Universitat Oberta de Catalunya
uoc.edu

Análisis de actividades sospechosas en la red

Álvaro Mira Carrillo
Master Interuniversitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Seguridad Empresarial

Borja Guaita Pérez
Victor García Font

Fecha de entrega: 13 de junio de 2024

FICHA DEL TRABAJO FINAL

Título del trabajo:	Análisis de actividades sospechosas en la red
Nombre del autor:	Álvaro Mira Carrillo
Nombre del consultor/a:	Borja Guaita Pérez
Nombre del PRA:	Víctor García Font
Fecha de entrega (mm/aaaa):	06/2024
Titulación:	Máster Interuniversitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Área del Trabajo Final:	Seguridad Empresarial
Idioma del trabajo:	Castellano
Palabras clave	Teletrabajo, NIDS, SIEM

Resumen del Trabajo

El objetivo de este trabajo es analizar los riesgos que puede suponer la red doméstica de un teletrabajador en la empresa para la que teletrabaja.
Para realizar este estudio, se hará uso de herramientas NIDS y SIEM que serán instaladas y configuradas dentro de una red doméstica, de este modo se podrá monitorizar el tráfico generado por todos los equipos de la red para posteriormente comparar con comportamientos asociados a problemas de seguridad, y en última instancia revisar si realmente lo son.
Las herramientas elegidas han sido Suricata como NIDS y la pila Elastic como SIEM. Tras analizar el tráfico de la red doméstica durante un periodo, se ha podido concluir que no hay amenazas en la red, aunque requiere un tiempo de aprendizaje y perfeccionamiento de la solución para poder reducir la cantidad de "falsos positivos" que se crean con las configuraciones mínimas del entorno.

Abstract

The aim of this work is to analyze the risks that a remote worker's home network may pose to the company for which they work remotely.
To carry out this study, NIDS and SIEM tools will be installed and configured within a home network. This will make it possible to monitor all the traffic generated by the devices on the remote worker's network, to compare it with behaviors associated with security issues, and finally to review if such issues are really present.
The chosen software were Suricata as NIDS and Elastic stack as SIEM. After analyzing the traffic of the home network during a time period, it has been concluded that there are no threats within the network, although this requires a time to learn and improve the solution in order to reduce the number of "false positives" that are generated with the software's minimum configuration.

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Contexto y justificación del Trabajo

Durante la crisis sanitaria, muchas empresas se vieron forzadas a posibilitar que sus trabajadores pudiesen seguir trabajando desde casa, por lo que se requirió habilitar el acceso a los servicios en red de la empresa desde fuera de ésta, servicios que muchas veces no estaban implementados en el día a día de las empresas. Esto ha supuesto un momento ideal para que los ataques a empresas aumenten, ya que en algunos casos los departamentos de IT han tenido que ofrecer estos servicios de trabajo remoto, sin tener mucha experiencia previa sobre como implementar estos servicios. Cuando no existía la necesidad de trabajo remoto, era más fácil controlar el acceso a la información y servicios de la empresa, ya que bastaba con filtrar el acceso a los servicios desde los rangos de red privados de la empresa.

Al habilitar el acceso externo a los recursos, surgen dos nuevos problemas de seguridad que pueden poner en riesgo la información. Por un lado, delimitar qué intentos de acceso externo son legítimos, ya que vendrán desde Internet, sin poder saber a priori la IP pública con la que el trabajador remoto intentará acceder.

Por otro lado, está el hecho de las redes domésticas de los teletrabajadores, ya que es una red no supervisada por la empresa, pudiendo poner en riesgo la confidencialidad en el intercambio de información entre el ordenador del trabajador remoto y los servicios de la empresa, y si a esto le sumamos que los teletrabajadores puede que hagan uso de sus propios equipos personales, surge un gran reto a la hora de implementar las políticas de seguridad, ya que al hacer uso de la VPN, estamos dando acceso a los recursos internos de la red desde un equipo, que puede que pase o no por el Firewall de la empresa cuando quiera acceder a recursos que estén en Internet (dependiendo

de la solución VPN implementada), poniendo en grave riesgo la integridad y confidencialidad de la información de la empresa.

En este proyecto lo que se pretende es analizar es las posibles amenazas que pueden existir en un entorno doméstico, y que por consiguiente, puedan afectar a los servicios o recursos de una empresa a través de los empleados que realizan teletrabajo.

Para poder realizar este análisis, se ha decidido implementar una solución NIDS+SIEM que permitará monitorizar todo el tráfico sospechoso que se produce desde "casa". La herramienta NIDS permitirá registrar el tráfico que se produce dentro de la red, para posteriormente contrastar este tipo de tráfico con posibles comportamientos asociados a ataques. Además de un NIDS es necesario utilizar una herramienta que permita analizar de manera estructurada todas las alertas generadas por NIDS, para ello se hará uso de una herramienta SIEM.

1.2. Objetivos del trabajo

El objetivo de este trabajo es analizar cuáles pueden ser los posibles riesgos de la red domestica de un teletrabajador para su empresa. Para realizar dicho análisis, se instalará una herramienta que permitirá monitorizar el tráfico que se produce en la red, y con la información obtenida se comparará con posibles amenazas de seguridad conocidas (NIDS), los resultados de esta comparativa se enviarán a una herramienta que facilite el análisis de los datos (SIEM). Se valorarán qué herramientas existen actualmente para ambas funcionalidades, para posteriormente implementar la solución que más se ajuste al objetivo de este trabajo.

1.3. Impacto en sostenibilidad, ético-social y de diversidad del trabajo

En este trabajo no existe un impacto en sostenibilidad, ni ético-social, ni de diversidad del trabajo. En cuanto a sostenibilidad, el resultado de este trabajo está orientado a hacer un estudio sobre los posibles riesgos de seguridad que puede suponer la red doméstica de un teletrabajador para una empresa, por lo que las herramientas y equipamiento utilizado para el trabajo, será única y exclusivamente para el estudio, y que permita hacer una extrapolación a una visión global, sin implementar más herramientas ni equipamiento en los entornos de los teletrabajadores.

Tampoco existe un impacto ético-social ni de diversidad del trabajo, ya que el resultado de este trabajo no está condicionado por ninguna de éstas, al estar puramente orientado a usuarios teletrabajadores (como individuo), sin que exista ninguna diferenciación en el resultado en función de ninguno de los rasgos éticos, sociales o de género.

1.4. Enfoque y método seguido

El enfoque de este proyecto es analizar que confianza se puede tener sobre la red local de un trabajador remoto, en cuanto a la seguridad de la información de los datos de la empresa para la que teletrabaja. Para poder conocer el grado de confianza se instalará una solución NIDS+SIEM, que ayudará a detectar si existe tráfico sospechoso. Tras haber realizado un estudio sobre las posibles soluciones que existen, se seleccionarán las herramientas y equipamiento que tengan el menor coste posibles, intentando en la medida de lo posible hacer uso de los recursos previos que se tengan, y a la hora de seleccionar herramientas de software se priorizarán aquellas que sean gratuitas. También se tendrá en cuenta la facilidad de conectar las herramientas elegidas para cada tipo. Al ser un ámbito nuevo para el autor de este trabajo, en algunos momentos se seguirá una metodología de prueba y error, y a partir de estas pruebas, ir adaptando la elección final del entorno.

1.5. Planificación del Trabajo

1. **Análisis de herramientas**

   a) *Investigación sobre NIDS*: se buscará información sobre que es un sistema NIDS y sus usos.

   b) *Búsqueda de herramientas NIDS más usadas*: se buscará por diferentes webs cuales son las herramientas NIDS más usadas o recomendadas.

   c) *Análisis de las web de los desarrolladores de las soluciones NIDS*: en las webs de los desarrolladores se revisará que versiones ofrecen que sean gratuitas, y las funcionalidades que incluyen. También se revisará que documentación o tutoriales ofrecen para su posterior implementación.

   d) *Análisis de requisitos de las soluciones NIDS*: se revisarán los requisitos de cada solución NIDS

e) *Investigación sobre SIEAM*: se buscará información sobre que es un sistema siem y sus usos.

f) *Búsqueda de herramientas SIEM más usadas*: se buscará por diferentes webs cuales son las herramientas SIEM más usadas o recomendadas.

g) *Análisis de las web de los desarrolladores de las soluciones SIEM*: en las webs de los desarrolladores se revisará que versiones ofrecen que sean gratuitas, y las funcionalidades que incluyen. También se revisará que documentación o tutoriales ofrecen para su posterior implementación.

h) *Análisis de requisitos de las soluciones SIEM*: se valorarán los requisitos hardware y software necesarios para cada solución, de cara a la elección de la herramienta final, teniendo en cuenta los recursos de los que se dispone.

2. **Elección de entorno**

a) *Elección de hardware*: A partir de los recursos que se tienen con anterioridad al inicio de este trabajo, junto con los requisitos mínimos/recomendados de las diferentes herramientas elegidas, se decidirá sobre el hardware a usar.

b) *Elección de software*: Se elegirán las solución NIDS y SIEM que más se adapte al objetivo de este TFM, y se realizará de manera conjunta con la Elección de hardware.

3. **Implementación del entorno**

a) *Instalación de herramientas NIDS*: Se llevará a cabo la instalación de la herramienta NIDS elegida para este proyecto, y se realizará la resolución de problemas que puedan surgir durante la instalación.

b) *Instalación de herramientas SIEM*: Se llevará a cabo la instalación de la herramienta SIEM elegida para este proyecto, y se realizará la resolución de problemas que puedan surgir durante la instalación.

c) *Integración de herramientas*: Se analizará y se realizará la integración de NIDS con SIEM, para que pasen la información de una herramienta a otra.

d) *Análisis de datos*: Se llevará a cabo la configuración de las herramientas para poder visualizar los datos obtenidos, y poder hacer un primer análisis de los resultados, mediante el uso de dashboards.

4. **Experiencia de uso**

   a) *Revisión de alertas generadas*: Se revisarán las alertas generadas por la solución implementada, para poder determinar sin son problemas de seguridad reales o si se trata de "falsos positivos".

5. **Conclusiones**

   a) *Obtención de conlusiones*: se realizará un resumen final del estudio realizado.

   b) *Propuestas de mejora*: se propondrán puntos de mejora que se podría implementar en sobre el entorno elegido.

1.6. Análisis de posibles riesgos

A continuación se expondrán las posibles situaciones que pueden generar riesgos en la elaboración de este TFM:

- **Obtener información concreta para este TFM**: Dado que el tipo de herramientas contempladas para el desarrollo de este trabajo es la primera vez en las que se trabaja con ellas, será importante poder obtener toda la información necesaria para las herramientas finalmente seleccionadas, que permitan la implantación del entorno. Como estas herramientas sirven para diferentes usos, existe el riesgo de que el entorno que se desea implementar no sea una de las soluciones más comunes, y por tanto, no exista tanta información, pese a que se pueda implementar, por lo que requerirá más tiempo de investigación y testeo.

- **Recusos hardware y software necesarios**: Las herramientas software de este TFM están enfocadas a entornos empresariales, por lo que puede existir riesgos para el desarrollo este TFM, ya sea por limitaciones del software, como puede ser versiones gratuitas con funciones limitadas o versiones de pago con costes demasiado altos para un laboratorio. Del mismo modo, también pueden surgir problemas en cuanto al hardware, ya sea por requisitos de funcionalidades o de recursos mínimos del hardware (Disco Duro, RAM, CPU,....). Como el objetivo de este trabajo es puramente de estudio, e inicialmente sin idea de ser una solución definitiva para el entorno de estudio, el coste económico puede ser un riesgo en la consecución del mismo.

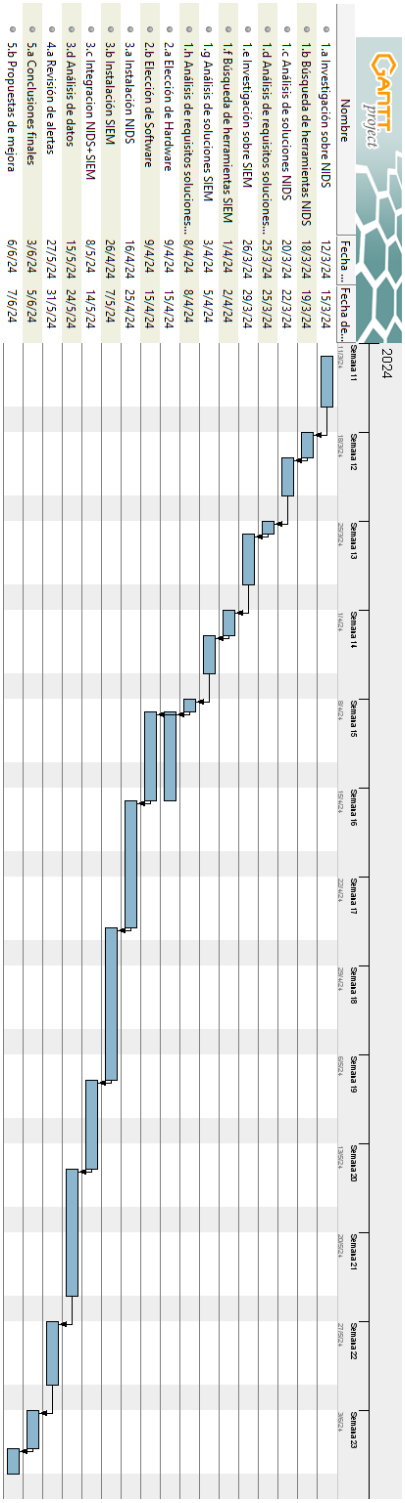

Figura 1.1: Diagrama Gantt de la planificación

- **El tiempo**: Como se ha comentado en los puntos anteriores, existen factores de riesgo que pueden suponer cambios de diseño según se vaya avanzando en el análisis e implementación de la solución, por lo que estos riesgos en sí suponen un riesgo adicional, el tiempo, ya que la necesidad de tener que cambiar el diseño durante el avance del trabajo, supondrá una desviación en el tiempo adicional la tener que rediseñar o adaptar el escenario a la nueva solución.

1.7. Breve descripción de los otros capítulos de la memoria

Tras haber contextualizado y definido este trabajo, se hará un pequeño resumen de los siguientes apartados que se abordarán a continuación:

- **Análisis de herramientas**: Se dará una explicación de que son las dos herramientas que se utilizarán en este trabajo (NIDS y SIEM), además de hacer un análisis de las soluciones más utilizadas en cada caso, para posteriormente valorar cual será la que encaje mejor, de acuerdo a los recursos disponibles y el objetivo esperado.

