UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

Implementación de un centro de operaciones de seguridad (SOC) de código abierto con elementos de red para sistemas industriales.

Mònica Martínez Gómez

Tutor: José Enrique López Patiño

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2020-21

Valencia, 12 de septiembre de 2021

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

VLC/ CAMPUS
VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS
Gestión ambiental verificada
REG.NO. ES-CV- 000030

Resumen

Los últimos años han sido críticos en cuanto a la visibilización y concienciación que supone la ciberseguridad actualmente para cualquier elemento que se pueda encontrar conectado a una red y en especial a las infraestructuras denominadas críticas llegando a convertirse en una prioridad. En este proyecto se pretende implementar un centro de operaciones de seguridad con herramientas de código abierto que abarcará desde los elementos de red necesarios para el control de tráfico hasta la interfaz de usuario donde se podrán visualizar las alertas pasando por todos los elementos necesarios para integrar estas dos (correladores, herramientas de parseo de logs, dashboards, gestión de alertas). Una vez definidos todos los elementos de necesarios para el correcto funcionamiento se procederá a montar la infraestructura necesaria en contenedores docker con la intención de que el resultado a implementar llegue de la manera más sencilla al usuario.

Resum

Els ultims anys han sigut critics en quant a la visibilització i conscienciació que suposa la ciberseguretat actualment per a quasecol element que es puga trobar connectat a una xarxa i en especial les infrestructures anomenades critiques arribant a convertir-se en una prioritat. En aquest projecte es prentén implementar un centre d'operacions de seguretat amb ferramentes opensource que abastarà des dels elements de xarxa necessaris per al control del tràfic fins a la interfície d'usuari on es podran visualitzar les alertes passant per tots els elements necessaris per a integrar aquestes dues (correladors, ferramentes de parseig de logs, dashboards, gestio d'alertes). Una volta definits tots els elements necessaris per al correcte funionament es procedirà a muntar l'infraestrucura necessària en dockers amb la intenció de que el resultat a implementar arribe de la manera mes senzilla a l'usuari.

Abstract

The last few years have been critical in terms of the visibility and awareness that cybersecurity currently supposes for any element that can be found connected to a network and especially to so-called critical infrastructures, becoming a priority. In this project we intend to implement a security operations center with opensource tools that will range from the network elements necessary for traffic sniffing to the user interface, where the alerts can be viewed, through all the elements necessary to integrate these two (correlators, log parsing tools, dashboards, alert management). Once all the necessary elements have been defined for the correct functioning, the necessary infrastructure will be assembled in docker containers with the intention that the result to be implemented will arrive in the simplest way to the user.

Agradecimientos

Trasmitir mi mas sincero agradecimiento a mis amigos y compañeros que me han seguido durante este largo proceso y han colaborado con este proyecto. En primer lugar, a Damià por ser el mejor *sysadmin* que conozco y siempre saber la solución a todos los errores sin buscar en Google y, en segundo lugar, a Rocío por saber como indicarme en todo momento como continuar y superar un momento de crisis. Finalmente agradecer a mi gata Coccolotti por ser mi animal de soporte y mi compañera de TFG favorita.

Índice

Capítulo 1. Introducción

1.1 Motivaciones

El ámbito de la ciberseguridad informática ha ido tomando un lugar importante durante las últimas décadas hasta convertirse en una parte fundamental de las empresas actualmente. Para ello muchas de estas se han visto obligadas a modificar y adecuar sus activos e infraestructura para poder adaptarse a los estándares mínimos establecidos por los organismos certificadores junto al impacto económico que esto supone.

En este TFG se pretende crear desde cero un mecanismo de control de red para la monitorización y detección de anomalías en redes de carácter industrial abarcando todos los elementos que serán necesarios para el funcionamiento completo de un centro de operaciones de seguridad SOC mediante herramientas de código abierto. Todo esto será presentado en contenedores Docker que facilitarán el despliegue e integración en un entorno que se encuentra ya en producción.

La importancia de los centros de operaciones es crucial a la hora de la detección temprana y erradicación de anomalías en la red. Una detección de amenaza identificada en las primeras horas de un ataque cibernético puede evitar la distribución de este en la red y, junto con una contención rápida y eficaz, se minimizaría el impacto de este sobre el correcto funcionamiento de la empresa evitando así ataques de ramsonware o de fuga de información.

