo es cómodo de administrar. Se puede utilizar tanto en entornos empresariales grandes como en los más reducidos gracias a su versatilidad y robustez.

5.2.9 Anti-Spam

En el caso del módulo Anti-Spam, en un primer momento se ha probado la herramienta *Spamassassin* [13], basada en el reconocimiento automático de spam analizando los correos entrantes siguiendo ciertas reglas definidas en su configuración, por ello cumple los requerimientos necesarios del proyecto.

En cambio se ha podido ver que conjuntamente con el módulo que también incorpora el dispositivo Hardware Firewall del entorno no acababa de sacar todo el partido a su potencial, la cual cosa significó que no contáramos con dicha herramienta ya que con una buena configuración del módulo que dispone el Firewall para ello, funciona a la perfección y actúa de manera óptima en caso de tener que utilizar un servidor mail propio, en nuestro caso como el servicio será ofrecido para clientes externos no se implementará dicho módulo ya que no se requiere la existencia

de un servidor mail centralizado teniendo en cuenta que cada futuro cliente usará un servidor mail distinto.

5.2.10 Monitorización

La elección de uso del stack ELK fue basada en la documentación que se obtuvo durante la investigación de herramientas disponibles, donde se pudo ver que actualmente forman un módulo muy potente, open-source y las cuales permiten una infinidad de posibilidades como se ha comentado anterioremente y que se ajustan a la perfección al proyecto en cuestión. Dado los requerimientos, fue una buena decisión poner en marcha estas herramientas y en concreto separarlas en una VM diferente para tener el sistema bien estructurado y aislado.

5.2.11 VPN

Como punto de acceso al sistema la opción decidida fue mediante el uso de una VPN, para ello la herramienta que se eligió fue OpenVPN, se trata de una solución basada en software libre (GPL) que permite la conectividad punto a punto con validación de usuarios y host concetados remotamente, además soporta un gran número de configuraciones y opciones interesantes como por ejemplo, balanceo de carga, introducir los servidores DNS para su tráfico etc. Otro de los factores que se tuvo en cuenta fue la gran documentación que se puede encontrar sobre dicha herramienta.

5.2.12 DNS

Dado que el número de peticiones DNS que se realizarán en el sistema será elevado, se intentó aumentar las medidas de seguridad lo máximo posible, por ello, para que nadie consiga capturar las peticiones DNS que se lleven a cabo en el entorno, se ha elegido DNSMasq para canalizarlas a través del túnel del servidor OpenVPN y así que viajen de forma cifrada y sea este servidor el encargado de resolverlas.

5.2.13 Log Forwarder

Debido a que los logs generados por las distintas herramientas no estarán situados en las máquinas donde serán tratados para la monitorización o almacenados, se requería una herramienta para facilitar este *forwarding* de los distintos logs. Al leer la documentación de el stack ELK se vió que utilizaban una herramienta llamada Filebeat la cual realizaba esa función para dichas herramientas. Por ello se comprobó que leyendo su configuración y realizándola correctamente su funcionamiento se ajustaba a los requerimientos del sistema. Permite la lectura de ficheros dada una cierta ruta, y su envío hacia Elasticsearch o Logstash según nuestra necesidad.

5.2.14 Iptables

Aprovechando el uso de un S.O Linux, se decidió que la mejor opción para interceptar, manipular y establecer reglas sobre los distintos paquetes de red sería Iptables, además era una herramienta que ya conocía su funcionamiento y era consciente de su buen rendimiento para estos casos.

5.2 Diseño

Tras analizar y ver todo lo que se disponía tras el Firewall local, se decidió crear una zona DMZ, es decir, una zona segura exclusivamente para dicho proyecto, aislada de la LAN donde están el resto de ordenadores y de la WLAN donde acceden los dispositivos que se conecten vía WiFi. Con esta configuración conseguíamos darle un punto más de seguridad y control al sistema ya que podremos controlar el tráfico específico que entre y salga de la DMZ teniendo así un control de acceso para las personas permitidas y evitando facilitar que en un posible caso de ataque se propague al resto de redes locales.

Por lo tanto, dichas herramientas serán implantadas siguiendo el diseño de la *figura 2*, es decir, se aislará dicho único servidor físico en la DMZ mencionada y se crearán tres máquinas virtuales con VirtualBox. Para la distribución de IP's se configuró el modo de red Bridge en cada una de ellas, asignadoles así una IP del rango 192.168.25.0/24, que es el propio de la DMZ donde esta situado el servidor físico.

La primera VM será nombrada como Escudo dado que dispondrá de todas las herramientas de protección y será el punto de acceso al sistema, la segunda será el Almacén, ya que será la encargada de recibir las alertas y los logs de las distintas herramientas y en caso necesario de reenviarlos. La tercera será el Monitor, su uso será exclusivamente el de tratar los logs y monitorizarlos.

En la VM Escudo se implantará un agente Ossec, un agente SNMP, el AntiVirus Linux Sophos, Docker y un contenedor Docker donde correrá el filtro-proxy web.

En el Almacen, se configurará un servidor Rsyslog, el Manager SNMP, el servidor Ossec y Filebeat.

En la Monitor se implantará el stack ELK, es decir, Logstash, Elasticsearch y Kibana, encargados de la pertinente monitorización.