- **Elección de entorno**: Tras haber realizado un análisis de las soluciones que existen en el mercado, se pasará a elegir la solución que a priori se adapta mejor a este TFM, así como la justificación de la elección tomada.

- **Implementación del entorno**: Durante este apartado se explicará paso a paso, el procedimiento llevado a cabo para instalar, configurar e integrar las diferentes soluciones hardware y software seleccionadas. Así como los primeros pasos realizados sobre la solución.

- **Experiencia de uso**: Se explicará los resultados obtenidos durante el tiempo de monitorización con la solución, y el procedimiento que se han ido siguiendo para analizar la información obtenida por la solución, para poder determinar si existen problemas de seguridad en la red, y las medidas tomadas.

Capítulo 2

Análisis de herramientas

Antes de entrar en detalle sobre la solución implementada para este trabajo, se explicará las diferentes herramientas que lo compondrán, así como las diferentes opciones que existen para cada una de ellas. Dado que este trabajo es una introducción a este tipo de soluciones, solo se valorarán aquellas aplicaciones que sean gratuitas.

2.1. Defición de NIDS

NIDS es un Sistema de Detección de Intrusos en Red (Network Intrusion Detection System), cuya principal función es la de analizar el tráfico que se produce dentro de una red para detectar ataques o tráfico malintencionado. Para ello, las herramientas NIDS monitorizan todo el tráfico que generan los equipos de la red, y en tiempo real, se va comparando cada paquete para determinar si es un tráfico sospechoso o no. Para marcar un paquete sospechoso existen dos formas, la primera sería basada en patrones asociados a ataques conocidos (o firmas), para este caso es necesario descargar dentro del software de NIDS una base de datos de firmas conocidas, aunque también es posible crear reglas de manera personalizada, tanto para crear una base de firmas desde cero como para complementar una ya preexistente. La segunda forma sería comparando el tráfico que se produce normalmente dentro de la red, y en el caso de que se produjese una tráfico anormal, este sería marcado como sospechoso. Este último método es interesante para los ataques Zero-Day, ya que al producirse un tráfico anómalo, este será marcado como sospechoso para ser revisado y actuar en consecuencia; en un sistema basado en firmas, es muy probable que no sea detectado ya que no existe una firma asociada. Aunque también requiere un conocimiento más profundo de la red que se está monitorizando, para poder definir reglas que contemplen el tráfico

"habitual", por ejemplo, ¿que equipos de usuario acceden a cada servidor?.

Cuando un paquete es identificado como sospechoso, se le asigna un nivel de alerta en función del problema de seguridad que puede suponer para los equipos de la red. Tras la identificación de un paquete como sospechoso se pueden realizar dos tipos de acciones según el resultado que se quiera obtener de NIDS. Por un lado estaría una solución pasiva (monitorización) en la que se registra ese paquete en un `log` para posteriormente revisarlo en detalle, y tras su análisis tomar las medidas que se consideren oportunas para mitigarlo (IDS); y por otro lado estaría la solución activa, en la que la herramienta NIDS descarta/bloquea el tráfico que se marque como sospechoso, pasando a ser un Sistema de Prevención de Intrusos (IPS). Esta última opción es la que ofrece mayor seguridad a la red, pero a su vez puede generar más inconvenientes entre los hosts, ya que se pueden descartar paquetes que puedan cumplir un patrón y que la comunicación sea válida (falsos positivos). Una de las soluciones que ofrecen las herramientas NIDS para reducir el número de falsos positivos, es el de descartar a partir de un nivel de alerta, por ejemplo, se puede indicar que se descarten solo aquellos paquetes que estén asociados a un patrón con un nivel de alerta medio o superior, por lo que los paquetes inferiores a este nivel de seguridad no serán descartados. Aún así, los paquetes no descartados se incluirá de igual modo en el `log`. Con esta opción intermedia entre ambas, se obtiene un nivel de seguridad algo inferior, ya que pueden haber paquetes de nivel bajo que supongan un problema para la red si no existen contramedidas en los equipos a los que pueda ir dirigido el ataque, pero reducirá la cantidad de falsos positivos detectados, y por tanto el impacto en los usuarios.

2.2. Herramientas NIDS

En este apartado se revisarán las principales herramientas que existen a día de hoy en el mercado. Dado que en este trabajo se va a realizar una primera aproximación a este tipo de soluciones, se valorarán solo aquellas que sean gratuitas.

2.2.1. Snort

Snort es una de los primeros sistemas de detección de intrusos en red, libre y gratuito (desarrollada por Cisco Systems en 1998), el cual está basado en la detección de tráfico sospechoso, de acuerdo a patrones conocidos (firmas). Una de las principales ventajas de esta herramienta es que al ser de las primeras que se crearon tiene una comunidad muy activa, por lo que las firmas están continuamente en crecimiento y existen bastantes foros de ayuda para la creación/configuración de reglas propias.

2.2.2. Suricata

Suricata es una herramienta de código abierto que se puede usar tanto como Sistema de Detección de Intrusos (IDS) como Sistema de Prevención de Intrusos (IPS). Fue desarrollada por la Open Information Security Foundation (OISF) y lanzada en 2010. Esta solución surge para corregir algunas de las carencias que tiene Snort, principalmente la de poder trabajar con varios hilos, y de este modo, sacar mayor capacidad de procesado tanto de los procesadores como de las tarjetas gráficas actuales. El formato para la creación de reglas dentro de Suricata es el mismo que emplea Snort, por lo que se puede hacer uso de ellas, aprovechando de este modo el soporte de la comunidad de Snort, además de la comunidad de Suricata que también es bastante activa.

2.2.3. Zeek

Zeek (anteriormente Bro) es una herramienta gratuita y de código abierto para el análisis de red. Fue creada en 1994 por Vern Paxson, para ser un sistema de monitorización de seguridad de red, pero que también puede ser usada como NIDS, pero no como IPS ya que no es una solución activa de seguridad. Zeek para la detección de amenazas hace uso tanto de firmas

como de analisis de comportamiento en la red, para esto el motor de Zeek ante un evento que sucede en la red va creando una serie de relaciones con los siguientes eventos que suceden. Zeek al igual que Suricata permite el funcionamiento multi-hilo siendo por tanto una solución con buena capacidad de procesado de paquetes.

2.3. Definición SIEM

SIEM nace de la combinación de la administración de información de seguridad (System Information Manager) y la administración de eventos de Seguridad (System Event Manager), dando como resultado un sistema de Gestión de Eventos e Información de Seguridad (Security Information and Event Management). Su función es la de centralizar toda la información de eventos que suceden en una red, los cuales son recogidos por diferentes sondas, y con los datos recopilados, realizar un análisis de dichos eventos, para poder detectar amenazas o vulnerabilidades, en tiempo real.

Por tanto la función más importante dentro de un SIEM es la capacidad de poder agrupar, relacionar y trabajar todos los datos recibidos. Con el auge del Machine Learning, está permitiendo que este proceso de análisis y toma de decisiones sea cada vez más automatizado, reduciendo por tanto, los tiempo de respuesta ante amenazas de seguridad, así como la reducción de los "falsos positivos".

2.4. Herramientas SIEM

2.4.1. OSSIM

Open Source Security Information Management (OSSIM) además de tener su propio motor de correlación de eventos y la integración mediante plugins de fuentes de datos, está formada por varias herramientas con licencia GPL, siendo cada una de ellas específica para una detección de anomalías (Snort, Openvas, Nagios, SPADE...). OSSIM fue desarrollada por AlienVault y en 2018 fue adquirida por AT&T Comunnications. Tras la adquisición se crearon 2 opciones para su implementación: una Open Source para instalaciones On-premise (AlienVault OSSIM), y otra de pago para servicios cloud, ofreciendo más funcionalidades (USM Anywhere).

2.4.2. Splunk

Splunk es un software de monitorización y análisis de datos de diferentes orígenes (sistemas, aplicaciones, infraestructuras...), que los indexa y correlaciona en tiempo real. El uso/orígen de los datos está pensado para una explotación más generalista dentro del mundo de IT, pero dispone de versiones de software específicas para su uso como SIEM (Splunk Enterprise Security). Splunk es la herramienta SIEM más utilizada a nivel mundial por noveno año consecutivo, de acuerdo al informe realizado por Gartner (Gartner (2024)). Splunk dispone de una versión gratuita con limitaciones de la versión Enterprise (limite de datos mensuales, instalación individual, sin posibilidad de alertas). También dispone de un portal con plugins desarrollados por la comunidad de Splunk (Splunkbase).

2.4.3. Elasticsearch

Elasticsearch es un motor de búsqueda y analítica distribuido, gratuito y abierto para todos los tipos de datos (texto, númericos, geoespaciales, estructurados y no estructurados). Uno de los principales usos de Elasticsearch es para analizar los `logs` de diferentes aplicaciones que pueden estar distribuidas dentro de la red que se quiere revisar. Al trabajar con `logs` de diferentes aplicaciones, el principal problema es que la estructura de estos `logs` varia de una aplicación a otra, por lo que dentro de la pila ELK existe un módulo que sirve para desgranar la información que devuelven los diferentes `logs` y poder normalizarla antes de incluirla dentro de Elasticsearch (Logstash), para así poder tratar toda la información obtenida desde distintos origenes de una manera global.

Con el incremento de uso de Elasticsearch, se han ido desarrollando diferentes integraciones, permitiendo así que esta normalización realizada por Logstash, se pueda hacer de una manera automática, al menos para un uso

general, aun así siempre se puede hacer uso de Logstash para que el usuario pueda determinar que datos del `log` de la aplicación en cuestión sean procesados (mayor o menor cantidad de metadatos).

Capítulo 3

Elección de entorno

3.1. Hardware

Para la selección del Hardware que intervendría en este laboratorio, se partió de la base de utilizar el hardware y recursos de los que ya se disponía, para así no tener que hacer ninguna inversión económica, y además, poder concluir hasta que punto es viable implementarlo en un hogar, ya que un hogar aunque no pueda ser a priori tan interesante para atacantes, no deja de ser un entorno vulnerable (más aún si cabe que una empresa). Se partió con los siguientes elementos:

- Router de proveedor de internet
- Dispositivos conectados a la red doméstica (ordenadores, móviles, tablets, IoT,...)
- Maquina virtual instalada en un ordenador de sobremesa configurando la tarjeta de red en modo puente para que pueda capturar paquetes de la red.

Tras instalar y configurar todas las herramientas (se comentarán en los siguientes apartados), se detectó que al ser un ordenador que no siempre estaba encendido (por el consumo energético) el tráfico obtenido era circunstancial y surgió el interés por crear un entorno de pruebas que sirviese como solución final. Por tanto, se decidió hacer uso de un NUC (Gigabyte Brix), ya que el consumo del procesador que lleva integrado es muy bajo, juntando todos los componentes de ambos el consumo total del NUC era la décima parte respecto al sobremesa. Se volvió a instalar todas las herramientas desde cero, pero en este caso no se hizo uso de una máquina virtual, sino que se instalaron directamente en el NUC. Tras tener durante unos días capturando

tráfico, se vio que el NUC no tenía suficiente capacidad de procesado para poder realizar las tareas de Suricata (NIDS) y de ElasticSearch (SIEM).

Tras valorar las ventajas e inconvenientes de las dos soluciones iniciales de Hardware se decidió implementar una solución mixta de ambas. Se instalaron en el NUC todas las herramientas de software necesarias para poder capturar el tráfico, así como un plugin de Elasticsearch para hacer la normalización y envio de datos a ELK. En una máquina virtual dentro del ordenador de sobremesa, se instaló el software necesario para recibir la información capturada por el NUC y para el análisis de la misma, ya que para el procesado de la información no es necesario que se realice en tiempo real, por lo que de manera periodica se iniciará la máquina virtual, para recoger todos los datos acumulados por el NUC. Esta solución final tiene dos principales inconvenientes, el primero es el tiempo necesario por parte del servidor para obtener el acumulado de datos durante el tiempo que no estuvo encendida la máquina virtual (varios minutos), ya que el NUC tiene primero que enviar todos los `logs`, y luego Elasticsearch tiene que incorporarlos y procesarlos. El segundo problema, es que al no tener la información en tiempo real, no es una solución que pueda alertarnos en tiempo real de las amenazas que puedan surgir en la red.

Como el objetivo de este trabajo es el de valorar si existen amenazas en un red doméstica, y por consiguiente si es segura, no es necesario obtener alertas en tiempo real, ya que no está dentro del trabajo la toma de medidas para corregir las amenazas al instante. En el caso de que aparezcan, tras revisar las vulnerabilidades, se tomarán medidas para evitar que se vuelvan a producir.

Cuando se implementó el entorno, se detectó que el tráfico obtenido por el NUC era únicamente el que tenía como origen o destino su propia IP, así como mensajes de broadcast. Se configuró en modo promiscuo la tarjeta de red del NUC, pero se obtuvo el mismo resultado. Tras revisar el problema, se detectó que todo el tráfico que pasa por el switch integrado en el router del operador de internet es a nivel 2. En un equipo de red de nivel 2, cuando se envía un paquete de red, éste contiene 2 campos MAC que son, la dirección física de origen y la dirección física de destino. El switch integrado en el router, tiene una tabla de MACs, en la cual se indica las direcciones MACs que hay asociadas a cada puerto físico del router, por tanto cuando un equipo externo a la red doméstica quiere enviar un mensaje (o responder) al ordenador, el router pondrá la dirección MAC de la tarjeta de red del ordenador y lo enviará por el puerto del switch por el que ha aprendido la MAC del ordenador, y no lo enviará por el resto de puertos del switch, de ahí que el NUC no reciba los mensajes dirigidos a otros dispositivos de la red.

Para solventar esta problemática se plantearon dos posibles soluciones:

1. Instalar el NUC en modo “Inline”, esto quiere decir que el NUC se pondría en medio entre los equipos de la red y el router del operador de internet. Pero para ello sería necesario que el NUC dispusiese de 2 tarjetas de red, una para conectar con los equipos de la red local y otra para conectar el NUC al router. El NUC empleado solo dispone de una tarjeta de red por cable y otra inalámbrica, por lo que implicaría perder velocidad de conexión de los equipos locales, así como dificultando el conexionado de todos los dispositivos a la red local. Además al poner el NUC inline, la carga de CPU aumentaría al tener que enrutar todo el tráfico de la red local hacia el router, pudiendo repetirse la misma situación que se comentó anteriormente al instalar tanto Suricata como Elastic en el NUC.