Un ejemplo actual de la necesidad de un centro de operaciones de seguridad junto con los elementos de red necesarios y una serie de reglas de detección de anomalías es el ataque que ha sufrido la multinacional Accenture durante el mes de agosto. La carencia de metodologías y reglas de detección adecuadas ha llevado a la mayor fuga de información que ha sufrido la empresa junto con la distribución del ramsomware LockBit teniendo como consecuencia el paro casi completo de la empresa y la necesidad de una vuelta atrás para poder reestablecer los servicios mínimos junto con las consecuencias económicas que esto conlleva. Por otra parte, la cantidad de información exfiltrada y publicada en internet afecta tanto a la propia empresa como a clientes y proveedores suponiendo un impacto grave sobre la imagen y socios de la empresa.

Para llevar a cabo este proyecto se han escogido herramientas de código abierto para poder proporcionar un sistema de detección funcional sin necesidad de realizar una gran inversión económica y así poder facilitar a quien lo necesite los servicios mínimos de detección y monitorización de red.

1.2 Objetivos

El objetivo principal del proyecto es la creación de un entorno SOC completo de fácil implementación en redes operacionales (OT) que no se encuentren monitorizadas.

Para el desarrollo del proyecto *SOC (Centro de operaciones de Seguridad) de código abierto con elementos de red para sistemas industriales* se han escogido herramientas de licencia libre. Esto es así ya que este tipo de herramientas proporcionan a las organizaciones la facilidad de introducir medidas completas de monitorización sin la gran inversión monetaria inicial que supone la adquisición del software necesario para la integración de la seguridad dentro de la organización de la empresa.

La finalidad de este TFG es proporcionar la información y herramientas necesarias para mejorar la seguridad de una organización que no disponga de elementos de seguridad de manera sencilla. Para ello se proporcionarán los conocimientos y herramientas necesarias para el control y alerta de amenazas de seguridad tanto en elementos de red como de punto final.

Con la intención de demostrar lo crucial que es de cara a un incidente de seguridad los SOC en redes corporativas se simularán diversos ataques para observar si la implementación llevada a cabo durante el proyecto podría alertar al equipo de seguridad de las vulnerabilidades de seguridad ejecutadas dentro del entorno simulado.

Capítulo 2. Introducción a los centros de operaciones de seguridad (SOC)

2.1 Definición

Un centro de operaciones de seguridad (SOC) [1] es un centro de mando para un equipo de profesionales de la tecnología de la información (IT) con experiencia en seguridad de la información (infosec) que supervisa, analiza y protege a una organización de los ciberataques.

Los SOC son una parte integral para minimizar los costes de una posible violación de datos, ya que no sólo ayudan a las organizaciones a responder a las intrusiones con rapidez, sino que también mejoran constantemente los procesos de detección y prevención.

La mayoría de las grandes organizaciones cuentan con SOC internos, mientras que las empresas que no cuentan con el personal o los recursos necesarios para mantener uno por sí mismas pueden optar por externalizar algunas o todas las responsabilidades del SOC a un proveedor de servicios gestionados (MSP).

Existen diferentes tipos de SOC que pueden ser implementados según las necesidades y características de la empresa [2]:

- **SOC dedicado o autogestionado**. Este modelo tiene una instalación en las instalaciones con personal interno.
- **SOC distribuido**. También conocido como SOC cogestionado, este modelo cuenta con miembros del equipo semidedicados a tiempo completo o parcial que son contratados internamente para trabajar junto a un proveedor de servicios de seguridad gestionados por terceros (MSSP).
- **SOC gestionado**. Este modelo cuenta con MSSPs que proporcionan todos los servicios SOC a una empresa.
- **SOC de mando**. Este modelo proporciona conocimientos de inteligencia sobre amenazas y experiencia en seguridad a otros centros de operaciones de seguridad, normalmente dedicados. Un SOC de mando no participa en las operaciones o procesos de seguridad reales, sólo en la parte de inteligencia.
- **Centro de fusión**. Este modelo supervisa cualquier instalación o iniciativa centrada en la seguridad, incluidos otros tipos de SOC o departamentos de TI. Los centros de fusión se consideran SOCs avanzados y trabajan con otros equipos de la empresa, como operaciones de TI, DevOps y desarrollo de productos.
- **SOC multifunción**. Este modelo tiene una instalación dedicada y personal interno, pero sus funciones y responsabilidades se extienden a otras áreas críticas de la gestión de IT, como los centros de operaciones de red (NOC).
- **SOCaaS**. Este modelo basado en la suscripción o en el software subcontrata algunas o todas las funciones del SOC a un proveedor en la nube.

En el SOC, el tráfico de Internet, las redes, los ordenadores de sobremesa, los servidores, los dispositivos *endpoint*, las bases de datos, las aplicaciones y otros sistemas se examinan continuamente en busca de signos de un incidente de seguridad. El personal del SOC puede trabajar con otros equipos o departamentos, pero este suele ser autónomo, con empleados que tienen conocimientos de alto nivel de IT y ciberseguridad, o se subcontratan estos roles a proveedores de servicios externos. La mayoría de los SOC funcionan las 24 horas del día, con empleados que trabajan por turnos para registrar constantemente la actividad y mitigar las amenazas.