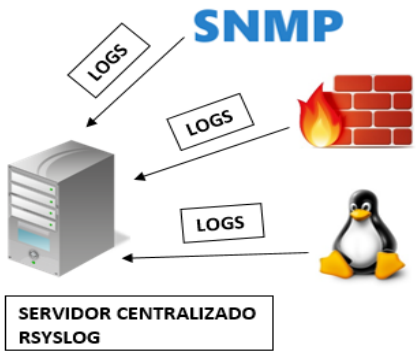

Figura 4. Esquema básico de un Rsyslog recibiendo logs de diferentes tipos de fuentes.

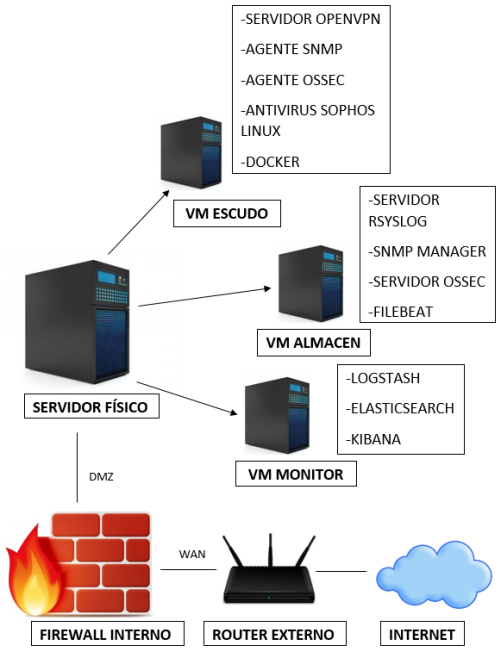

Figura 2. Diseño de la DMZ del proyecto.

5.3 Desarrollo

Durante esta última fase del proyecto, se estudió de qué manera se podía crear un prototipo que sirviese como entorno de pruebas para tener una aproximación lo más real posible a lo que será el producto final y pasar a la implementación de todo el sistema. La elección fue aprovechar el potente servidor que se dispone en el entorno de trabajo donde se está realizando el proyecto y así crear varias máquinas virtuales donde encontraremos lo siguiente:

5.3.1 El Escudo

Al Escudo se le ha instalado un S.O Debian, conteniendo así la pertinente instalación de Docker, que se encargará de ejecutar el contenedor del filtro-proxy Web.

También contará con la configuración de una VPN utilizando OpenVPN [14], la configuración de un agente Ossec y otro SNMP.

Como se ha mencionado, esta primera VM dispone de la instalación de un servidor VPN utilizando OpenVPN, este ha sido configurado con lo siguiente: A más a más de los certificados básicos que tiene que tener este servidor OpenVPN se le ha implementado el uso de TLS, es decir, un protocolo que permite y garantiza el intercambio de datos en un entorno securizado, añadiéndole una nueva clave que nos permite agregar soporte para usar la autenticación TLS y con ello fortificar la seguridad de dicho servidor. La clave generada servirá para introducir una firma HMAC (Mecanismo de autenticación de mensajes para proteger su integridad) en todas las transacciones que se realicen mediante el protocolo de handshake o intercambio de claves SSL/TLS entre el cliente y el servidor, de esta forma podremos verificar la integridad de los paquetes intercambiados entre el cliente y el servidor VPN viendo así si un cliente intenta conectarse a este servidor VPN sin poseer la clave para firmar los paquetes, la conexión se rechazará de forma automática.

Otro aspecto interesante ha sido el uso de DNSMASQ, una herramienta que permite desplegar un servidor DNS de forma sencilla, en este caso con el objetivo de que que nadie consiga capturar nuestras peticiones DNS. Lo que se ha realizado con ello ha sido canalizar la totalidad de estas peticiones a través del túnel del servidor OpenVPN y así dichas peticiones DNS que se hagan utilizando esta conexión, se enviaran a este mismo servidor OpenVPN de forma cifrada y será el servidor OpenVPN el encargado de resolverlas mediante esta herramienta.

También cabe destacar el uso del plugin de PAM (Pluggable Authentication Module) utilizado para darle un paso más al entorno e introducir la existencia de la autenticación mediante usuario y contraseña en OpenVPN. Cada vez que queramos que un nuevo usuario use el sistema, se le creará su fichero de configuración con los pertinentes certificados y claves asignándole un nombre y contraseña para que más tarde el inicie sesión usando dichas credenciales asignadas.

Por último para acabar de complementar todo este entorno, se ha definido el parámetro de configuración del servidor OpenVPN "redirectgateway" para que todo el tráfico de red IP originado por los distintos clientes pase a través de dicho servidor.

El módulo Ossec, como en el apartado test se ve explicado, no se ha encontrado su funcionamiento óptimo desplegándolo en un contenedor Docker dados los requisitos que necesitamos en nuestro sistema, por tanto, se ha hecho lo siguiente: En primer lugar se ha instalado el agente Ossec en esta primera máquina virtual ya que será donde interesa sacar y monitorizar la información de las distintas alertas de seguridad que se puedan producir. Por ello, en esta VM se ha implementado en el fichero /etc/rsyslog.con la creación de un nuevo módulo que vaya leyendo el fichero de logs que crea Ossec situado en la carpeta donde lo hayamos instalado, en este caso en /var/ossecagente/log/alerts/alerts.log.