2. Instalar switch gestionable. En algunos de estos tipos de switches, existe una funcionalidad pensada para el análisis de tráfico de la red (port mirroring), que lo que hace es reenviar una copia de todo el tráfico que pasa por el switch hacia el puerto que marquemos como port mirroring.

Finalmente se eligió la segunda opción, ya que la primera con los recursos actuales sería una solución más ineficiente.

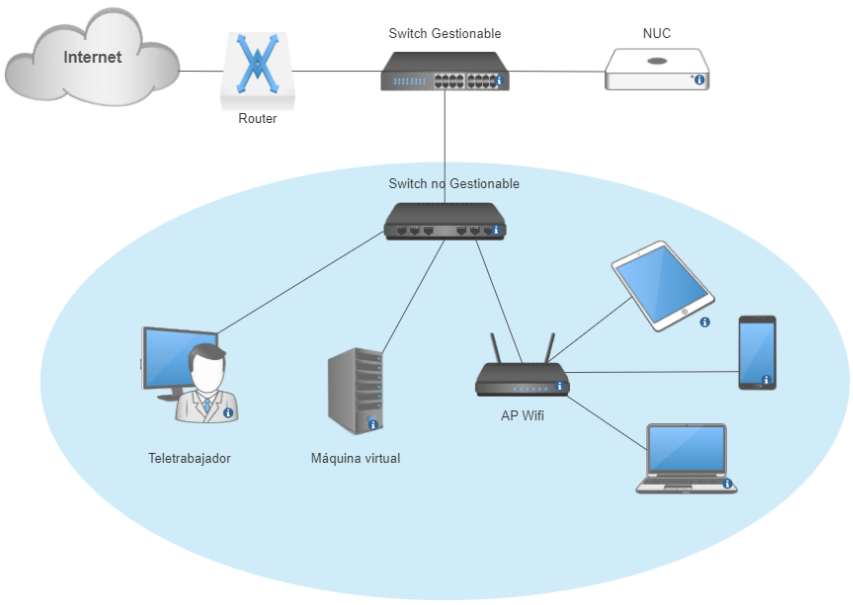

Figura 3.1: Esquema laboratorio

3.2. Software

3.2.1. Eleccion NIDS

Como se indicó en el apartado 3 para la elección del entorno en este trabajo, la monitorización del tráfico de la red doméstica, se realizará sobre un miniPC con recursos limitados. Para poder reducir la cantidad de servicios o aplicaciones instaladas, así como para poder sacar el máximo provecho a los reducidos recursos, se optó por usar Ubuntu Server sin entorno gráfico y con los servicios mínimos del sistema, siendo OpenSSH Secure el único servicio que se añadió durante el proceso de instalación del sistema operativo, para así poder tener gestión remota del mismo.

Tras revisar las diferentes soluciones NIDS indicadas en el apartado 2.2, se elegió Suricata como herramienta para este fin dentro de la solución a implementar.

Los motivos principales para la elección de esta aplicación fueron:

- Software de código abierto: ya que además de no suponer un coste adicional el uso de esta herramienta, el código es modificado por programadores de manera libre, ofreciendo la posibilidad desarrollar mejoras sobre el código fuente
- Mejora de procesado respecto a Snort
- Comunidad muy activa en los foros.

3.2.2. Elección SIEM

La solución SIEM que finalmente se eligió fue la de la pila ELK, por los siguientes motivos:

- Software gratuito: aunque Elastic tiene versiones de pago, se analizaron las funcionalidades y limitaciones que tenía la versión gratuita, siendo la versión gratuita suficiente para el objetivo de este TFM.
- Durante el análisis realizado para las herramientas SIEM, se revisó los tutoriales que incluía la empresa en su web, siendo bastante intuitivos y guiados.
- El foro de ayuda en la propia web era bastante activo.

Capítulo 4

Implementación del entorno

4.1. Instalación herramienta NIDS

Tras instalar el sistema operativo lo siguiente fue instalar Suricata, para ello se siguió la Wiki de la web oficial.

Para Ubuntu/Debian además de indicar los paquetes mínimos y recomendados, también hay un repositorio propio de Suricata, lo cual facilita bastante la instalación de la aplicación:

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update
```

Tras añadir el repositorio y actualizarlo, solo sería ejecutar el siguiente comando para realizar la instalación:

```
sudo apt-get install -y suricata jq
```

Lo siguiente sería realizar la configuración de Suricata, para ello lo primero que habrá que comprobar es la interfaz de la tarjeta de red que se usará para conectarse a la red (figura 4.1).

Esta información, junto con los rangos de la red domestica, habrá que indicarlo en el fichero de configuración de Suricata, que se encuentra en **`/etc/suricata/suricata.yaml`** (Figura 4.2).

Tras indicar esta información en el fichero de configuración de Suricata, ya lo que quedaría por realizar es la carga de las bases de firmas con las que se quiere comparar el tráfico monitorizado. Dentro de las bases de firmas, existen algunas que son gratuitas y otras que son de pago, por lo que de

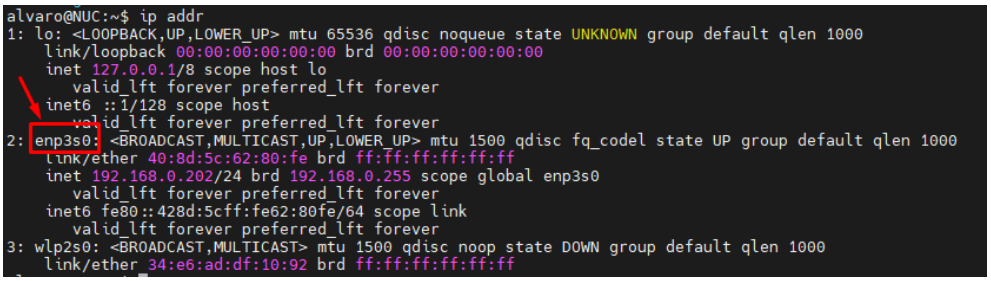

Figura 4.1: Interfaz de red NUC

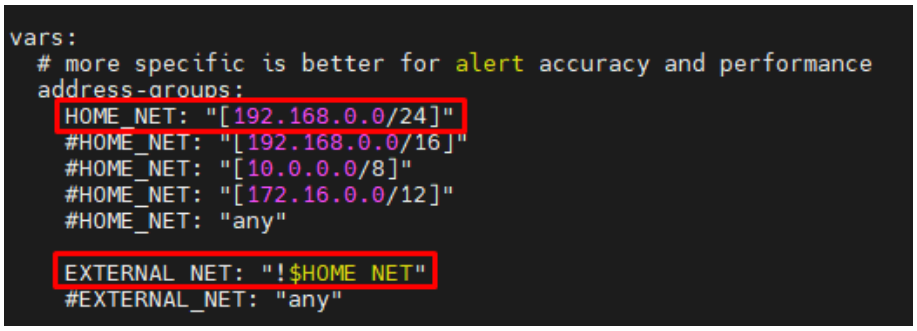

Figura 4.2: Fichero configuración Suricata - Redes

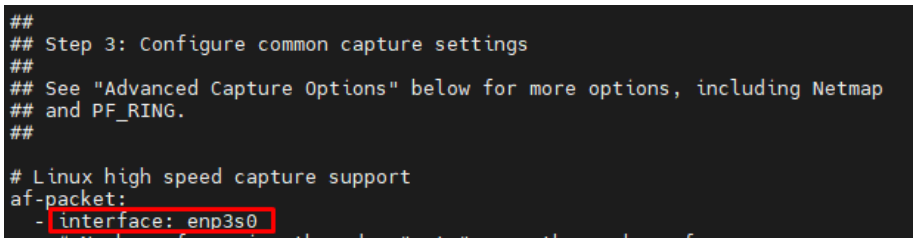

Figura 4.3: Fichero configuración Suricata - Interfaz

acuerdo al enfoque del proyecto de buscar soluciones con el menor coste posible, solo se instalarán las bases de firmas que son gratuitas. Para hacer la instalación de las firmas, primero hay que ejecutar el comando:

```
sudo suricata-update list-sources
```

Y del listado que aparece, habrá que ir añadiendo individualmente cada base de firmas que se necesite con:

```
sudo suricata-update enable-source NombreDeLaFirma
```

Aunque las firmas se pueden personalizar, y aplicar acciones en función de cada firma, en este trabajo se usarán varias bases de firmas gratuitas pero con sus configuraciones por defecto.

Es importante revisar que la tarjeta de red que se vaya a usar esté en modo promiscuo, ya que en caso de no estarlo, todo el tráfico de red que contenga como destino una dirección MAC diferente de la que tiene asociada la tarjeta de red, será descartado y por tanto no será monitorizado por Suricata. Esto también se puede habilitar por defecto con el arranque del sistema creando el fichero `/etc/systemd/system/bridge-promisc.service` con el siguiente contenido:

```
[Unit]
Description=Makes interfaces run in promiscuous mode at
boot
After=network-online.target
[Service]
Type=oneshot
ExecStart=/usr/sbin/ip link set dev enp3s0 promisc on
TimeoutStartSec=0
RemainAfterExit=yes
[Install]
WantedBy=default.target
```

Al crear este servicio, habrá que habilitarlo con el arranque del sistema operativo con el comando:

```
sudo systemctl enable bridge-promisc
```

Se puede comprobar si la tarjeta de red está en modo promiscuo con el comando *ip link* (figura 4.4):

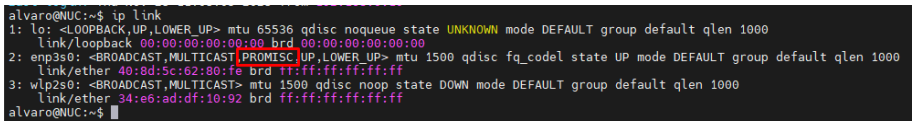

Figura 4.4: Modo promiscuo en interfaz de red

Para confirmar que Suricata está capturando tráfico se puede hacer uso de una firma creada para pruebas dentro del rule-set de *et/open*, para ello se tebdrña que acceder a la siguiente web `http://testmynids.org/uid/index.html` (desde consola de comandos se podrá hacer con `curl` antes de

la URL), y para confirmar que se ha detectado, se puede el comando que aparece en la figura 4.5 indicando el ID de firma asociada.

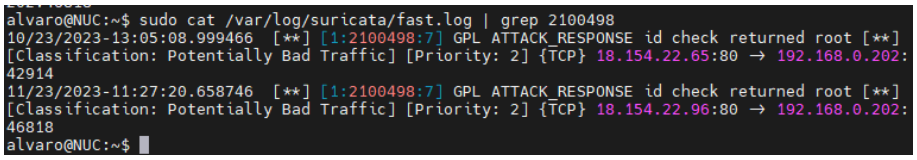

Figura 4.5: Test Suricata

4.2. Instalación herramienta SIEM

Para la instalación del SIEM, lo que se optó fue por crear una maquina virtual en VirtualBox habilitando 6 núcleos del procesador del host y 8Gb de RAM, junto con un espacio de almacenamiento de 80 Gb. En la máquina virtual se instaló como Sistema Operativo Ubuntu Server con los recursos mínimos (al igual que se hizo con el NUC para Suricata).

Tras esto se instaló la pila ELK, la cual debe ser instalada en un orden en concreto, para que la integración entre herramientas sea lo más automática y sencilla posible.

Lo primero fue añadir el repositorio de ElasticSearch, para así poder instalar todas las herramientas necesarias:

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-
elasticsearch | sudo gpg -dearmor -o
/usr/share/keyrings/elasticsearch-
keyring.gpg

sudo apt-get install apt-transport-https

echo "deb [signed-by=/usr/share/keyrings/elasticsearch-
keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt
stable main"| sudo tee /etc/apt/sources.list.d/elastic-8.x.
list
```

El primer módulo del stack que se instalará es ElasticSearch, que será el encargado de realizar búsquedas sobre la información almacenada. Posteriormente se habilitó para que arrancase como servicio del sistema al iniciar el Sistema Operativo.

```
sudo apt-get update && sudo apt-get install elasticsearch
-y

sudo systemctl enable elasticsearch.service
```

```
sudo apt-get update && sudo apt-get install elasticsearch
-y
sudo systemctl enable elasticsearch.service
```

Uno de los cambios introducidos en la última versión de ELK (version 8), es que se ha introducido por defecto las funciones de securización de la herramienta, siendo estas funciones la de autenticación y autorización mediante un usuario/contraseña con permisos de administrador, los cuales son generados aleatoriamente por la aplicación durante la instalación, por lo que al hacer la instalación desde consola con el comando indicado anteriormente, es importante revisar el `log` para guardar el usuario y contraseña generados (figura 4.6).

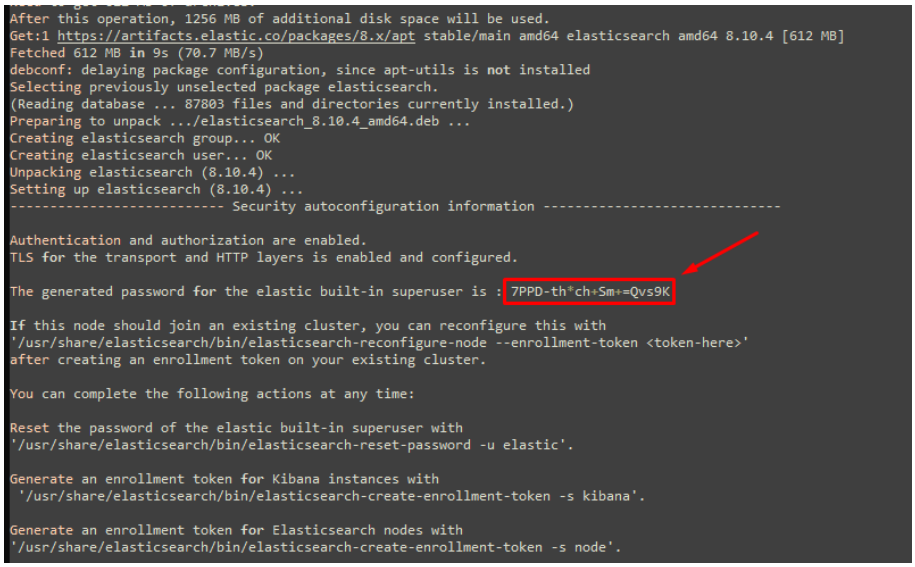

Figura 4.6: Autogeneración de contraseña para usuario elastic

También se crean certificados y claves para TLS tanto para las comunicaciones como para la conexión HTTP, que son habilitadas por defecto, por lo que la conexión via web se realizará mediante HTTPS (HTTP seguro). Estos certificados y claves que se generar durante la instalación, son almacenados en la ruta `/etc/elasticsearch/certs/http_ca.cr`.