Los SOC se han construido normalmente en torno a una arquitectura radial, en la que un sistema de gestión de eventos e información de seguridad (SIEM) agregadores y conectores de los datos de las fuentes de seguridad. Los radios de este modelo pueden incorporar una variedad de sistemas, como soluciones de evaluación de vulnerabilidades, sistemas de gestión de riesgo y cumplimiento (GRC), escáneres de aplicaciones y bases de datos, sistemas de prevención de intrusiones (IPS), análisis del comportamiento de usuarios y entidades (UEBA), detección y corrección de puntos finales (EDR) y plataformas de inteligencia de amenazas (TIP)

El SOC suele estar dirigido por un director de SOC, y puede incluir personal de respuesta a incidentes, analistas de SOC (niveles 1, 2 y 3), cazadores de amenazas y directores de respuesta a incidentes. El SOC depende del CISO, que a su vez depende del CIO o directamente del CEO de la empresa.

Los SOC se encuentran habitualmente en los sectores de la sanidad, la educación, las finanzas, el comercio electrónico, la administración pública, las operaciones militares y la tecnología avanzada.

2.2 Funciones del SOC

La función de un equipo de operaciones de seguridad y, con frecuencia, de un centro de operaciones de seguridad (SOC), es vigilar, detectar, investigar y responder a las ciberamenazas las 24 horas del día. Los equipos de operaciones de seguridad se encargan de supervisar y proteger tanto los activos, como la propiedad intelectual, los datos del personal, los sistemas empresariales y la integridad de la marca. Como componente de implementación del marco general de ciberseguridad de una organización, los equipos de operaciones de seguridad actúan como punto central de colaboración en los esfuerzos coordinados para supervisar, evaluar y defenderse de los ciberataques.

La estrategia general de un centro de operaciones de seguridad se centra en la gestión de amenazas. Esto supone la recopilación de datos y el análisis de estos para detectar actividades sospechosas con el fin de aplicar medidas de mejora en la seguridad en una organización. Los datos en bruto que supervisan los equipos del SOC son relevantes para la seguridad. Estos se recogen de los cortafuegos mientras que la información sobre amenazas proviene de los sistemas de prevención y detección de intrusiones (IPS/IDS), las sondas y los sistemas de gestión de eventos e información de seguridad (SIEM). Se diseñan alertas para comunicar inmediatamente a los miembros del equipo si alguno de los datos es anormal o muestra indicadores de compromiso (IOC) [3].

Las responsabilidades básicas de un equipo SOC son las siguientes [4]:

- **El descubrimiento y la gestión de activos.** Estas tareas implican obtener un alto conocimiento de todas las herramientas, el software, el hardware y las tecnologías utilizadas en la organización. También se centran en garantizar que todos los activos funcionen correctamente y sean parcheados y actualizados con regularidad.
- **La supervisión continua del comportamiento** incluye el examen de todos los sistemas las 24 horas del día, los 7 días de la semana, durante todo el año. Esto permite a los SOC dar la misma importancia a las medidas reactivas y proactivas, ya que cualquier irregularidad en la actividad se detecta al instante. Los modelos de comportamiento entrenan a los sistemas de recogida de datos sobre qué actividades son sospechosas y pueden utilizarse para ajustar la información que podría registrarse como falsos positivos.
- **Mantener registros de actividad.** Esto permite a los miembros del equipo SOC retroceder o localizar acciones anteriores que puedan haber dado lugar a una anomalía. El SOC debe registrar todas las comunicaciones y actividades de una empresa.
- **La clasificación de la gravedad de las alertas** ayuda a los equipos a garantizar que las alertas más graves o urgentes se traten primero. Los equipos deben clasificar regularmente las amenazas de ciberseguridad en términos de daño potencial.
- **El desarrollo y la evolución de las medidas de seguridad** es importante para ayudar a los equipos del SOC a estar al día. Los equipos deben crear un plan de respuesta a incidentes (IRP) para defender los sistemas contra ataques nuevos y antiguos. Los equipos también deben ajustar el plan según sea necesario cuando se obtenga nueva información.
- **La recuperación de incidentes mediante copias de seguridad** permite a una organización recuperar los datos comprometidos. Esto incluye reconfigurar, actualizar o hacer copias de seguridad de los sistemas.
- **El mantenimiento del cumplimiento** es clave para garantizar que los miembros del equipo del SOC y la empresa sigan las normas reglamentarias y organizativas al llevar a cabo los planes de negocio. Normalmente, un miembro del equipo supervisa la legislación y la adecuación del cumplimiento de las medidas legales necesarias.
- **Las capacidades adicionales del SOC** podrían incluir la ingeniería inversa, el análisis forense, la telemetría de red y el criptoanálisis, según las necesidades específicas de la organización.