El paso siguiente ha sido la creación de la configuración necesaria para enviar estos logs, por tanto creamos un nuevo archivo independiente llamado logs_ossec.conf en /etc/rsyslog.d donde establecemos la severidad de los logs que queremos enviar, en este caso 7, que es el nivel de depuración (debuggin) y la ip:puerto del servidor Rsyslog, que es la VM Almacen el destino en cuestión. Con esto ya podemos enviarlos a medida que se generen y además en el Syslog destino se creará una nueva carpeta con el nombre o identificador de nuestro sistema origen y se creará el fichero syslog.log donde finalmente se almacenan estas alertas en forma de log.

En la configuración de este agente Ossec, inicialmente se introdujo la clave generada por la parte del Servidor Ossec (situado en la *VM Almacen*), donde esta es creada añadiendo la IP e ID del agente que quieres añadir, después se definió también el uso del Syscheck, es decir, la activación del demonio de control de integridad que dispone esta herramienta y que permite reportar los cambios producidos en el sistema de ficheros como vemos en la *figura 3*, al ser modificado un fichero, su hash ya no es el mismo y Ossec lo detecta correctamente dando una alerta.

Otra de las características potenciales de esta herramienta es su sistema de reglas que dispone, del cual se hablará en la fase de test.

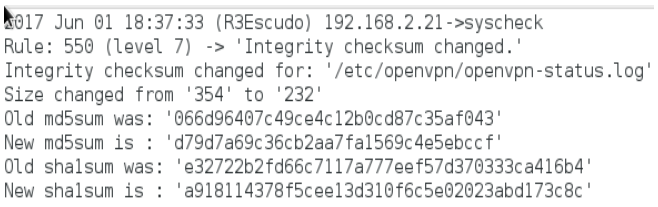

Figura 3. Alerta enviada por el agente Ossec al cambar el checksum de un cierto log monitorizado.

Por parte del filtro-proxy Web, su despliegue ha sido mediante un contenedor Docker utilizando el comando requerido para ello, una vez se dispone del contenedor activo podemos acceder a su interfaz Web por localhost en el puerto 8000. En este punto se vió una multitud de opciones y configuraciones en las cuales se puede restringir el uso de todo tipo de dominios, URL's, bloquear IP's, limitar el acceso a diferentes tipos de categorías de webs, monitorizar el tráfico y ver los reports, bloquear ciertos archivos, entre muchas otras opciones. Con todo esto se estableció una configuración personalizada que permitiese probar cada una de estas características importantes de la herramienta en la posterior fase de test. Al utilizar un contenedor para desplegar esta herramienta, Docker permite acceder al interior de cada uno de ellos que se esté ejecutando y abrir una Shell del mismo, lo que se ha implementado aprovechando esto ha sido la personalización de la página HTML que muestra el proxy cuando salta una de las reglas aplicadas restrictivas, informando así al usuario de la prohibición de acceso en esa Web o dominio dándole un toque personal al entorno.

Como AntiVirus base, dispondrá de lo ya confirmado anteriormente, Sophos Linux, que se configuró para que actuara de modo OnAcces filtrando todo el tráfico que pase por esta VM de manera continua.

Por último, también tendrá implementado un agente SNMP, configurado para que escuche por todas las interfaces que disponga por el puerto 161 y con una política rocommunity public localnetwork, para que cualquier sistema de la red local que disponga de SNMP instalado pueda realizar operaciones de lectura en él, es decir, consultar la MIB, que es la base de datos jerarquizada en forma de árbol que se dispone en este protocolo. Focalizando, en concreto en el Manager, se le permitirá además de estas operaciones de lectura, realizar operaciones de Escritura mediante una política rwcommunity private IP-Manager, ya que será quien administre a dicho agente.

5.3.2 El Almacen

En el Almacen con otro S.O Debian instalado, encontramos el SNMP Manager mencionado en el punto anterior, configurado para que también se encargue de recibir las traps que se hayan configurado en el agente y así poder obtener información básica y útil del estado del servidor, como podría ser cuando un directorio determinado se llene más de un tanto por ciento, cuando el uso de la CPU sobre pase un porcentaje definido, o cuando el propio agente SNMP se caiga, reinicie o se inicie. Estos parámetros son configurables en el fichero situado en /etc/snmp/snmpd.conf. Dichas traps, están configuradas en formato Syslog para que estas también sean procesadas posteriormente en el servidor Rsyslog de forma adecuada, además se ha implementado que a medida que vayan llegando a este, se vayan guardando en un nuevo fichero de log llamado snmptraps.log para facilitar las fases siguientes por las que pasarán estos datos y serán procesados.

Esta segunda VM mencionada, tambien cuenta con la implementación de un servidor Rsyslog escuchando por el puerto 514 UDP por el cual irá recibiendo los diferentes logs de los módulos pertinentes. Para facilitar el aislamiento y saber el origen de los datos que nos llegan, se ha definido un template en la configuración de este Syslog server para que automáticamente cuando le llega un log de un nuevo sistema, IP o aplicación se cree una carpeta con el nombre del mismo conociendo así dicho origen (según consiga extraer) y dentro de esa carpeta un fichero llamado syslog.log donde irá guardando todos estos logs que le vayan llegando. Con esto conseguimos dar independencia y facilidad en los ficheros de logs a tratar. El firewall físico hardware también se ha configurado para que todos sus logs los envíe a este servidor Rsyslog introduciéndole así la IP, puerto del mismo y el tipo de nivel de logs que queremos obtener. Por último necesitabamos una herramienta encargada de hacer el forwarding de los logs entre esta máquina virtual y la tercera que es donde se encuentra todas las herramientas de procesamiento y monitorización de los logs. Para ello se utilizó Filebeat, configurándolo para que envie los logs pertinentes a la IP:PUERTO donde recae la instalación de Logstash. Una vez con toda esta configuración e implementación anterior lista, es hora de empezar a tratar los logs y es entonces cuando entra en acción el stack ELK instalado en la última máquina virtual.