Para confirmar que ElasticSearch está funcionando de manera correcta, se podrá comprobar mediante el comando `curl`, al cual habrá que pasarle como parametros:

- Ruta en la que se encuentra el certificado mencionado anteriormente para HTTPS.
- El usuario para acceder, que por defecto es “elastic”.
- la URL de acceso, que al lanzarse el comando localmente, se puede hacer uso de la URL por defecto `https://localhost:9200`.

En la figura 4.7, se puede ver un ejemplo del comando y el resultado que debería dar tras introducir la contraseña del usuario “elastic”..

Figura 4.7: Prueba de conectividad a ElasticSearch mediante consola

El siguiente módulo será Kibana, que es el encargado de mostrar la información almacenada en ElasticSearch de manera sencilla haciendo uso de Dashboards. Para instalar Kibana solo se tendrá que instalarlo desde la linea de comandos haciendo uso del repositorio que se añadió anteriormente:

```
sudo apt-get install kibana
```

Tras la instalación, habrá que asociar Kibana con ElasticSearch, para ello se generará un token desde ElasticSearch, y este token será el que habrá que adjuntar al comando de vinculación de Kibana a ElasticSearch (figura 4.8).

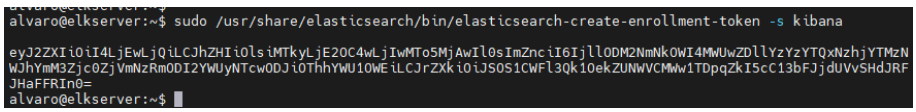

Figura 4.8: Generación de token de ElasticSearch para Kibana

```
sudo /usr/share/kibana/bin/kibana-setup -enrollment-token
<TOKEN>
```

Para poder acceder via web a Kibana, si se va a acceder desde otro ordenador diferente al que está instalado Kibana, será necesario entrar en el fichero de configuración de Kibana (`/etc/kibana/kibana.yaml`) y modificar el campo `server.host`, ya que por defecto vienen con el valor `localhost`, por lo que será solo accesible desde el propio host. Si se desea, también se podrá modificar el puerto al que habrá que conectarse para acceder a Kibana, que por defecto es el 5601 (figura 4.9).

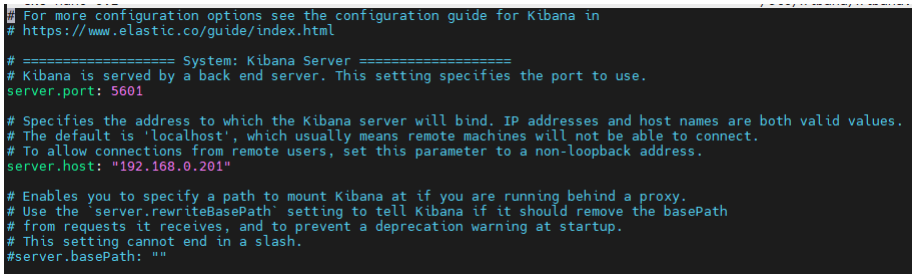

Figura 4.9: Configuración de parametros en kibana.yml

Para comprobar que Kibana queda correctamente configurado se accederá mediante HTTP a la IP y puerto indicados en el párrafo anterior, haciendo uso de los credenciales de ElasticSearch (figura 4.10).

Para poder acceder mediante HTTPS a Kibana, será necesario configurar o crear un certificado. Como no se dispone de un certificado emitido por una empresa certificadora, y con el objetivo de poder cifrar las comunicaciones con Kibana, se generará un certificado localmente. Para ello se debe ejecutar el siguiente comando:

```
/usr/share/elasticsearch/bin/elasticsearch-certutil csr
-name kibana-server -dns tfm.com,www.tfm.com
```

Esto generará un fichero zip con dos ficheros: la clave privada y un certificado sin firmar. Para obtener un certificado firmado localmente (sin empresa certificadora), será necesario lanzar el siguiente comando, pasando como parámetros la clave privada y el certificado sin firmar, además de indicar donde se debe guardar el certificado firmado:

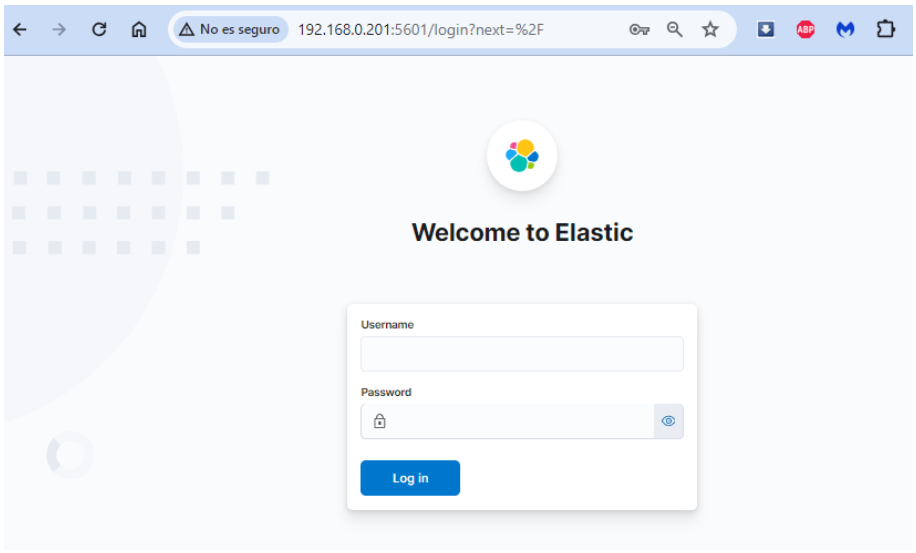

Figura 4.10: Prueba de conectividad a Kibana mediante navegador

```
openssl x509 -req -days 365 -in /usr/share/elasticsearch/
kibana-server/kibana-server.csr -signkey
/usr/share/elasticsearch/kibana-server/kibana-server.key
-out /usr/share/elasticsearch/kibana-server/kibana-server.crt
```

Tras esto, será necesario habilitar la comunicación mediante HTTPS a Kibana, indicando dentro del fichero de configuración de Kibana (`kibana.yml`) donde se encuentran la clave privada y el certificado firmado. A continuación se muestra las lineas de configuración que habría que añadir para este trabajo:

```
server.ssl.certificate: /usr/share/elasticsearch/kibana-
server/kibana-server.crt server.ssl.key:

/usr/share/elasticsearch/kibana-server/
kibana-server.key server.ssl.enabled: true
```

Como el certificado no está firmado por una empresa certificadora, cuando se intente acceder el navegador dará un aviso de que no es segura la conexión, ya que no le ha sido posible confirmar la autenticidad del certificado, al estar firmado localmente (figura 4.11).

Lo siguiente será instalar Elastic Agent, que se ocupará de hacer las co-

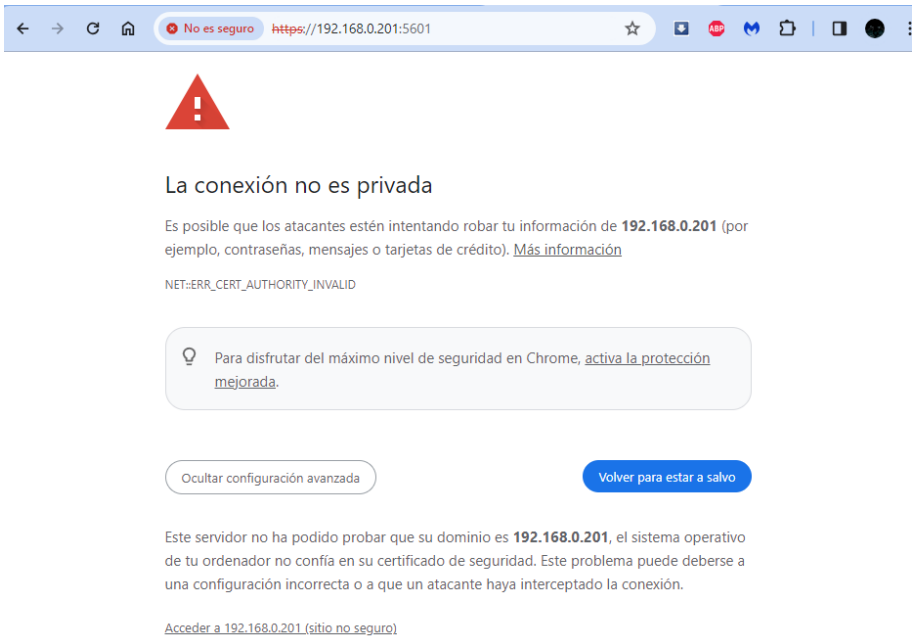

Figura 4.11: Aviso notificación acceso a Kibana sin certificado validado

nexiones entre diferentes equipos (servidor y hosts) para el paso de la información recopilada por cada equipo. Para esto, solo será necesario instalar el paquete elastic-agent y después habilitar éste como servicio del sistema. Esto será necesario realizarlo en cada uno de los hosts que se quiera monitorizar.

```
sudo apt-get install elastic-agent
sudo systemctl enable elastic-agent.service
```

Desde la versión 8 de ElasticSearch, el proceso de asociar los diferentes agentes para el paso de información, se ha simplificado gracias al módulo de Fleet de Kibana. Con Fleet se indica primero qué Elastic Agent será el que actúa como servidor (recibirá la información) y luego se irán añadiendo el resto de agentes, mediante el uso de "tokens". Los agentes que se van añadiendo a Fleet, pueden ser monitorizados, actualizados e incluso configurados desde Fleet, de ahí la simplificación y gestión de esta parte.

Se accederá desde el menú lateral que hay en Kibana (figura 4.12), y luego se tendrá que crear el Fleet Server para posteriormente poder ir añadiendo hosts (figura 4.13).

Se tendrá que indicar la IP/hostname para poder realizar las conexiones entre los nodos y Fleet Server, y esto generará una policy por defecto

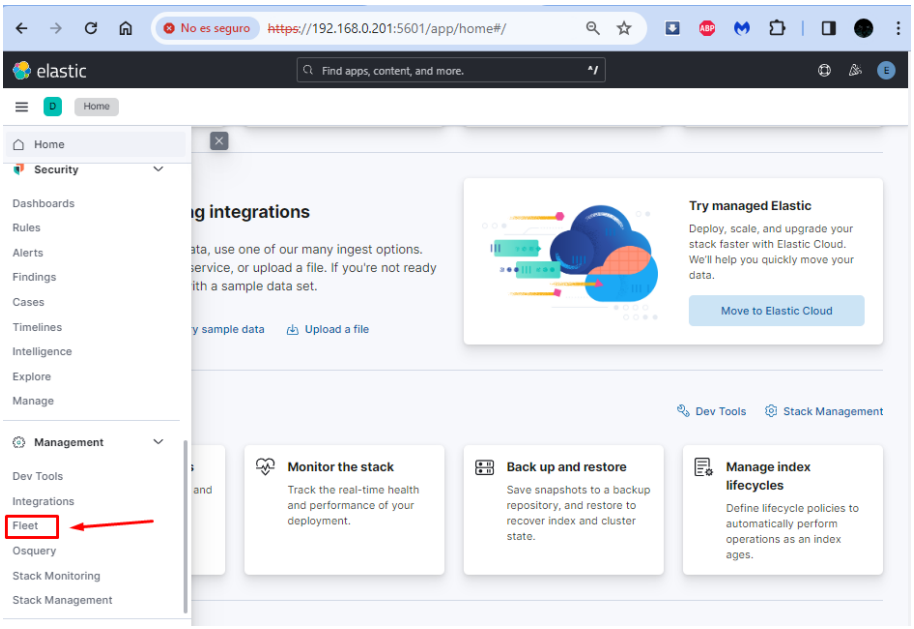

Figura 4.12: Acceso a Fleet desde Kibana

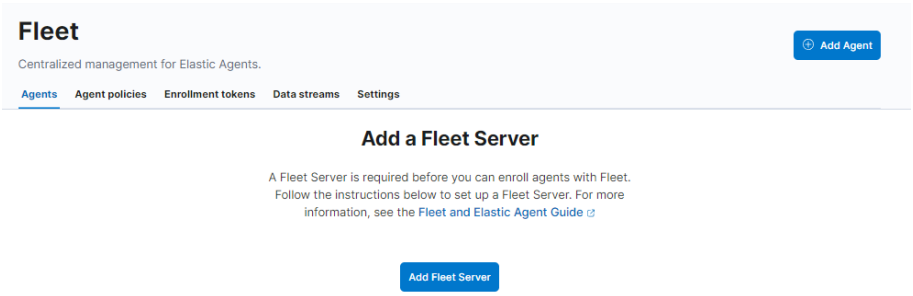

Figura 4.13: Añadir Fleet Server en Kibana

(figura 4.14). Tras generarse las policy solo faltará crear la vinculación de nuestro Fleet Server, y para ello, habrá que lanzar un comando pasando como parametros: la URL de Fleet, el token generado durante este proceso, las politicas, el certificado y el puerto:

```
sudo elastic-agent enroll
-fleet-server-es=https://192.168.0.201:9200
-fleet-server-service-token=AAEAAWVsYXN0aWMvZmxlZXQtc2VydmVy
L3Rva2VuLTE3MDI1NTUwMDE3NTk6OXprUkgwak1TVGVCUkRrWk1qdEhzUQ
-fleet-server-policy=fleet-server-policy
-fleet-server-es-ca-trusted-fingerprint=3000f3c37c66e64eb593
a9aec2c189d2486f9c0a5bad8bb374e871908dc2d412
-fleet-server-port=8220
```

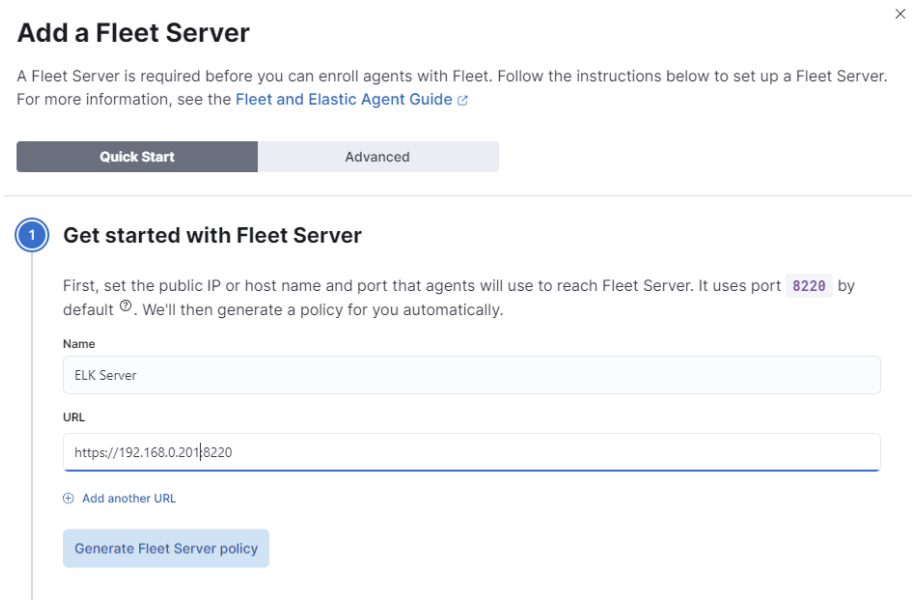

Figura 4.14: Configuración Fleet Server en Kibana

Cuando se haya creado Fleet Server, ya se podrán añadir los diferentes host a monitorizar, de manera similar a como se acaba de realizar para Fleet Server. Basta con seguir el paso a paso al clickar sobre "Add Agent" (figura 4.15).