En la siguiente figura se puede ver de forma esquematizada el funcionamiento de un SOC tras la detección de una amenaza teniendo en cuenta las diferentes fuentes y activos anteriormente nombrados.

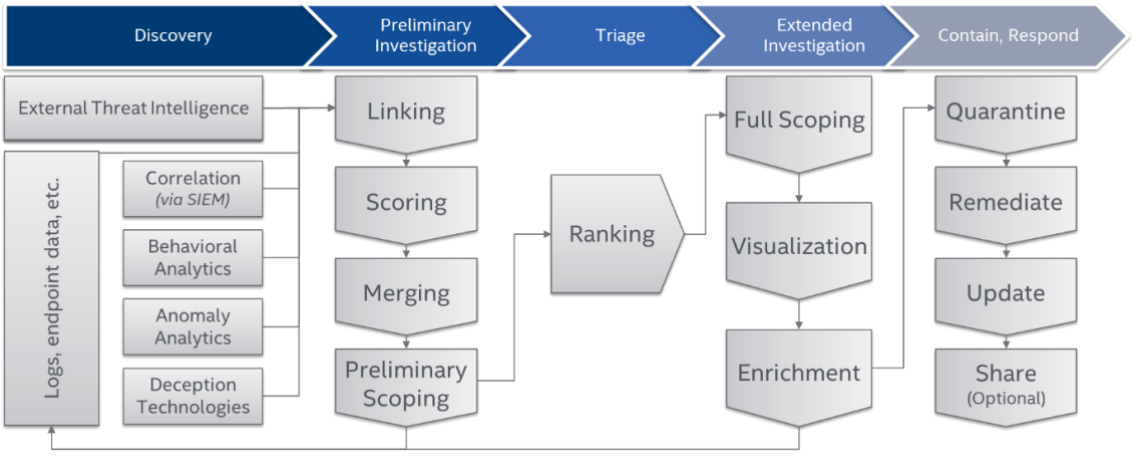

Figura 1. Esquema de funcionamiento de un SOC [4]

2.3 Herramientas de monitorización

Las herramientas empleadas como parte de la capacidad de un SOC son fundamentales para garantizar su capacidad de vigilar y detectar activamente las últimas amenazas. Constantemente surgen nuevas tecnologías para las diferentes áreas de los SOC, por lo que es probable que las tecnologías que actualmente proporcionan mejores resultados terminen quedando obsoletas en un periodo corto de tiempo.

Las herramientas desplegadas por el SOC deben mantenerse actualizadas y gestionarse adecuadamente para asegurar el correcto funcionamiento de este. El conjunto de herramientas no sólo debe tener en cuenta los avances tecnológicos del mercado, sino, lo que es más importante, los cambios en las necesidades de la empresa. La implementación de una herramienta específica dentro del SOC se deberá considerar:

- Las razones empresariales y técnicas de la implantación de nueva tecnología y el impacto de estos cambios sobre la organización.
- La compatibilidad de las nuevas herramientas con las otras ya implementadas en la empresa.
- Hay que asegurar que el personal del SOC está formado en el uso de las nuevas herramientas.
- El equipo SOC ha de tener una comprensión estratégica de cómo las nuevas herramientas contribuyen a las necesidades generales del SOC

2.3.1 Tecnología de gestión de eventos e información de seguridad- SIEM

SIEM son las siglas en inglés de la gestión de eventos e información de seguridad y proporciona a las organizaciones la detección, el análisis y la respuesta ante las posibles amenazas de seguridad en la empresa.

El software SIEM combina la gestión de la información de seguridad (SIM) y la gestión de eventos de seguridad (SEM) para proporcionar un análisis en tiempo real de las alertas de seguridad generadas por las aplicaciones y el hardware de la red. El software SIEM coteja los eventos con las reglas de los motores de análisis y los indexa para realizar búsquedas en menos de un segundo con el fin de detectar y analizar las amenazas avanzadas utilizando la inteligencia recopilada globalmente. Esto proporciona a los equipos de seguridad tanto una visión como un registro de las actividades dentro de su entorno de IT, proporcionando la capacidad de análisis de datos, correlación de eventos, agregación, informes y gestión de registros.

El software SIEM proporciona una serie de características y beneficios, entre ellos:

- Agregación en un único punto de múltiples fuentes de datos.
- Cuadros de mando personalizados y gestión del flujo de trabajo de las alertas.
- Integración con otros productos.