5.3.3 El Monitor

En el Monitor la primera herramienta por la que pasarán los distintos logs será Logstash. Desde Logstash lo que se ha hecho ha sido analizar los diferentes tipos de logs tal cual llegan de los módulos y ver si hay algún campo a extraer o alguna forma de personalizar sus campos para analizarlos más fácilmente. Por ejemplo, en los que llegan del Firewall físico. Para estos se ha creado un nuevo fichero de configuración de Logstash donde le indicamos en el input la ruta del log donde leer, que empiece a hacerlo desde el principio del fichero, después en el apartado de filtrado, le aplicamos dos modificaciones para que se creen nuevos campos en dicho log que nos han resultado interesantes como puede ser su origen y timestamp, finalmente en la parte de output le indicamos hacia donde serán enviados después de este procesamiento, en este caso hacia Elasticsearch a su puerto de escucha y en el índice donde se almacenarán. Esta es una configuración que hay que tener para cada tipo de log distinto adaptada a las necesidades del mismo para obtener así logs más intuitivos y fáciles de monitorizar.

Una vez la configuración para cada tipo de log haya sido generada e implementada, los logs como se ha dicho,

a medida que van siendo procesados son enviados a Elasticsearch donde serán indexados dada la configuración que se le ha establecido, en nuestro caso contaremos con un único nodo en nuestro clúster que creará un shard por cada índice generado, es decir, cada índice estará entero en el mismo servidor y aparte tendremos una réplica de cada uno como backup. En caso de que en un futuro quisiéramos añadir otro nodo (servidor) al clúster, sería totalmente posible ya que es una herramienta totalmente escalable y permitiría así el balanceo de carga de los distintos shards en caso de caída de un nodo como vemos en un ejemplo en la *figura 5*.

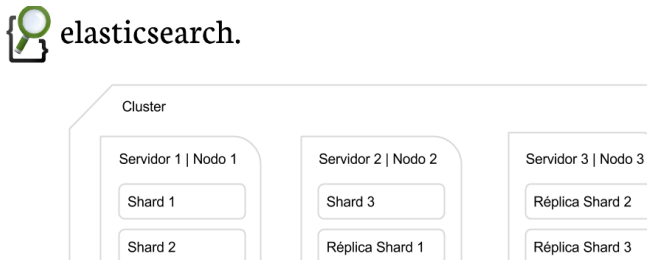

Figura 5. Esquema de un clúster Elasticsearch compuesto por tres nodos con tres shards y una réplica de cada uno de ellos.

Por último, al tener ya todos los logs indexados correctamente en Elasticsearch, es el momento de monitorizarlos mediante Kibana. Accederemos a dicha herramienta mediante su puerto 5601 configurado para acceder a través de localhost. Una vez en Kibana, se han introducido los distintos índices creados con anterioridad para que se pueda acceder a ellos y le empiecen a llegar dichos datos ordenados por timestamp. Con ello, se han creado distintos Dashboards para obtener gráficos más visuales y que nos permita por ejemplo en el caso de Ossec, ver las alertas recibidas por su control de integridad como se puede apreciar en la *figura A2* del apéndice.

En este módulo se han tenido que crear los diferentes tipos de gráficos adaptados a las necesidades de cada tipo de log y herramienta, ya que para todos no queremos ver el mismo tipo de información ni de la misma manera, ya sea por el factor temporal, o por el formato del gráfico. Dada toda esta explicación, podemos ver un ejemplo más visual del funcionamiento completo del sistema con uno de los módulos incluidos, Ossec en la *figura 6.*

En relación a esta figura con nuestro sistema, como se ha explicado anteriormente, disponemos inicialmente, de un agente Ossec situado en el Escudo, el cual irá enviando las alertas generadas en formato log al Servidor Ossec situado en el Almacen. Dicho Servidor Ossec irá enviando estas alertas recibidas al Servidor Syslog donde se creará el correspondiente fichero de logs independiente de este módulo. Una vez generado este, será leído por Logstash el cual lo irá procesando según la configuración establecida para este tipo de logs para que posteriormente sean indexados e almacenados por Elasticsearch. Una vez correctamente indexados, el proceso llega a su fase final permitiendo a Kibana la realización de consultas a este índice creado y su correspondiente monitorización en tiempo real de los logs que vayan llegando creando así gráficos útiles adaptados al tipo de log, en este caso alertas por minuto es un tipo de información significativa, podemos ver un ejemplo en la figura A4 del apéndice.

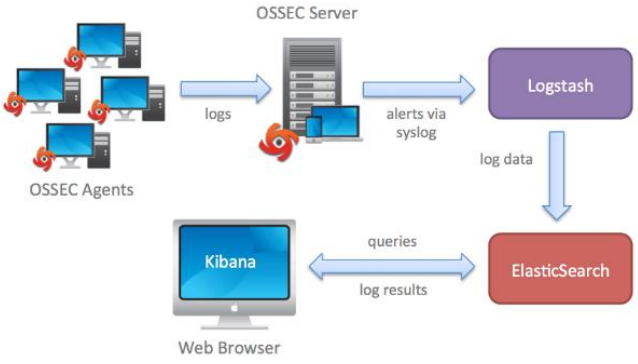

Figura 6. Esquema de funcionamiento del módulo Ossec con monitorización.