Será necesario crear una política nueva para los host (agents), en la cual se indicará que tipo de información será recolectada de los agentes instalados en los host, lo cual permitirá realizar modificaciones sobre la información que se quiera recopilar de manera centralizada sobre todos los agentes que estén asociados a esta policy (en vez de tener que ir modificando individualmente cada agente).

Al igual que se hizo para Fleet Server, será necesario instalar en el host a monitorizar el paquete 'elastic-agent', y tras esto solo quedará lanzar el

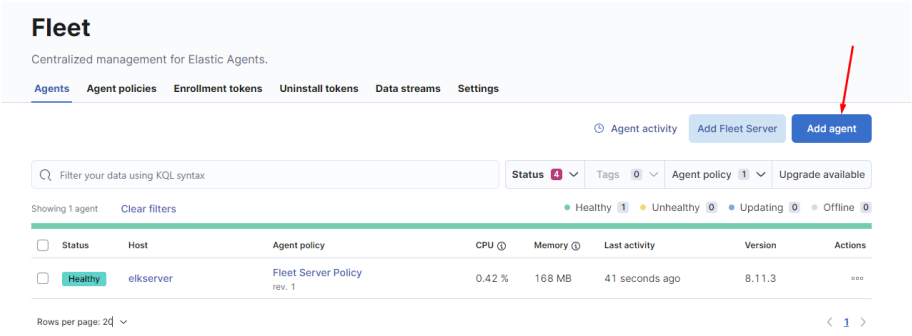

Figura 4.15: Añadir host en Fleet

comando para asociarlo a Fleet Server utilizando el token generado en Fleet Server y la URL donde se encuentra Fleet Server (figura 4.16)

```
sudo elastic-agent enroll -url=https://192.168.0.200:8220
-enrollment-token=UXFnOWFJd0I3dTk0ZDBFb0w0bWM6Q3pvR1FlQzNUbi
1uajdFOE1hQUhidw==
```

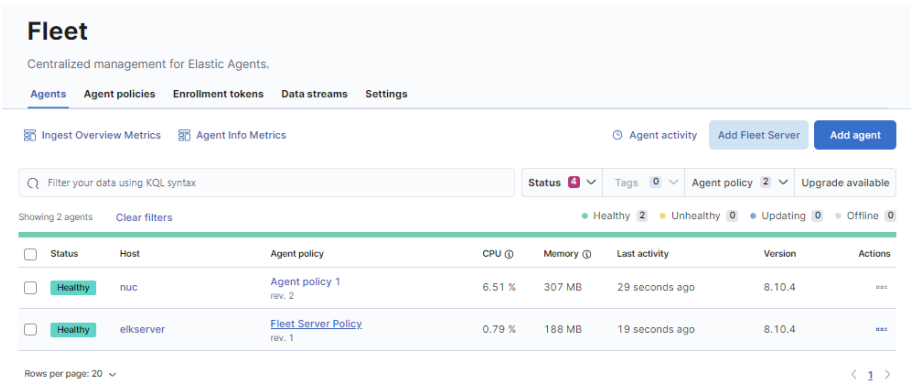

Figura 4.16: Estado de los agentes de Fleet

4.3. Integración de herramientas

Al tener creado el canal de comunicación para el paso de información entre el agente (NUC) y el Servidor ELK, el siguiente paso será el de transformar toda la información que es recopilada por Suricata al formato usado por ElasticSearch. En las últimas versiones de Elastic, han desarrollado un

módulo/funcionalidad dentro de ELK que facilita este trabajo a los usuarios (para un uso no avanzado), ya que es capaz de localizar los `logs` que generan las aplicaciones y desgranar los datos relevantes que se incluyen dentro, para posteriormente enviarlos normalizados de manera transparente al usuario hacia ElasticSearch, esta funcionalidad se llama "Integrations", y se puede encontrar en el home de Kibana (figura 4.17).

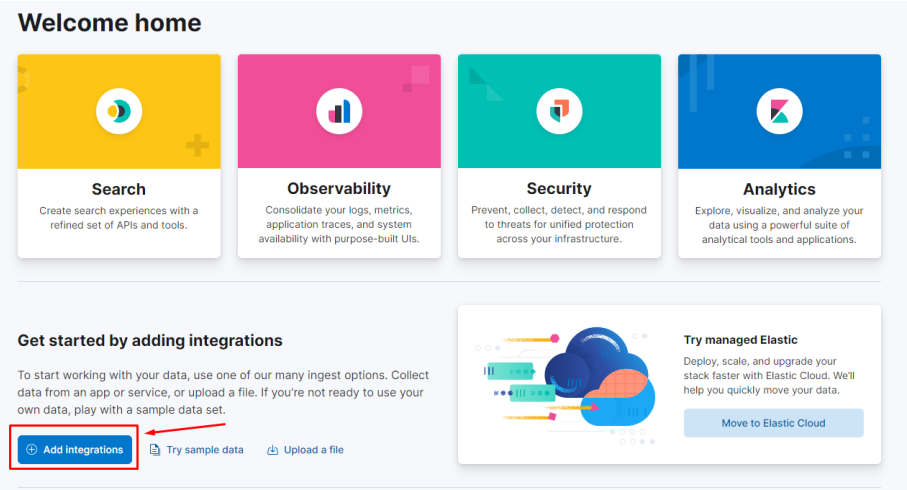

Figura 4.17: Añadir integraciones desde Kibana

Al acceder a Integrations, lo que aparecerá será el listado de aplicaciones para las cuales se han creado esa normalización de datos "automátizada" hacia ElasticSearch. En el momento de elaboración de este TFM, el número de integraciones existentes es de 365, entre la que se incluye Suricata (figura 4.18). Con el paso del tiempo se van añadiendo integraciones para más aplicaciones.

El procedimiento para incluir esta integración es bastante intuitivo, bastará con añadir la integración deseada (en este caso Suricata), y a continuación rellenar algunos parámetros que en este caso los obligatorios serían (figura 4.19):

- **Nombre de la integración**: con esto lo que se permite es que por ejemplo se pueda tener varias integraciones de la misma aplicación, pero con parametrizaciones diferentes, y de este modo, poder asociar cada integración al tipo de configuración que tenga el host. Esto es muy útil cuando existen varios host que tienen el mismo entorno (en los siguientes parámetros se comprenderá mejor este punto).

- **Carpeta de Logs**: inicialmente la integración ya propone una ruta

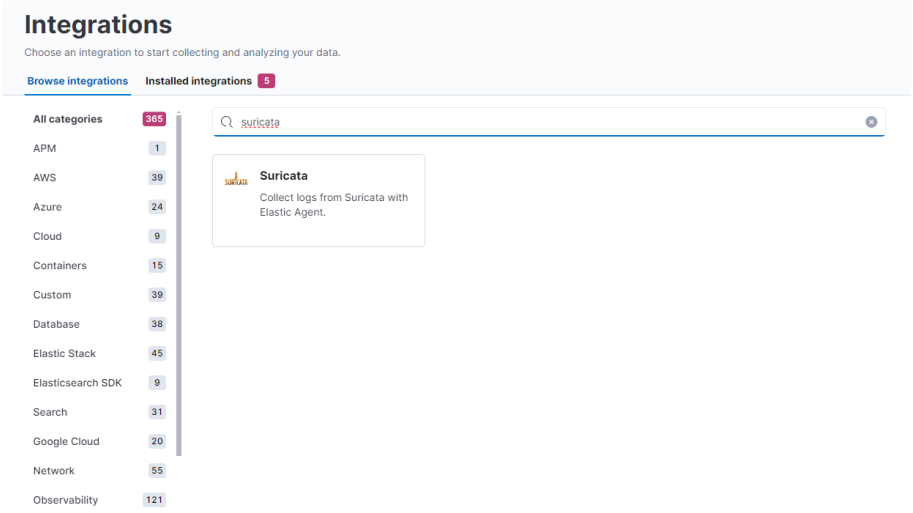

Figura 4.18: Ejemplo listado integraciones en ELK

hacia dicha carpeta, pero es posible cambiarla por otra diferente en el caso de ser necesario.

- **Mantener el evento original**: esto lo que hará será incluir como un parámetro "normalizado" el evento al completo generado por Suricata.

Existen más parámetros parametrizables, pero al tratarse de una primera toma de contacto con la solución no se profundizará en las opciones avanzadas de la integración.

El último paso, sería indicar sobre que Hosts queremos aplicar esta integración, para ello se hará uso de Policies, pudiendo elegirse el añadir esta integración sobre una policy existente o crear una nueva policy. También permite incluir a la vez, una integración para la monitorización del estado del host (figura 4.20).

Para este TFM, cómo solo se hará uso de un agente/host (NUC), se creará una nueva policy ya que no se dispone de ninguna (a excepción de la policy generada de manera automática para el servidor ELK). El uso de policies en un entorno con gran cantidad de agentes, aporta beneficios como pueden ser:

- **Escalabilidad**: si tenemos varios agentes que comparten integraciones, por ejemplo, todas tienen Suricata instalado, cuando añadimos un nuevo agente, bastará con asociarlo a la policy que se haya creado, y desde Kibana/Fleet, se podrá enviar toda la parametrización establecida para esa policy.

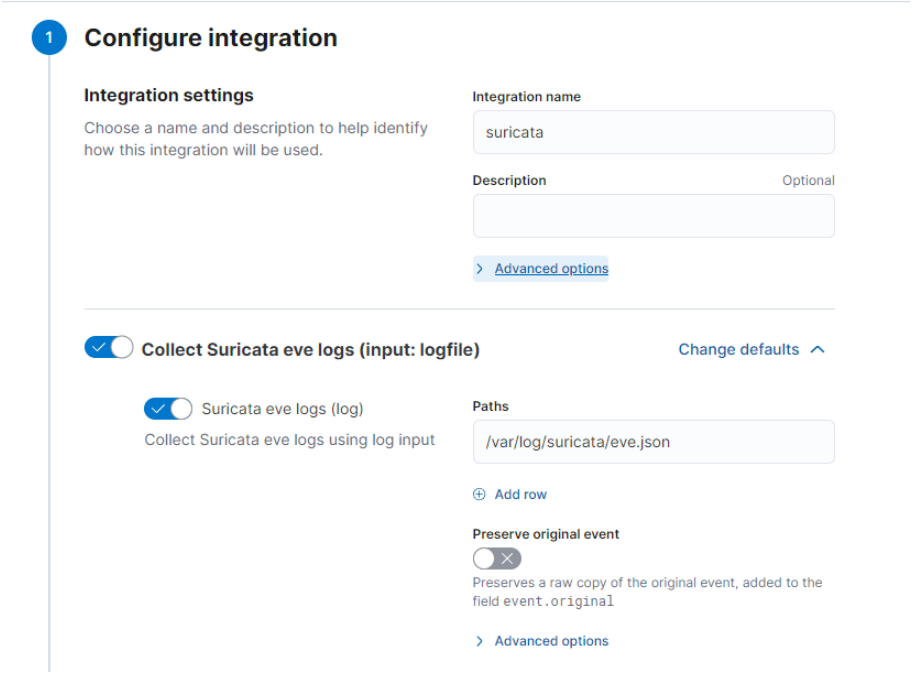

Figura 4.19: Configuración de integración de Suricata (I)

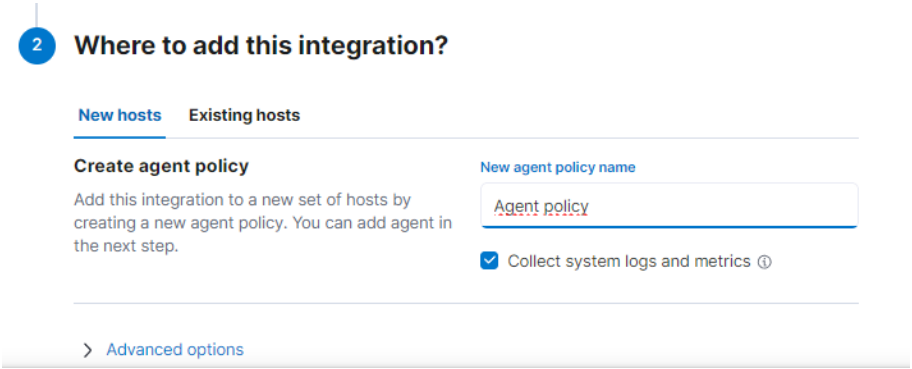

Figura 4.20: Configuración de integración de Suricata (II)

- **Facilidad de despliegue**: si es necesario realizar una modificación de algún parámetro dentro de la integración, bastará con realizarla sobre la policy, para que todos los agentes asociados a esta policy actualicen sus parámetros de acuerdo a las nuevas necesidades.