Las principales funciones de un SIEM son las siguientes:

- **Agregación de datos**: Las soluciones SIEM necesitan datos de múltiples fuentes como parte de la agregación de datos, trasladando la información a un único lugar. Los sistemas SIEM recogen los datos por sí mismos o utilizan reenviadores para enviar los registros de otros sistemas al sistema SIEM.
- **Colectores SIEM**: Los colectores pueden configurarse para obtener datos de registro de los equipos de la red, pudiendo conectarse directamente al sistema y obtener los registros.
- **Transmisores SIEM**: Los SIEM *Forwarders* pueden enviar datos de registro a la solución SIEM desde el equipo, ya que implican una instalación de software, conocida como agente que puede reenviar eventos a la solución SIEM.
- **Correlación de eventos de seguridad**: La correlación de eventos de seguridad consiste en tratar de detectar patrones en los datos recogidos por los sistemas para encontrar identificadores de actividad maliciosa en la red. Si se descubren patrones sospechosos, se marcan, lo que permite a los analistas de seguridad investigar más a fondo y tomar medidas correctivas.
- **Análisis avanzado**: El análisis avanzado puede incluir el análisis del comportamiento mediante el análisis de los datos recogidos en el SIEM para detectar cambios de comportamiento inesperados dentro de la red empresarial o el estudio de una serie de acciones realizadas por un usuario consideradas como indicadores de un posible ataque.
- **Automatización del SOC**: Las capacidades de automatización del SOC de un sistema SIEM pueden actuar sobre los eventos de seguridad haciendo el análisis como lo haría el analista del SOC, pudiendo así dar un resultado. Por ejemplo, un evento de seguridad es analizado por la solución SIEM y, basándose en criterios, la solución SIEM determina que este evento es un incidente grave y, en consecuencia, plantea un incidente en el sistema de gestión de incidentes con una prioridad crítica.

- **Cuadros de mando e informes**: Todas las soluciones SIEM disponen de cuadros de mando para facilitar la visualización del panorama de las amenazas y ofrecerles indicadores de lo que está ocurriendo en los sistemas de los que el SIEM está recopilando datos. Estos cuadros de mando también permiten la elaboración de informes para poder ver el número de amenazas identificadas durante un periodo definido, lo que permite a las organizaciones determinar el nivel de amenazas al que se han enfrentado a lo largo del tiempo.
- **Búsqueda de amenazas**: Utilizando las herramientas de análisis de búsqueda proporcionadas por el SIEM, los analistas de seguridad pueden consultar los datos del SIEM y determinar el impacto en la organización durante los últimos meses o incluso años permitiendo la búsqueda de nuevos patrones y amenazas que van surgiendo diariamente con las publicaciones de nuevos *exploits* y vulnerabilidades.
- **Análisis forense o respuesta a incidentes**: Cuando las organizaciones descubren que han sufrido un ataque, necesitan determinar rápidamente cuándo se produjo la brecha, qué acciones se llevaron a cabo y si los responsables siguen dentro de la red, es decir, siguen dentro de los sistemas de la organización. El análisis forense permite analizar los datos recogidos durante un periodo de tiempo e intentar determinar la gravedad de los acontecimientos que condujeron a la violación, incluidos los ataques iniciales, el momento de la primera violación y las actividades realizadas después de entrar en los sistemas de la organización.

Las mejores herramientas SIEM actualmente según *Gartner* son Splunk, IBM QRadar, Exabeam, LogRythm, Securonix, Rapid7 y Dell Technologies (RSA), por nombrar algunas. Estas herramientas SIEM suelen estar clasificadas como líderes en el *Gartner Magic Quadrant* que se publica cada año. [5]

2.3.2 Software de monitorización de seguridad

Mientras que la monitorización de la red proporciona una recopilación de datos para el análisis de los flujos de tráfico básicos, la estructura general y la integridad de sus sistemas, la monitorización de la seguridad de la red protege de las numerosas vulnerabilidades y explotaciones.

Incluso más importante que la monitorización general, la supervisión de la seguridad analiza una multitud de factores complejos (carga útil de la red, protocolos de red, comunicaciones cliente-servidor, sesiones de tráfico cifrado, patrones de tráfico y flujo de tráfico) para alertar a los analistas de actividades maliciosas conocidas en un intento de contener un ataque.

La herramienta de supervisión adecuada le ofrece un servicio permanente de vigilancia a un entorno empresarial en busca de amenazas y comportamientos sospechosos. Los administradores y analistas pueden entonces investigar y calibrar los patrones anormales de los usuarios y tomar las medidas adecuadas.

A diferencia de la supervisión operativa de la red, la supervisión de la seguridad de la red y los analistas que la utilizan también deben ser capaces de detectar intrusiones y todas las formas de ataque incluidas las amenazas nuevas, de día cero y de última generación, para poder tomar decisiones basadas en estas acciones anómalas.