6 Test

Una primera parte de la fase de test ha ido siendo realizada a medida que la fase de desarrollo del proyecto avanzaba y se iban introduciendo nuevas herramientas, características o mejoras y estas eran testeadas utilizando inicialmente pruebas unitarias y posteriormente de integración mediante entornos de prueba controlados, es decir, se realizaban *snapshots* en las VM donde se trabajaba, puntos de restauración en el tiempo para revertir a un estado anterior dicha VM. Para ello se realizó un documento específico de casos de prueba el cual había que ir siguiendo y así poder comprobar si el sistema y cada una de las herramientas iba cumpliendo los requisitos técnicos y funcionales que fueron definidos.

La segunda parte de la fase de test consistió en proporcionar unas cuentas de prueba de OpenVPN para poder acceder y utilizar el sistema a un grupo reducido de usuarios para verificar la usabilidad, compatibilidad con los diferentes dispositivos, rendimiento y un nuevo análisis de las características mediante la obtención del feedback que nos proporcionen.

Con ambas partes del la fase de test, se vió que en cuanto a las características y configuraciones introducidas, se han cumplido los requerimientos establecidos inicialmente y las necesidades presentadas por los potenciales usuarios que utilizarán un sistema como este.

7 Discusión de resultados

En este apartado se expondrán los resultados que se han podido obtener hasta el momento, gran parte de ellos obtenidos por la correspondiente fase de test explicada anteriormente.

Teniendo en cuenta los objetivos planificados se ha podido realizar la exhaustiva investigación y estudio de la selección de herramientas y tecnologías disponibles obteniendo con ello un nivel de conocimiento que ha permitdo seleccionar dichas herramientas de la manera más acertada

posible. Para el diseño del prototipo, se ha visto que al utilizar una DMZ para aislar el servidor físico y sus tres VM's permite incrementar la seguridad y el control del entorno ya que si se hubiera implementado todo en una misma máquina, al caer esta o ser infectada, todo el sistema se veria comprometido.

A partir del sistema diseñado, el prototipo permite dar un acceso seguro a los usuarios al sistema mediante la creación de su certificado y clave para el uso de la VPN utilizando OpenVPN. Para su uso, es obligatorio el inicio de sesión de cada usuario del sistema utilizando el nombre de usuario y contraseña asignados. Una vez el usuario se conecta correctamente a la VPN, mediante Wireshark, se ha comprobado que su tráfico en su totalidad y en especial las peticiones DNS, como se aprecia en la *figura 7* se realizan al Servidor OpenVPN, es decir, viaja por el túnel que se crea con tal conexión y es cuando los distintos módulos entran en funcionamiento.

Si el usuario intenta navegar por una web no permitida por la configuración del filtro proxy-web, como medida de protección se le muestra un mensaje de aviso personalizado explicando tal acontecimiento y se le bloquea el acceso. Por otra parte, si cualquier potencial atacante intenta hacer un ataque de fuerza bruta por ejemplo al servicio SSH, se ha podido ver que el HIDS implantado Ossec envía una alerta (*figura A2*) que será visualizada en el fichero de log correspondiente y, además será monitorizada a través de Kibana como el resto de alertas y logs en general.

Otro de los resultados interesantes que se han podido ver ha Sido en otra de las alertas de Ossec, viendo como cuando se abre un puerto en dicho sistema, el output del comando *"netsat -tan"* es lógicamente distinto y la herramienta lo notifica como podremos ver en la *figura A1* situada en el apéndice.

Una de las variables finales del proyecto que se ha podido comprobar a través de las pruebas ha sido que, al utilizar Ossec a través de un contenedor de Docker, tan solo se puede utilizar el agente en la interfaz de Docker del sistema donde ha sido instalado. Este factor no es muy óptimo ya que en caso de que el sistema creciese nos interesaría monitorizar otros sistemas, con lo cual, la mejor opción es la de realizar la instalación del agente Ossec en la *VM Escudo* e implementar el servidor Ossec centralizado en el *Almacen* y así tener la opción de enfocar el sistema para ser escalable y poder asignar tantos agentes monitorizables como sean necesarios.

Como parte de las medidas de protección interna se ha visto un posible caso crítico donde un potencial atacante o cualquier administrador del sistema de seguridad, descargue un archivo infectado o potencialmente sospechoso, entonces es cuando el AntiVirus configurado, dará una alerta mostrando un mensaje de aviso y bloqueará dicho archivo, se ha probado ejecutar desde el simple EicarTest hasta la descarga de varios tipos de malware y lo detecta a la perfección.

Una prueba básica que se ha podido observar por parte del Agente SNMP es que, cuando reiniciamos, apagamos o iniciamos dicho agente, por defecto, se envía una *trap* al SNMP Manager viendo así la correcta conexión y configuración que se establece entre ambos.

Uno de los módulos que hasta realizar la fase de test no se había podido comprobar sus limitaciones, es el de *anti-spam.*En este módulo se ha comprobado que para su correcto funcionamiento debe contar con un servidor *relay* de correo dedicado donde realizar sus filtros, reglas y operaciones pertinentes para detectar el posible spam que llegue en este tipo de tráfico. Por lo tanto, esta solución dado nuestro proyecto donde un cliente puede acceder a su correo personal desde cualquier navegador/servidor *SMTP* por ejemplo, no es válida y se ha decidido prescindir de dicho módulo y esperar para futuras mejoras de los productos actuales que permitan incorporar alguna solución adaptada a las necesidades de nuestro sistema.