Tras realizar la integración, toda la información que es recopilada por Suricata, se podrá visualizar con un formato normalizado para poder analizarla, esta parte de normalización cobra mucho sentido cuando se hace uso de

diferentes aplicaciones, para que se entienda mejor se mostrará un ejemplo.

Partiendo de un evento generado por Suricata, la información que devolvería en el `log` de Suricata (`eve.json`) sería el que aparece en la figura 4.21, y ese mismo evento ya normalizado visto desde Kibana aperece en la figura 4.22.

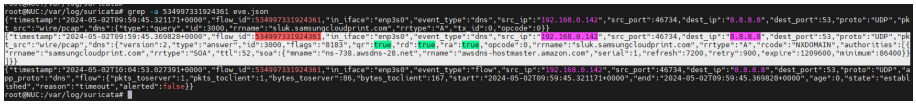

Figura 4.21: Ejemplo de evento en log de Suricata (eve.json)

observer.type	ids
observer.vendor	OISF
related.ip	[192.168.0.142, 8.8.8.8]
source.address	192.168.0.142
source.ip	192.168.0.142
source.port	46,734
suricata.eve.dns.authorities.rrname	samsungcloudprint.com
suricata.eve.dns.authorities.rrtype	SOA
suricata.eve.dns.authorities.soa.expire	1,209,600
suricata.eve.dns.authorities.soa.minimum	86,400
suricata.eve.dns.authorities.soa.mname	ns-738.awsdns-28.net
suricata.eve.dns.authorities.soa.refresh	7,200
suricata.eve.dns.authorities.soa.retry	900
suricata.eve.dns.authorities.soa.rname	awsdns-hostmaster.amazon.com
suricata.eve.dns.authorities.soa.serial	1
suricata.eve.dns.authorities.ttl	52
suricata.eve.dns.id	3,000
suricata.eve.dns.opcode	0
suricata.eve.dns.rcode	NXDOMAIN
suricata.eve.dns.rrname	sluk.samsungcloudprint.com
suricata.eve.dns.rrtype	A
suricata.eve.dns.type	answer
suricata.eve.event_type	dns
suricata.eve.flow_id	534997331924361

Figura 4.22: Ejemplo del evento de Suricata normalizado en ELK

En este ejemplo se puede observar como en `eve.json`, la IP origen del evento aparece como `src_ip`, pero al llegar a ElasticSearch lo transforma

a `source.ip`. También se puede observar en la captura que además de la información normalizada del evento de Suricata, también se añaden otros metadatos, pero estos están más orientados a la posterior gestión de este evento dentro de todos los eventos almacenados en el servidor ELK (agente, fecha de ingesta, etc).

4.4. Análisis de datos

Tras tener todos los eventos incorporados a ElasticSearch, el siguiente paso será el de analizar la información, y así, poder detectar posibles problemas dentro del tráfico que se producen en la red. Para este próposito, Kibana tiene el apartado de "Dashboard", en él se pueden encontrar diferentes visualizaciones para las integraciones que se han instalado, además de algunas para la gestión/monitorización del propio entorno ELK.

Para el caso de Suricata, existen 2 dashboards ya prediseñados. Uno sería el de "[Logs Suricata Events Overview]" (figura 4.23), en el que se puede observar la siguiente información:

- Diagrama de barras temporal, en el que se agrupan a lo largo del tiempo, la cantidad de eventos detectados por Suricata y clasificados por el protocolo asociado al evento.
- Listado de los equipos que más eventos han generado.
- Varios gráficos circulares en función de tipos de eventos, protocolos de transporte y protocolos de red.
- Listado de mayor a menor en función del país origen y otro del país destino.
- Listado completo de los eventos.

Otro sería el de "[Logs Suricata Alerts Overview]"(figura 4.24), en este caso solo aparecerán eventos que sean alertas, es decir, aquellos que Suricata ha marcado con un cierto nivel de riesgo, y por tanto, susceptibles de su revisión. La información que aparece en este dashboard sería:

- Listado de los hosts que han generado más alertas.
- Listado de mayor a menor en función del país origen y otro del país destino.

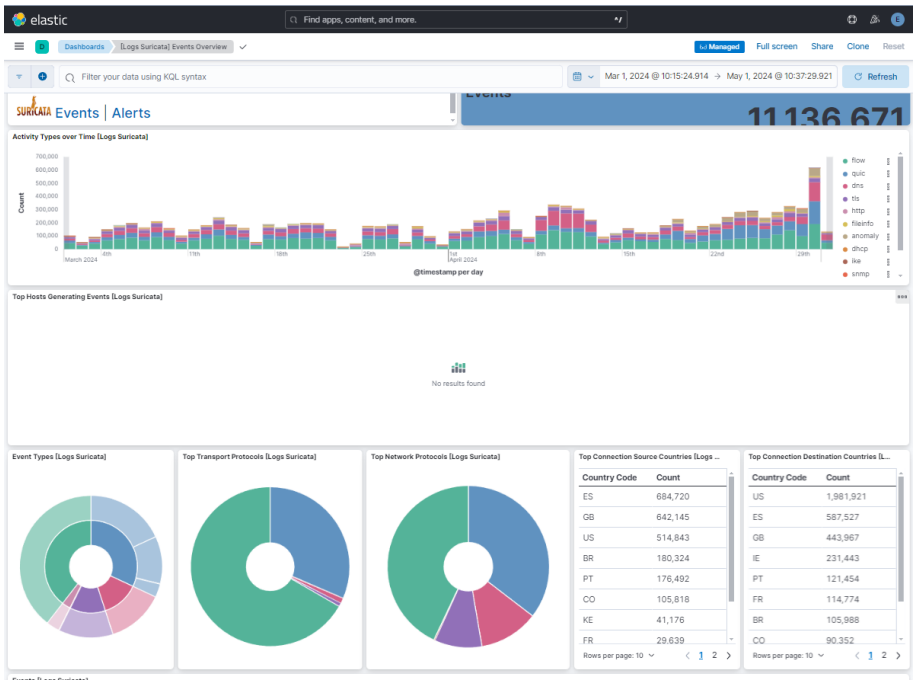

Figura 4.23: [Logs Suricata Events Overview]

- Listado de mayor a menor en función de la firma asociada a la alerta y el tipo de categoría al que pertenece.
- 2 mapas mundiales con puntos de diferente tamaño, en función del número de alertas de acuerdo al origen o al destino.
- Listado completo de los eventos.

En todos los dashboard de Kibana se pueden establecer filtros, para interactuar con la información que aparece y poder analizarla según se requiera en cada análisis. El primer filtro que se debe aplicar sería el filtro temporal, es decir, el rango de tiempo que queremos revisar, para ello se puede establecer el periodo de dos maneras:

- **Periodo relativo**: que serían los últimos eventos ocurridos en las últimas horas, días, semanas... pudiendo elegir el valor (Ej: últimas 12 horas).
- **Periodo absoluto**: en el que podemos elegir la fecha y hora de inicio, y la fecha y hora de fin. En la figura 4.23, se puede ver un ejemplo de filtrado entre 2 fechas.

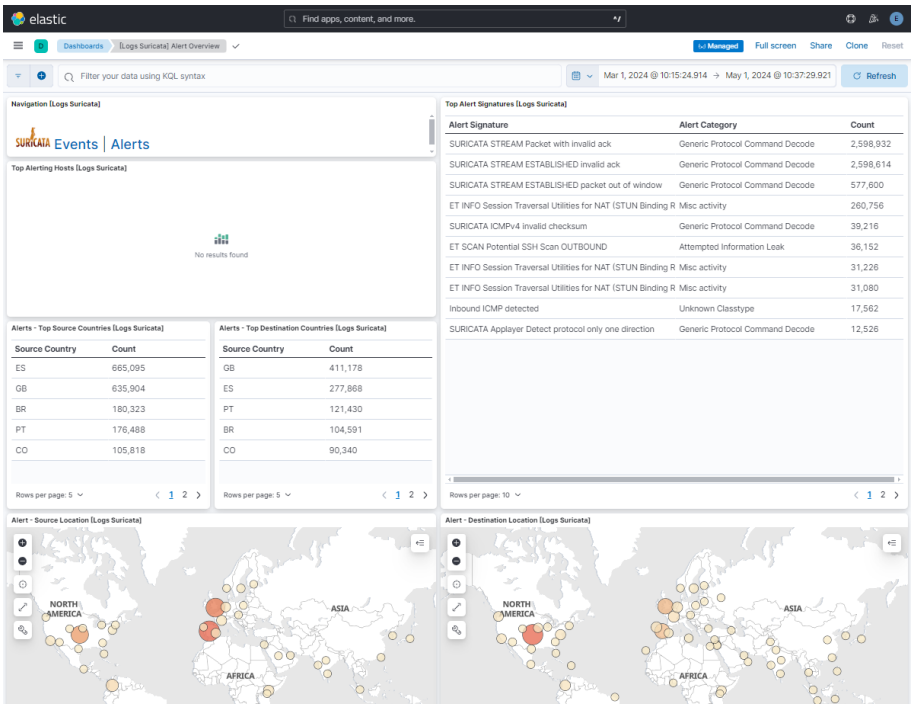

Figura 4.24: [Logs Suricata Alerts Overview]

Otro de los filtros que se puede aplicar, es en función de los valores de los diferentes atributos asociados a cada evento. Con este tipo de filtros, las opciones o combinaciones que existen son bastante amplias, por lo que se explicará las diferentes maneras de aplicarlos pero sin profundizar.

La opción más sencilla sería a partir de los datos que aparecen en el propio dashboard, pudiendo seleccionar, en alguno de los gráficos o tablas, el valor que se busca dentro de los eventos a analizar. Se verá con un ejemplo para que se entienda mejor, como por ejemplo “Protocolo HTTP” (figura 4.25).

Como se puede observar en la figura 4.25, nos da la opción de filtrar por solo el valor seleccionado (“Filter for”) o eliminar este valor de la lista de valores (“Filter out”).

Además del filtrado en función de la información reflejada por el propio dashboard, se puede realizar un filtrado más avanzado utilizando expresiones en el campo que aparece en la parte superior del dashboard. Este filtrado es más complejo, ya que se puede hacer uso de cualquiera de los campos creados por Elastic, por lo que no solo los reflejados en el dashboard, por tanto, se podría incluso filtrar por campos que sean propios de Elastic, como podría ser la versión del agente de Elastic que está reportando la información (ver figura 4.27):

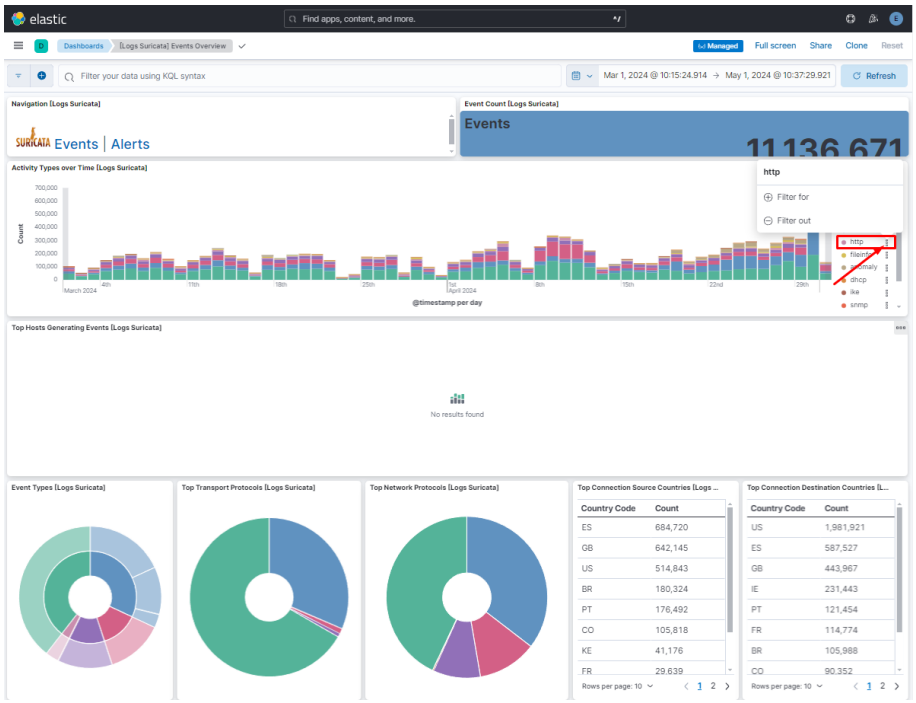

Figura 4.25: Filtrado de acuerdo al valor de un atributo

Este filtrado se puede personalizar combinando más variables y haciendo uso de operadores lógicos (ver figura 4.28)

Además de los dashboards que vienen por defecto en Kibana, se pueden crear también personalizados. Para ello habrá que acceder a la parte de Dashboard y seleccionar la opción de "Create Dashboard". Para reducir la cantidad de información con la que se trabajará, se establecerá un filtro para que solo se muestre la información relativa a los datos de Suricata, eliminando de este modo toda la información al estado de los servicios de Elastic (ver figura 4.29).