Ningún experto en seguridad puede garantizar una protección del 100% frente a los ataques, pero las nuevas tecnologías de supervisión y análisis continuos de la red proporcionan niveles de apoyo a la detección y mitigación que pueden disminuir considerablemente la posibilidad de un ataque o una brecha. Aquellos que puedan aprovechar la supervisión, el análisis y la corrección continuos de la seguridad de la red en tiempo real también se beneficiarán de una reducción del tiempo de detección y de la capacidad de reducir o evitar drásticamente los daños resultantes.

Es importante señalar que un atacante tarda sólo unos minutos en comprometer y exfiltrar datos. Por lo tanto, la calidad de un sistema de supervisión de la seguridad de la red equivale a la velocidad con la que se notifica el tráfico sospechoso a los analistas.

Los elementos de monitorización de seguridad se pueden dividir principalmente entre elementos de monitorización de red y elementos de monitorización de equipos.

Los principales elementos de monitorización de red son:

- **Cortafuegos**: Es un dispositivo de seguridad de red que supervisa el tráfico de red entrante y saliente y permite o bloquea los paquetes de datos basándose en un conjunto de reglas de seguridad. Su objetivo es establecer una barrera entre la red interna y el tráfico entrante de fuentes externas como Internet para bloquear el tráfico malicioso. [6]
- **IDS/IPS**: Es un sistema que supervisa el tráfico de la red en busca de actividad sospechosa y alerta cuando se descubre dicha actividad. Aunque la detección de anomalías y la elaboración de informes son las funciones principales, algunos sistemas de detección de intrusiones (IPS) son capaces de tomar medidas cuando se detecta actividad ilícita o tráfico anómalo, llegando a provocar el bloqueo del tráfico enviado desde direcciones IP maliciosas. Un IDS únicamente supervisa los paquetes de red en busca de tráfico de red potencialmente malicioso sin ejecutar acciones sobre este. [7]
- **Proxy de navegación**: Herramienta que traduce el tráfico entre redes o protocolos. Es un servidor intermediario que separa a los usuarios de los destinos por los que navegan. Los servidores proxy ofrecen diferentes niveles de inspección de red, privacidad y seguridad según el tipo de tráfico, la política o las necesidades de la empresa. [8]
- **Sondas**: Es la tecnología encargada de monitorizar la red y detectar comportamientos maliciosos. Detecta ataques y programas malignos mediante el análisis red en tiempo real por medio de las reglas implementadas en la misma. Para su correcto funcionamiento necesitará un duplicado del tráfico que se desee analizar, habitualmente se encuentran en elementos críticos de las redes industriales ya que permiten el análisis detallado a bajo nivel de los protocolos utilizados por componentes de este tipo de redes. [9]

Por otro lado, las principales herramientas de monitorización de equipos de punto final son:

- **Detección y respuesta a los puntos finales (EDR):** También conocida como detección y respuesta a las amenazas en los puntos finales (ETDR), es una solución de seguridad integrada para los puntos finales que combina la supervisión continua en tiempo real y la recopilación de datos de los puntos finales con capacidades de respuesta y análisis automatizados basados en reglas. Investiga actividades sospechosas en los equipos, empleando un alto grado de automatización para permitir a los analistas de seguridad identificar y responder rápidamente a las amenazas. [10]
- **Sistema de monitorización de procesos:** Permite en equipos Windows la monitorización y el registro de la actividad del sistema en el registro de eventos de Windows. Proporciona información detallada sobre la creación de procesos, las conexiones de red y los cambios en el tiempo de creación de archivos del equipo.

Capítulo 3. Diseño e implementación del centro de operaciones de seguridad (SOC)

3.1 Introducción

El siguiente capítulo busca detallar tanto el proceso de creación y diseño de un centro de operaciones de seguridad (SOC) mediante el uso de herramientas de código abierto como las medidas necesarias para segmentar correctamente una red industrial en la cual se introducirá el sistema de monitorización.

La idea de que la implementación de herramientas de seguridad aporta suficiente para una correcta securización de las organizaciones es errónea. Tal y como se ha indicado en los objetivos este TFG se pretende mejorar la seguridad de la empresa, es por esto por lo que, para una correcta monitorización y control de la red esta debe seguir las indicaciones detalladas en la norma IEC 62443 para redes operacionales y la ISO 27000 para las redes técnicas.

En los siguientes puntos se detallarán las herramientas que se han utilizado y se especificarán las configuraciones aplicadas para el correcto funcionamiento de los diferentes elementos de monitorización introducidos en un entorno industrial.