Como vemos en las *figuras A4* y *A5* del anexo, se le ha conseguido implantar al sistema un intuitivo sistema de monitorización mediante la configuración del stack ELK y en concreto con la herramienta Kibana, la que permite visualizar datos muy útiles y valiosos de todo lo que esta ocurriendo en los diferentes módulos monitorizados.

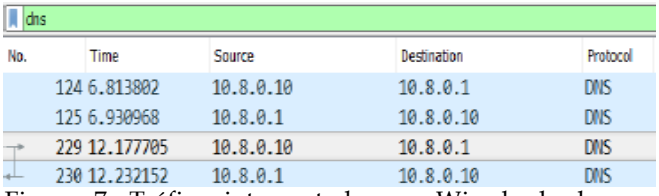

Figura 7. Tráfico interceptado por Wireshark de una petción DNS realizada a través de la VPN del servidor implementado.

7 Conclusión

Durante el desarrollo del presente proyecto y su diseño, el sistema final realizado, ha podido dar una posible solución a la problemática actual anteriormente comentada respecto a la deficiencia existente en el ámbito de la ciberseguridad. En líneas generales, se puede afirmar que, el sistema está en completo funcionamiento excepto algunos pequeños test y ajustes que quedan por realizar de cara a la puesta en producción para acabar de optimizar el sistema y dejar un proyecto consistente en su totalidad para los clientes.

Estos ajustes recaen por ejemplo en el tipo de restricciones a establecer por parte del filtro proxy-web, mejorar un punto más el formato de los logs de forma más legible y sencillo con Logstash, configurar algunas de las traps necesarias que debería enviar de más el SNMP así como seguir añadiendo algunas reglas a implementar en Ossec para evitar alertas innecesarias o poco relevantes que no

dan información realmente interesante del sistema monitorizado.

Finalmente, se ha visto que para todos aquellos usuarios que constantemente esten saliendo a internet desde redes no seguras o para los que deseen implantar dicho sistema en una empresa para su uso interno, este sistema ofrece unas características técnicas y funcionales que permiten mejorar la seguridad en la red y tener un buen nivel de monitorización consiguiendo así evitar en un mayor grado las posibles ciber amenazas que se puedan presentar.

Como futura implementación, se ha pensado añadirle un mecanismo de alertas basado en *machine learning* el cual permita obtener información aún más sospechosa del sistema mediante la búsqueda de patrones de los logs, sistema de ficheros distintos tipos de tráfico. Un posible ejemplo significativo podría ser el siguiente: si vemos que en un log que nos llega, el campo de los bytes enviados son mayores a una cierta cantidad y que además la petición previa ha sido realizada desde USA, pues podría indicarnos la posibilidad de que algún potencial atacante nos está extrayendo nuestros datos y así poder revisar dicha incidencia para solucionarla en el remoto caso que sea cierta. Este mecanismo sería darle una vuelta más a nuestro sistema y dotarle de una mayor inteligencia, cada vez más necesaria dada las necesidades actuales donde la seguridad cada vez está más solicitada y requerida en los sistemas, por otra parte, los antivirus que solo detectan malware por firma, cada vez estarán mas obsoletos.

Otra medida evolutiva del proyecto es ir migrando las herramientas utilizadas que se vean adecuadas a su despliegue en contenedores Docker ya que facilitarían la migración de estos a otros entornos una vez se tienen bien configurado, ya que muchas de estas herramientas tienen ciertas limitaciones en Docker actualmente según el uso que se le vaya a dar.

AGRADECIMIENTOS

Me gustaría agradecer a mi tutor, Ramon Martí, la ayuda y el soporte que me ha proporcionado dándome una visión crítica para mejorar los informes que han sido presentados

También me gustaría agradecer el apoyo recibido por mi familia y mi pareja desde el primer momento.

REFERENCIAS

[1] “Logstash” [En línea].Disponible: https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html

[2] ”Elasticsearch: The Definitive Guide” [En línea].Disponible: https://www.elastic.co/guide/en/elasticsearch/guide/current/intro.html

[3] “How to use Kibana Dashboards and Visualizations” [En línea].Disponible: https://www.digitalocean.com/community/tutorials/how-to-use-kibana-dashboards-and-visualizations

[4] ”Analiza la viabilidad de tu idea” [En línea].Disponible: http://www.emprendedores.es/crear-una-empresa/idea-negocio-viable/negocio-viable-5

[5] ”Viabilidad económica” [En línea].Disponible: https://es.wikipedia.org/wiki/Viabilidad_econ%C3%B3mica

[6] ”Factibilidad del proyecto empresarial“ [En línea].Disponible: http://www.decoop.cl/Inicio/FomentoCooperativo/CursosenL%C3%ADnea/FACTIBILIDADDELPROYECTOEMPRESARIAL/tabid/130/Default.aspx

[7] “Agile, un nuevo marco de trabajo para el desarrollo de proyectos” [En línea].Disponible: http://www.rosallop.com/blog/agile-un-nuevo-marco-de-trabajo-para-el-desarrollo-proyectual/#sthash.H81mUTVN.dpbs

[8] ”Proceso y roles de Scrum” [En línea].Disponible: https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html