Lo siguiente será crear una visualización, que son las diferentes zonas de información (gráficos, diagramas de barras, tablas...) que aparecían en los dashboard precreados los cuales se han visto anteriormente. Como primera visualización se creará una tabla que cuente el número de eventos ocurridos durante un rango de tiempo, agrupados por el grado de severidad asignado por Suricata al tipo de firma relacionada. En la figura 4.30, se puede apreciar en el lado izquierdo la cantidad de campos que están disponibles para utilizar en la visualización (solo relativos al dataset de Suricata, que es el primer filtro aplicado), siendo en este caso de 323 posibles campos, de ahí que exista una gran cantidad de opciones. En este caso se ha seleccionado

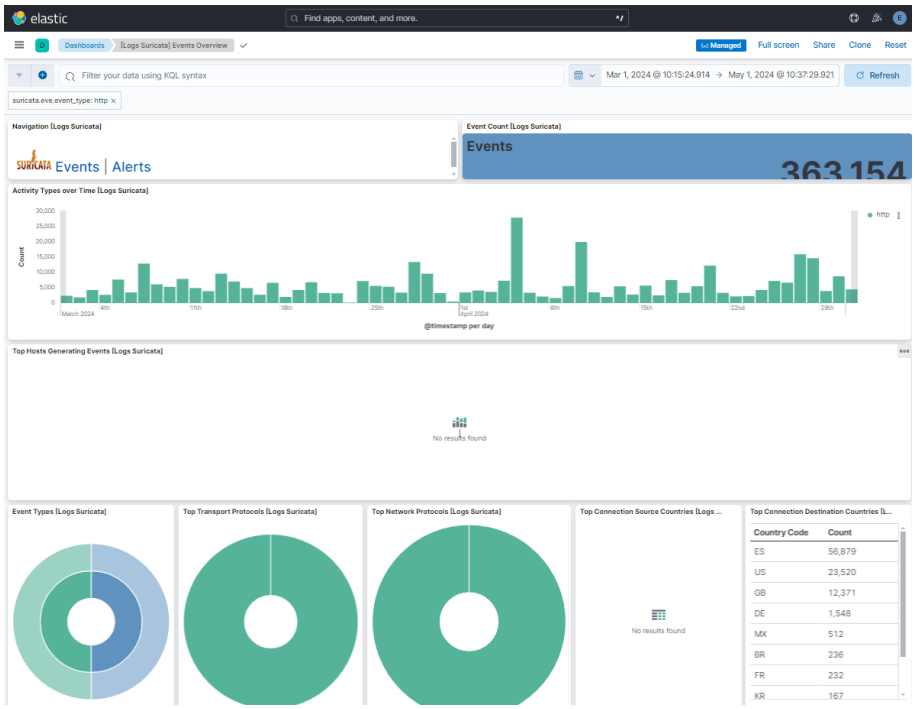

Figura 4.26: Ejemplo de filtrado de acuerdo al valor de un atributo

los campos “Records” y “event.severity”, que hacen referencia a los eventos y a la severidad, respectivamente.

Después de haber creado esta primera visualización, lo siguiente sería crear más visualizaciones que estén relacionadas con esta primera, para posteriormente hacer uso del filtrado que se vió anteriormente (Filter In y Filter Out) y llegar al grado de detalle que se quiera. En la figura 4.31, aparece un ejemplo de una visualización con diagrama de barras que muestra las 5 IPs de la LAN que están involucradas en más eventos, esto requiere un filtro dentro de esta visualización, para que solo incluya las IPs comprendidas dentro del rango LAN (192.168.1.0-192.168.1.255). En este ejemplo, se ha hecho uso de los campos “related.ip” y “Records”.

Otra opción para añadir visualizaciones al dashboard, es haciendo uso de “Add from library” el cual permite añadir visualizaciones de las que está precreadas por ejemplo se puede añadir “Events [Logs Suricata]” que está incluido dentro de los dashboards por defecto de Suricata, de este modo podemos ver el listado de eventos con información relevante de cada uno de los eventos.

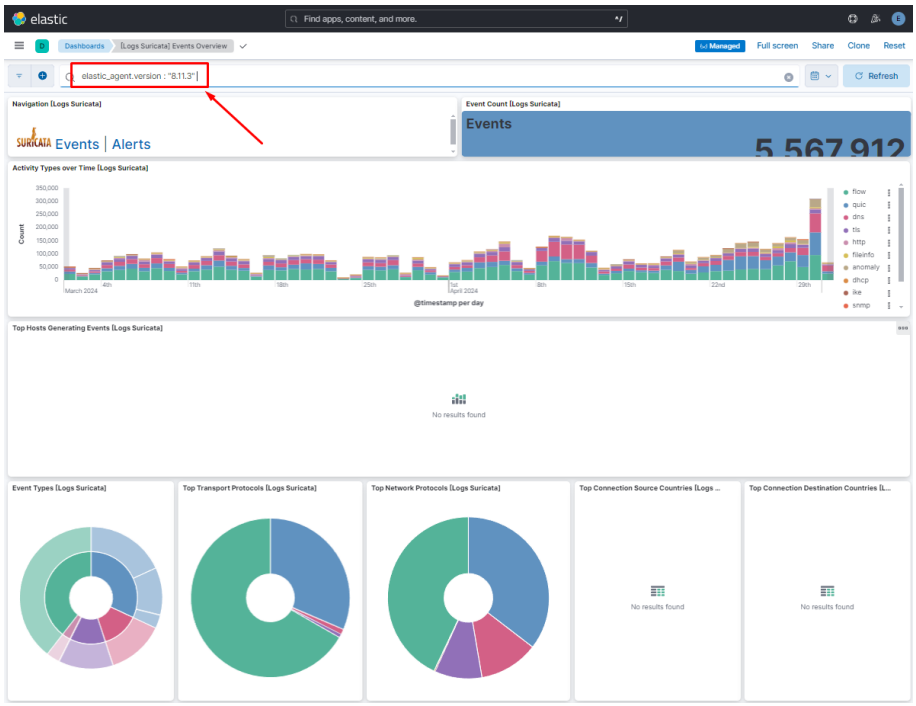

Figura 4.27: Ejemplo de filtro avanzado de acuerdo a la versión de Elastic Agent

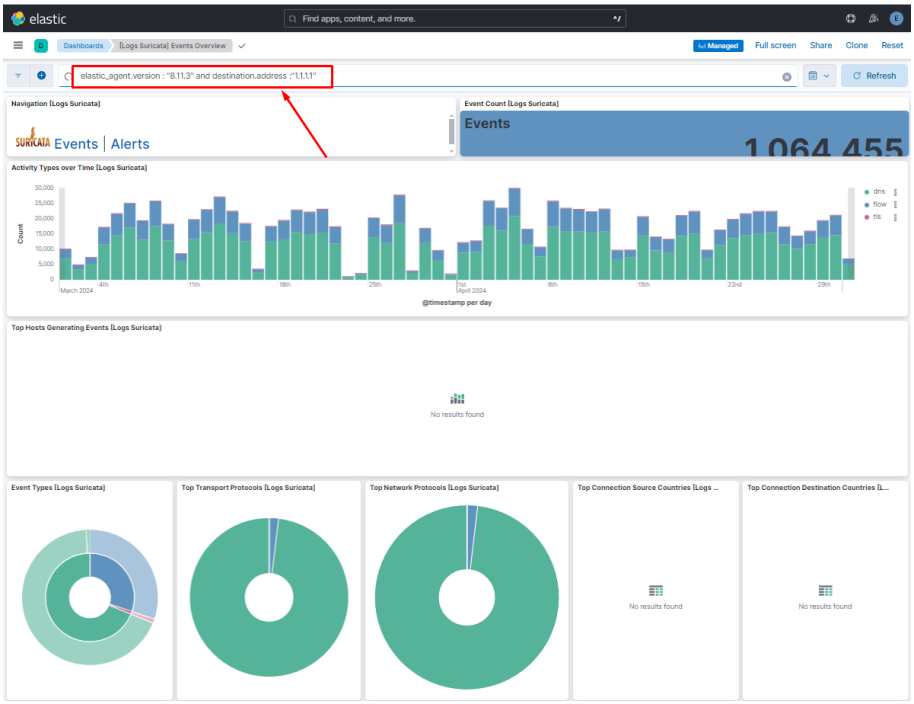

Figura 4.28: Ejemplo de filtro avanzado utilizando operadores lógicos

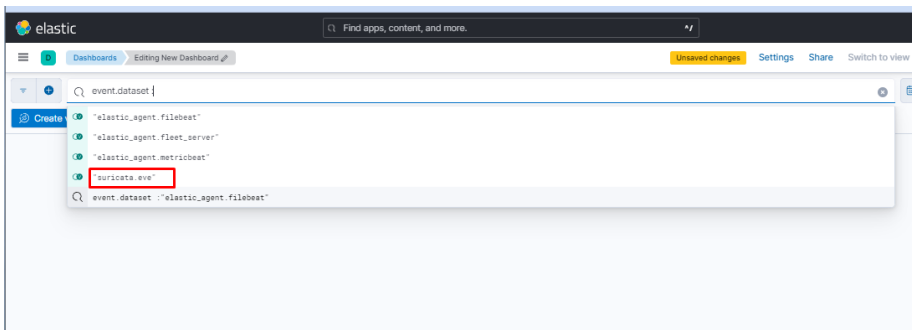

Figura 4.29: Filtrado de datos provenientes de Suricata

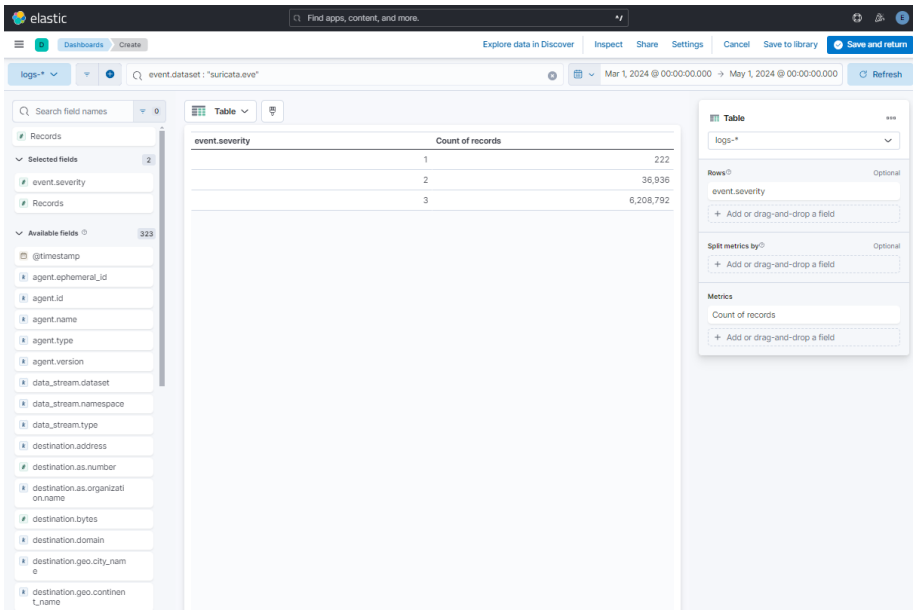

Figura 4.30: Creación tabla de eventos por severidad

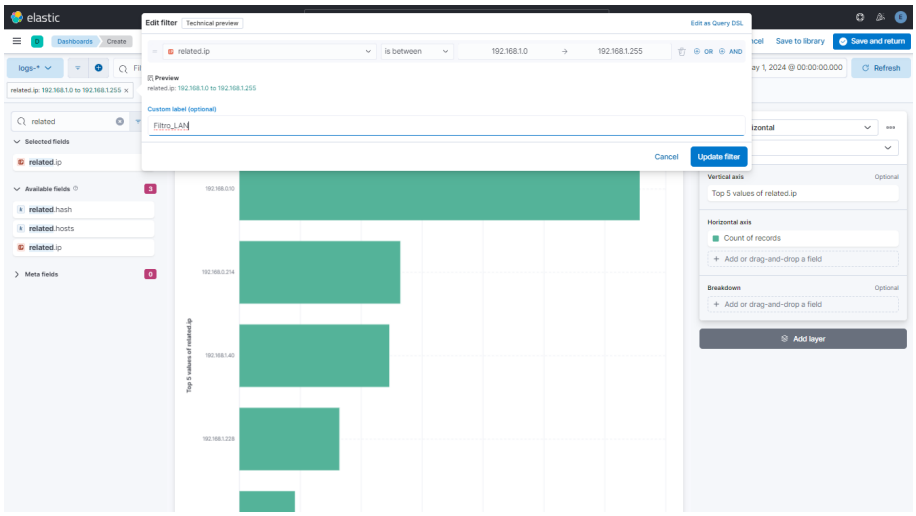

Figura 4.31: Creación tabla de eventos por severidad

Capítulo 5

Experiencia de uso

En este capitulo, se explicará como a partir de la creación de varios dashboards se ha realizado un análisis de las alertas indicadas por Suricata, para determinar si existen riesgos dentro de la red local.

5.1. Servicio UPnP en router de proveedor

Durante los primeros análisis, se vio que algunas alertas marcadas por Suricata se repetían de manera consecutiva cuando sucedían, por lo que se decidió cambiar uno de los atributos usados en los dashboards. Concretamente se cambió el atributo "Records" por el de "suricata.eve.flow_id", ya que Suricata asigna un identificador para cada comunicación abierta entre 2 equipos, y así todos los mensajes intercambiados entre estos 2 equipos se considerarían como una única alerta.

Lo primero que se ha hecho es filtrar en el dashboard los eventos con el valor de severity más alto (mayor criticidad), en el periodo analizado el valor más alto que se obtuvo es de Severity igual a 3, que de acuerdo a la categorización de Suricata estaría asociado a un valor "high" de riesgo.

A continuación se filtró por la IP de origen de LAN que tenía más alertas, ya que a priori puede ser el que tenga más probabilidades de problemas de Seguridad. En el escenario implementado la IP con más alertas de severidad 3, es la IP 192.168.0.10, la cual se corresponde con un ordenador de mesa. El destino al cual tenía mayor número de alertas era la IP 192.168.0.1, que se corresponde con el router de proveedor 5.1, con las siguientes descripciones de alertas

Otro de los detalles interesantes que se observó es que las comunicaciones iban dirigidas contra el puerto 5431 del router de proveedor. Tras realizar una búsqueda en internet, se encontró que el puerto 5431 está asociado al

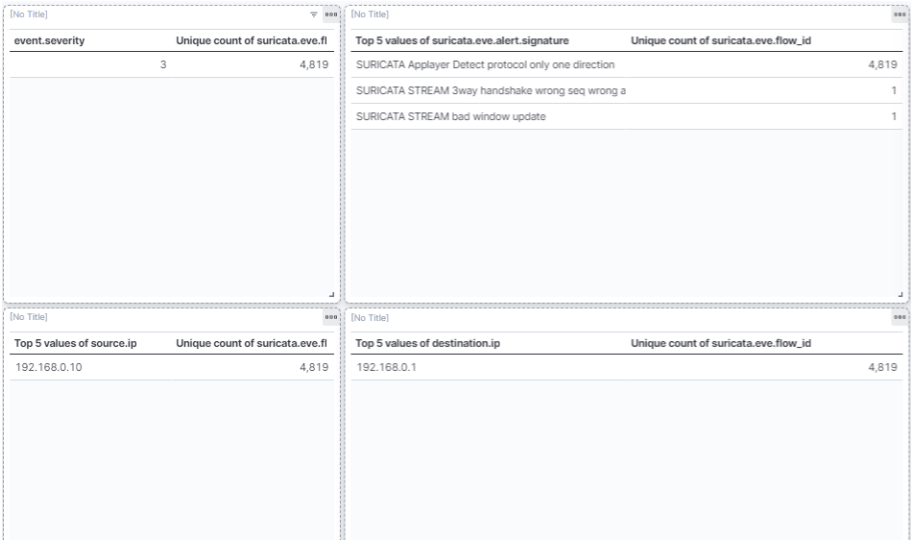

Figura 5.1: Eventos con severidad 3 entre PC y Router proveedor

servicio UPnP en algunos routers (dependiendo del fabricante), el cual puede presentar problemas de seguridad, por lo que tras revisar la configuración del router de proveedor, y ver que el servicio UPnP estaba activo, se decidió deshabilitarlo.