Se detallarán como debe segmentarse los diferentes tipos de redes para garantizar los mínimos exigidos por las normas anteriormente citadas.

Todas las configuraciones y archivos necesarios para la implementación de este TFG se encuentran en el ANEXO externo debido al volumen de datos.

3.2 Análisis de los objetivos

La seguridad de una organización engloba diferentes aspectos a los que hay que prestar especial interés. Si bien ya ha sido comentado anteriormente un sistema completo de monitorización de activos de red es importante al igual que también lo es un correcto diseño de las redes. Es por esto por lo que durante el análisis de los objetivos se han incluido las medidas necesarias tanto para introducir todo un sistema de monitorización de seguridad como las indicaciones necesarias para un diseño correcto de redes operacionales.

A continuación, se detallarán las medidas básicas de seguridad implementadas en este TFG para introducir la monitorización de activos en un entorno corporativo.

Los puntos que desarrollar basándose en los objetivos fijados en este TFG son:

- Diseño y segmentación de redes operacionales
- Diseño de la centralización de los diferentes archivos de log y despliegue en la infraestructura.
- Sistema de detección de anomalías en la red
    - SIEM
    - Paneles de visualización
    - Reglas y alertas de seguridad

3.3 Fase de diseño

Durante la fase de diseño se ha buscado garantizar la mayor securización de una red industrial mediante la manera más sencilla de implementación de los activos y modificaciones necesarias para conseguirlo.

Para ello se ha dividido en dos puntos claves Diseño de la segmentación de una red industrial y Diseño de la centralización, visualización y monitorización de logs

Finalmente se detallarán las especificaciones técnicas necesarias a tener en cuenta durante el dimensionamiento de las herramientas durante el proceso de introducción en una red industrial en el apartado Especificaciones técnicas de las herramientas.

3.3.1 Diseño de la segmentación de una red industrial

La seguridad de los sistemas de control industrial (ICS) es más importante que nunca en el entorno actual. Las normas de ciberseguridad de los sistemas de control industrial, como NERC CIP, ISA-99 (IEC-62443) y NIST 800-82, se han desarrollado para ayudar a identificar e implementar las mejores prácticas de seguridad de los ICS.

De acuerdo a la normativa de seguridad IEC-62443 [11], así como muchos expertos en ciberseguridad, coinciden en tres elementos clave para implementar redes industriales seguras.

- **Arquitectura de red**: La piedra angular del diseño de una red industrial segura es emplear un enfoque de defensa en profundidad que incluya zonas y comunicaciones seguras entre las diferentes subredes. Diseñar un acceso remoto seguro y fiable se está convirtiendo rápidamente en una necesidad desde un punto de vista práctico.

- **Configuración segura de los dispositivos:** Con las redes convergentes de hoy en día, la configuración de los dispositivos de la red industrial es cada vez más complicada, y las funciones de seguridad se desactivan con frecuencia para facilitarla. Como resultado, las redes ICS son vulnerables a violaciones tanto deliberadas como involuntarias.

- **Gestión de la seguridad de la red**: Un sistema competente de gestión de la seguridad de la red no sólo le ayudará a implantar y hacer cumplir las políticas de seguridad en su red ICS, sino que también le permitirá supervisar y registrar los eventos de la red, así como proporcionar alertas de eventos de seguridad en tiempo real.