[9] ”Administración de Sistemas Operativos” [Enlínea].Disponible: http://www.adminso.es/index.php/4.2.5._IDS_basados_en_host_(HIDS)

[10] “Welcome to Ossec’s documentation” [En línea].Disponible: http://ossec-docs.readthedocs.io/en/latest/

[11] ”About Docker” [En línea].Disponible: http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

[12] ”Internet Content Adaptation Protocol” [En línea].Disponible: https://en.wikipedia.org/wiki/Internet_Content_Adaptation_Protocol

[13] “Spamassassin: Deshaciéndonos del SPAM”[En línea].Disponible: https://www.elserver.com/spamassassin-deshaciendonos-del-spam/

[14] “How to OpenVPN” [En línea].Disponible: https://openvpn.net/index.php/open-source/documentation/howto.html

APÉNDICE

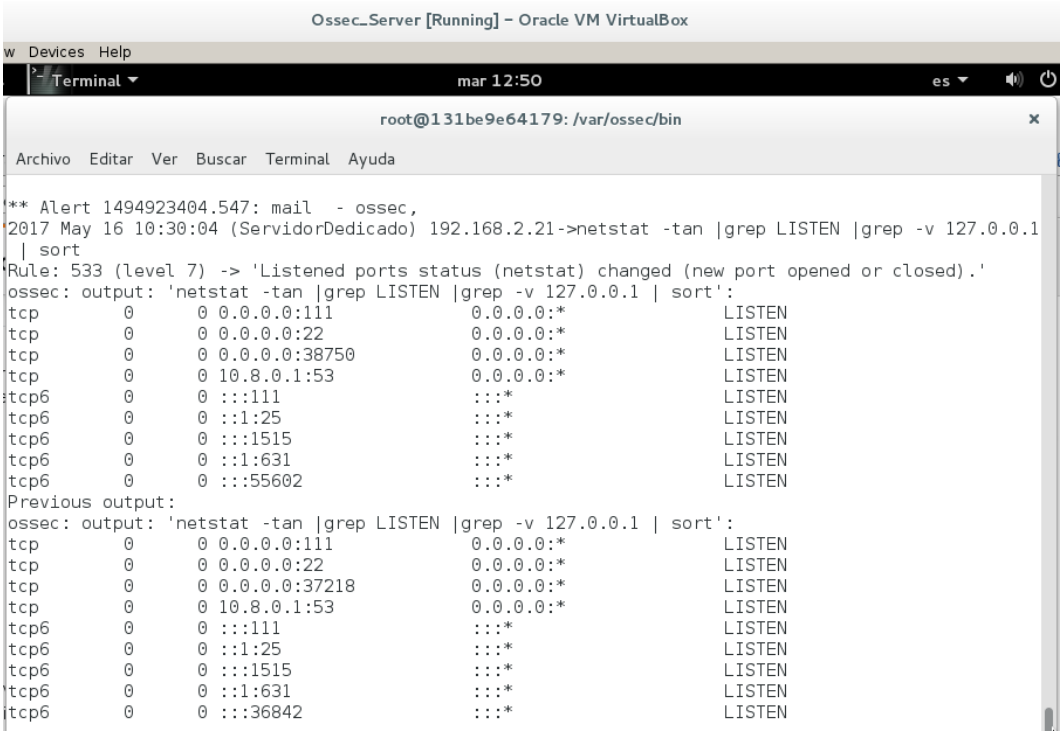

Figura A1. Alerta enviada por el agente Ossec al cambar el output del comando "netstat -tan" y notificando de que un nuevo puerto ha podido ser abierto o cerrado.