5.2. Ping desde LAN hacia Internet

Tras eliminar del dashboard los eventos analizados en el apartado anterior, se pasó a analizar el siguiente tipo de alerta con severidad 3, "Outbound ICMP detected", el cual avisa de que se está haciendo un ping desde un equipo de la red local hacia internet. Para llevar a cabo un analisis adaptado a este tipo de alerta, se creó un nuevo Dashboard, filtrando por suricata.eve.alert.signature="Outbound ICMP Detected", y también se añadió un filtro en el que "related.ip" estuviese dentro del rango LAN (192.168.0.0-192.168.0.255). En cuanto a las visualizaciones, se crearon 3 tablas (ver figura 5.2):

- Tabla con las 10 IPs origen con más alertas
- Tabla con las 10 IPs destino con más alertas
- Tabla con los 20 destinos más usados, agrupados por el nombre de la organización asociada a la IP pública (destination.as.organization.name)

Esta última tabla es bastante importante, ya que por ejemplo Microsoft dispone de un gran número de direcciones IPs públicas, por lo que conviene agruparlas por el propietario de las IPs.

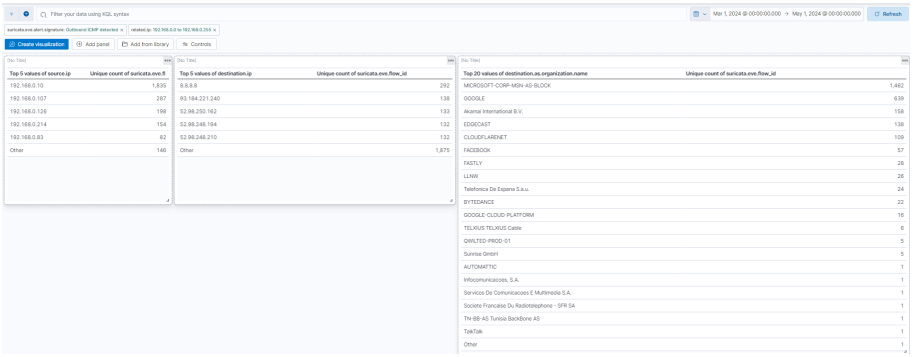

Figura 5.2: Dashboard ICMP LAN hacia Internet

Aunque no supone un riesgo en sí mismo, se puede usar generalmente para conocer la IP pública con la que se está conectando el equipo a Internet y posteriormente realizar algún ataque. La IP de LAN con más mensajes ICMP generados era nuevamente la IP 192.168.0.10, pero tras revisar los destinos a los que realizaba los ICMPs, se observó que estos eran mensajes conocidos (realizados con motivos laborales). Por lo que se eliminó esta IP del listado y se revisó el resto. Tras analizar los datos obtenidos a priori no se detectó ningún ICMP hacia ninguna IP que fuese sospechosa, ya que la 72 % del tráfico ICMP estaba dirigido a los servidores de Google, y entorno a un 7 % hacia Microsoft y otro 7 % hacia CloudFlare.

5.3. Suricata ICMPv4 invalid checksum

Otra de las alertas que más notificaciones daba son la de “SURICATA ICMPv4 invalid checksum” y la de “SURICATA ICMPv4 unknown type”, tras buscar información relacionada con estas alertas de Suricata, se encontró un hilo en el que se indicaba que este error venía por la gestión multihilo de la tarjeta de red en la que Suricata está instalado, ya que muchas tarjetas de red ofrecen esta opción por defecto para tener más rápidez a la hora de gestionar el tráfico, pero esto supone un inconveniente para un entorno en el que se va a usar como IDS, por lo que desde Suricata se recomienda deshabilitarlo (Open Information Security Foundation (2024a)). Por tanto, se decidió eliminar estas alertas, ya que en principio no suponen ningún problema de seguridad, aunque sería conveniente seguir las recomendaciones de Suricata, para no tener que eliminar estas alertas.

5.4. DNS over HTTPS

Entre el listado de alertas también existían varias que hacían referencia DNS over HTTP, este es un protocolo que se creó para poder encriptar/camuflar las peticiones a servidores DNS, ya que DNS no tiene grandes medidas de seguridad para evitar ataques man-in-the-middle, es por ello que se desarrollo un protocolo para poder enviar de manera segura las peticiones a los servidores DNS, si poder observar la petición en sí. Esto se convierte en un arma de doble filo, ya que de cara un usuario legitimo es menos susceptible de ataques DNS Poisoning (sitios web falsos), y por el otro lado, se pueden tener grandes problemas de seguridad, ya que al ir encriptada la petición DNS no se puede saber a que URL intenta acceder el usuario, por lo que no se puede hacer uso de blacklist de sitios web maliciosos (Cimpanu, 2019).

Suricata advierte sobre estas alertas y los cataloga como “ET INFO” que quiere decir que es una alerta para Informar, y que corresponde a Emerging Threats (Amenazas Emergentes), por lo que es importante revisar este tipo de alertas, y tomar las medidas oportunas para reducir los riesgos. Al ser un tema bastante complejo y no se dispone de mucho tiempo, no se profundizará más en este apartado.

Capítulo 6

Conclusiones

En este capitulo se detallarán las conclusiones finales de este TFM

6.1. Conclusiones finales

- Cada vez más se conectan dispositivos a la red de "casa" que no ofrecen posibilidad de instalar herramientas de seguridad, como podría ser un antivirus. Este hecho hace que la instalación de una solución NIDS+SIEM que monitorice la red local, y que genere alertas (aunque sean falsos positivos), ayuda a ofrecer un mayor grado de seguridad.

- A lo largo de este TFM, se ha podido apreciar el esfuerzo que está haciendo Elastic para hacer más accesible el uso de sus herramientas y su integración con herramientas de terceros, para que todo este proceso sea más sencillo en usuarios que se inician con el uso de sus herramientas. Al inicio de este trabajo, se empezó con la versión 7.13 la cual no integraba Fleet y Elastic Agent (aunque sí aparecían como funcionalidades beta), a partir de la versión 7.14 ya se incluyeron ambas funcionalidades. Pero hasta que no pasó un tiempo desde el lanzamiento de la versión 7.14, estas funcionalidades no aparecían en los tutoriales que hay en internet, por lo que no se hizo uso de ellas en el TFM desde el principio.

- A pesar de lo indicado en el punto anterior, como se ha podido ver en el capitulo 5, para poder hacer un análisis de alertas realista (número reducido de falsos positivos), requiere de una curva de aprendizaje que permita ir poco a poco afinando tanto algunos parámetros de configuración de las herramientas (Suricata y ELK) como de las reglas.

- Tras revisar la gran mayoría de las alertas generadas, a priori no se ha encontrado ningún problema de seguridad, lo cual puede deberse a la concienciación sobre ciberseguridad que existe en el hogar por parte de todos los habitantes, o de las medidas de seguridad que hay implementadas en los dispositivos de mayor uso y que ofrecen dicha posibilidad. O cabe la posibilidad que debido a la falta de experiencia sobre la solución implementada no se haya sido capaz de detectar estos problemas de seguridad.

- Uno de los problemas más recurrentes a lo largo de la fase de implementación, ha sido cuando se actualizaba alguna de las herramientas de ELK, generando problemas de integración de herramientas o la necesidad de configurar algún parámetro nuevo que en versiones anteriores no era necesario. Principalmente los parámetros nuevos que han ido surgiendo con los upgrade de versión en la mayoría de ocasiones han sido parámetros orientados a la seguridad del propio stack, como por ejemplo, uso de certificados entre herramientas para la encriptación de los mensajes o la autenticidad entre emisor y receptor. Por lo que también se ha podido ir viendo la evolución de la obligatoriedad marcada por Elastic para hacer uso de estas medidas de seguridad (muchas existían anteriormente pero era opcionales).

- Se ha revisado el tráfico que el ordenador personal se usaba para realizar el teletrabajo, y se ha podido concluir que al hacer uso de la VPN empresarial, todo el tráfico que se generaba desde el equipo, viajaba encriptado a través de la VPN, incluso el tráfico hacia internet o la red doméstica. Por lo que todos los equipos de la red local dejaban de ser accesibles (impresora, NUC, otros PCs...). Así que todo el tráfico generado por el equipo no ha podido ser monitorizado, siendo la única opción la de instalar un agente en el propio ordenador para que guarde los logs, y cuando vuelva a tener conectividad con servidor ELK (al apagar la VPN empresarial) se envíen para su posterior análisis. Aún así, el tráfico generado por este equipo y que sí ha podido ser monitorizado, tras un primer estudio no ha presentado signos de problemas de seguridad, por lo que, inicialmente, no supone una amenaza para la empresa.

6.2. Posibles puntos de mejora

- **Monitorización en tiempo real**: debido a las limitaciones de los recursos de los que se disponía, la solución SIEM se instaló en una ma-

quina virtual que no siempre estaba encendida, por lo que tras volver a encender la máquina virtual requería de un tiempo para ingestar toda la información obtenida por Suricata, y ser procesada. Este tiempo aumentaba en función del tiempo que hubiese estado apagada la máquina virtual, en algunos casos llegó hasta 30 minutos hasta tener toda la información procesada. Por ello, sería interesante instalar ELK en un equipo que estuviese siempre encendido.

- **Integración de NIDS y SIEM en un único equipo**: Al tener ambas herramientas integradas en un único equipo, reduce el tráfico de red y se puede limitar a que las conexiones entre las diferentes herramientas este limitada al localhost, reduciendo así el riesgo de accesos desde otras IPs, habilitando solo el acceso a Kibana desde otros host, aunque se podría valorar también la posibilidad de que Kibana solo fuese accesible desde el propio equipo.

- **Notificaciones en tiempo real**: Kibana ofrece la posibilidad de enviar alertas por correo cuando se cumple ciertos criterios que se establezcan. En este TFM no se ha analizado esta parte, ya que como se ha comentado anteriormente, Kibana estaba instalada sobre una maquina virtual que no siempre estaba encendida, por lo que carecía de sentido, pero que en una solución para poner en producción puede ser bastante interesante.

- **Identificación de equipos**: para poder hacer un seguimiento a lo largo del tiempo de cada equipo, es conveniente usar direcciones IPs estáticas o el uso de DHCP estático de acuerdo a la MAC de cada equipo que se conecte a la red doméstica. Durante el trabajo, solo existían algunos equipos de la red que tenían IPs fijas a lo largo del tiempo, pero con la evolución del trabajo se fueron asignando a cada equipo IPs fijas haciendo uso del DHCP estático del router de proveedor. Pero sería interesante valorar otras soluciones de DHCP estático, y que a la vez permita hacer uso de un blacklist de sitios webs y aumentar la seguridad. Una posible solución para este punto podría ser el uso de Dnsmasq.

- **Uso de nombres en vez de IPs**: a pesar de que se fueron haciendo asignaciones de IPs fijas a cada equipo con la evolución del trabajo, uno de los problemas era saber a que equipo correspondía la IP, siendo necesario el uso de una tabla con la relación IP con equipo. Aunque no se ha tenido tiempo durante este trabajo, puede que esto sea viable realizarlo desde el propio Kibana, aunque si se implementase una

solución para el punto anterior puede que también se pudiese asignar nombres a los equipos desde esta solución.

- **Revisión de la configuración de Suricata**: Como se vió durante uno de los apartados del capitulo 5, la configuración por defecto de Suricata genera falsos positivos en determinados tipos de tráfico por el procesamiento en paralelo de la tarjeta de red, por lo que sería necesario analizar más en detalle este problema y la posible solución.
- **Analizar la posibilidad de usar Elastic Security**: en las últimas versiones de ELK, dado que cada vez es más usado como SIEM, Elastic ha desarrollado un módulo específico para la parte de SIEM y análisis de seguridad (Elastic Security), por lo que podría ser interesante analizar más en detalle si aporta beneficios sobre la solución implementada o si también es gratuita.

Bibliografía

AT&T (2020). *Open Source IDS Tools: Comparing Suricata, Snort, Bro (Zeek), Linux*.
https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview.

AT&T (2024). *AlienVault OSSIM*.
https://cybersecurity.att.com/products/ossim.

Cimpanu, Catalin (2019). *DNS-over-HTTPS causes more problems than it solves, experts say*.
https://www.zdnet.com/article/dns-over-https-causes-more-problems-than-it-solves-experts-say/.

Cisco Systems (2024). *Web Snort*.
https://www.snort.org/.

Copper, Stephen (2024). *The Best Network Intrusion Detection Systems Software & NIDS Tools*.
https://www.comparitech.com/net-admin/nids-tools-software/.

Dnsmasq (2024). *Dnsmasq*.
https://dnsmasq.org/doc.html.

Dnsstuff (2019). *10 Best Free and Open-Source SIEM Tools*.
https://www.dnsstuff.com/free-siem-tools.

Elastic (2024). *Elastic*.
https://www.elastic.co/es/.

Gartner (2024). *2024 Gartner® Magic Quadrant™ for SIEM*.
https://www.splunk.com/en_us/form/gartner-siem-magic-quadrant.html.

IBM (2024). *¿Qué es la SIEM?*.
https://www.ibm.com/es-es/topics/siem.

Instituto Nacional de Ciberseguridad (2020). *¿Qué son y para qué sirven los SIEM, IDS e IPS?*.
https://www.incibe.es/empresas/blog/son-y-sirven-los-siem-ids-e-ips.

Open Information Security Foundation (2024a). *Packet Capture*.
https://docs.suricata.io/en/latest/performance/packet-capture.html.

Open Information Security Foundation (2024b). *Web Suricata*.
https://suricata.io/.

Paxson, Vern (2024). *Web Zeek*.
https://en.wikipedia.org/wiki/Suricata_(software).

Splunk (2024). *Splunk products*.
https://www.splunk.com/en_us/products.html.

Wikipedia (2024). *Wikipedia*.
https://es.wikipedia.org/wiki/.