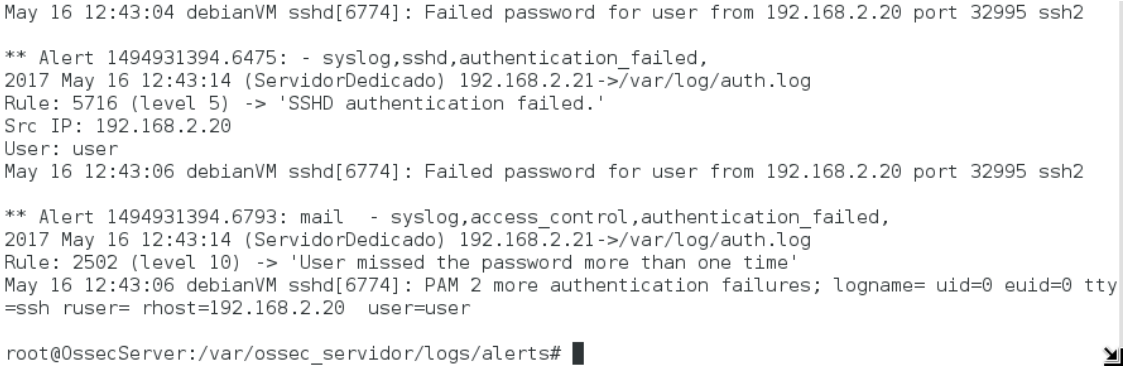

Figura A2. Alerta enviada por el agente Ossec al ver que un usuario está fallando las credenciales de acceso al servicio ssh.

```
root@R3Syslog:/var/log# cat snmptrapd.log
NET-SNMP version 5.7.2.1 AgentX subagent connected
NET-SNMP version 5.7.2.1
2017-05-17 15:29:42 192.168.2.21(via UDP: [192.168.2.21]:41040->[192.168.2.22]:1
62) TRAP, SNMP v1, community private
        NET-SNMP-MIB::netSnmpNotificationPrefix Enterprise Specific Trap (NET-SN
MP-AGENT-MIB::nsNotifyShutdown) Uptime: 0:16:44.63

2017-05-17 15:29:42 <UNKNOWN> [UDP: [192.168.2.21]:38297->[192.168.2.22]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (100463) 0:16:44.63    SNMPv2-M
IB::snmpTrapOID.0 = OID: NET-SNMP-AGENT-MIB::nsNotifyShutdown   SNMPv2-MIB::snmp
TrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpNotificationPrefix
2017-05-17 15:29:43 192.168.2.21(via UDP: [192.168.2.21]:58972->[192.168.2.22]:1
62) TRAP, SNMP v1, community private
        NET-SNMP-MIB::netSnmpAgentOIDs.10 Cold Start Trap (0) Uptime: 0:00:00.08

2017-05-17 15:29:43 <UNKNOWN> [UDP: [192.168.2.21]:34543->[192.168.2.22]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (8) 0:00:00.08 SNMPv2-MIB::snmp
TrapOID.0 = OID: SNMPv2-MIB::coldStart  SNMPv2-MIB::snmpTrapEnterprise.0 = OID:
NET-SNMP-MIB::netSnmpAgentOIDs.10
root@R3Syslog:/var/log#
```

Figura A3. Trap enviada por el agente SNMP notificando su conexión seguidamente de un reinicio.

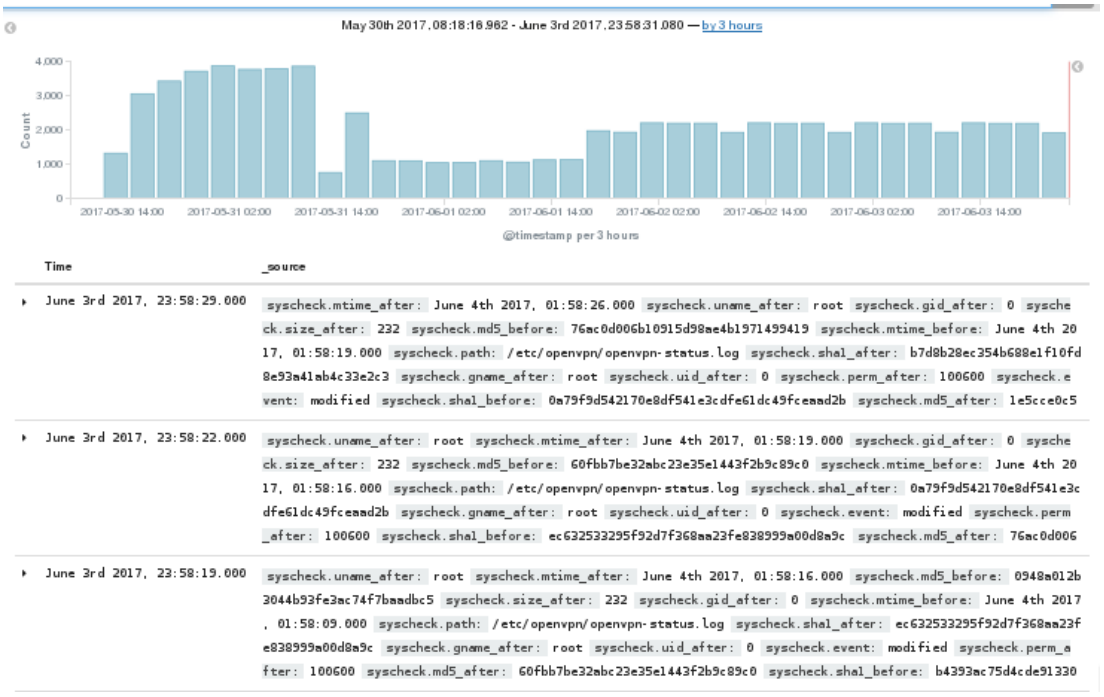

Figura A4. Dashboard de Kibana mostrando la cantidad de alertas recibidas y su correspondiente información.

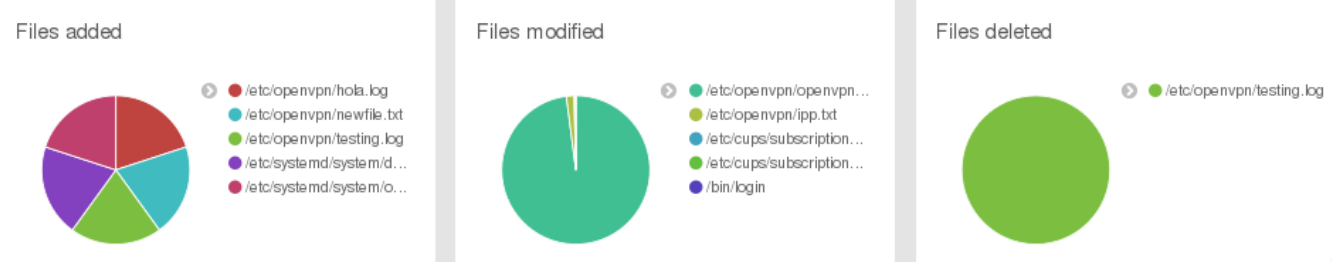

Figura A5. Dashboard de Kibana mostrando las alertas de Ossec recibidas por los últimos archivos añadidos, modificados y eliminados.