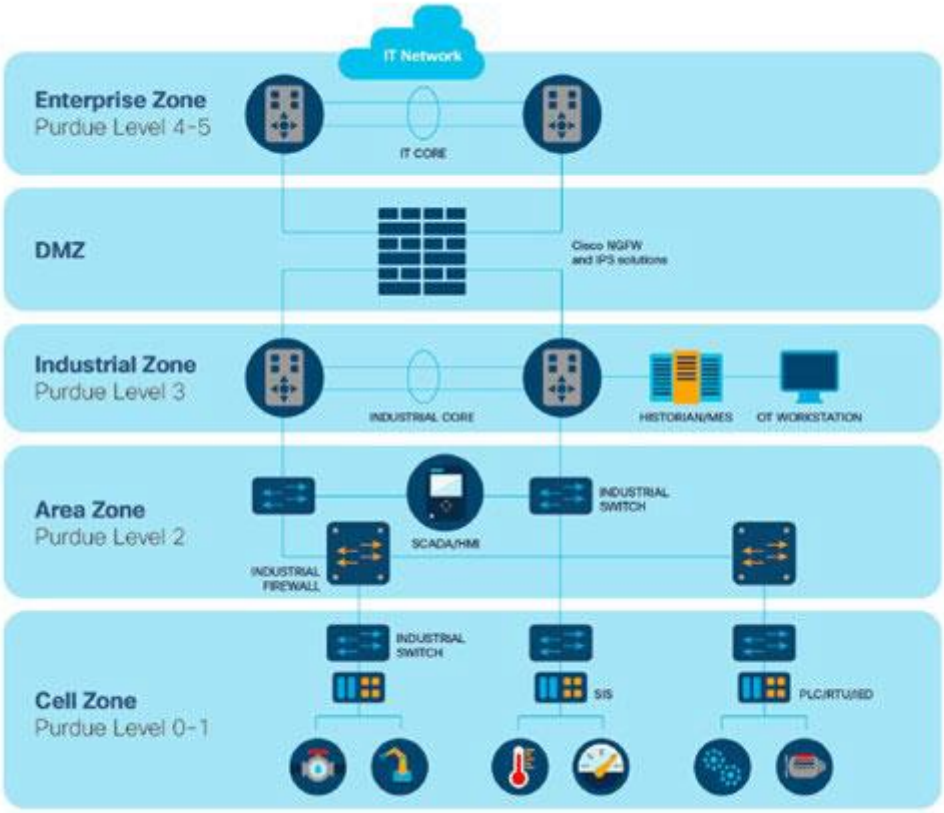

Figura 2. Diferentes zonas seguras de una red operacional [12]

Según la norma la mejor práctica actual para construir una red ICS es utilizar una arquitectura de seguridad de defensa en profundidad tal y como se indica en la Figura 2. Diferentes zonas seguras de una red operacional [12]que divide el tráfico de la red en zonas definidas y sólo permite que determinado tráfico pase entre ellas. Dentro de estas zonas, este diseño permite una comunicación fiable y oportuna, a la vez que limita la posible amplitud de una brecha en una zona determinada.

Una arquitectura de defensa en profundidad puede diseñarse en tres pasos:

- **La segmentación de la red**: La división de la red en zonas físicas o lógicas con requisitos de seguridad similares se conoce como segmentación de la red. La ventaja de segmentar la red es que cada componente puede centrarse en los problemas de seguridad individuales a los que se enfrenta cada sección del ICS. Debido a que cada dispositivo es responsable de un área específica de la red, en lugar de ser responsable de la seguridad de todo el ICS, el uso de la estrategia de segmentación es aconsejable.

- **Definir las interacciones entre zonas**: El tráfico no autorizado se puede filtrar mediante cortafuegos una vez que se haya definido el tráfico preciso que debe pasar por las zonas seguras. La lista blanca del tráfico que debe pasar entre cada zona y el bloqueo del resto del tráfico es una práctica habitual. La inspección profunda de paquetes se utiliza habitualmente en los IDS para filtrar los protocolos industriales a un nivel más fino que los cortafuegos normales. Muchos IDS también incluyen un modo transparente que permite instalarlos en redes existentes sin necesidad de modificar el esquema IP de la red.
    - Otra buena práctica de seguridad para vincular las redes ICS a las redes informáticas de la empresa o a Internet es utilizar un cortafuegos para crear una zona desmilitarizada (DMZ). Aunque no hay conexión directa entre la red ICS protegida y la red de la empresa cuando se utiliza una DMZ, ambas pueden acceder al servidor de datos. Al eliminar la conexión directa entre las redes de ICS y de la empresa, se reduce en gran medida el riesgo de que el tráfico ilegal viaje a través de las zonas, lo que podría poner en peligro la seguridad de toda la red.
- **Acceso remoto seguro**: Por último, en el mercado de los sistemas de control y automatización industrial hay una creciente demanda de supervisión y mantenimiento de las instalaciones a distancia. Esto aumenta en gran medida las posibilidades de que un usuario hostil acceda a la red. Las redes que requieren que el sitio remoto esté conectado al ICS en todo momento deben emplear una red privada virtual (VPN) que utilice un mecanismo de cifrado seguro, como IPsec u OpenVPN, para evitar que usuarios no autorizados accedan a la red. El uso de una VPN tiene tres ventajas principales.
    - La trasmisión de datos va cifrada extremo a extremo.
    - Se exige que tanto el remitente como el destinatario de la sesión verifiquen sus identidades, lo que garantiza que los datos sólo se transfieren entre dispositivos verificados.
    - Puede proteger la integridad de los datos al exigir el cifrado y la autenticación.

Siguiendo estas normas y basándonos en un marco teórico de una red industrial simple determinada por tres zonas.

- Zona de red corporativa de usuarios
- Zona de red corporativa de servidores
- Zona de red industrial
    - Red operacional de supervisión
    - Red operacional de control

Se ha desarrollado la siguiente propuesta de red con elementos de seguridad y segmentaciones basándose en los criterios anteriormente citados y las normas NERC CIP, ISA-99 (IEC-62443) y NIST 800-82

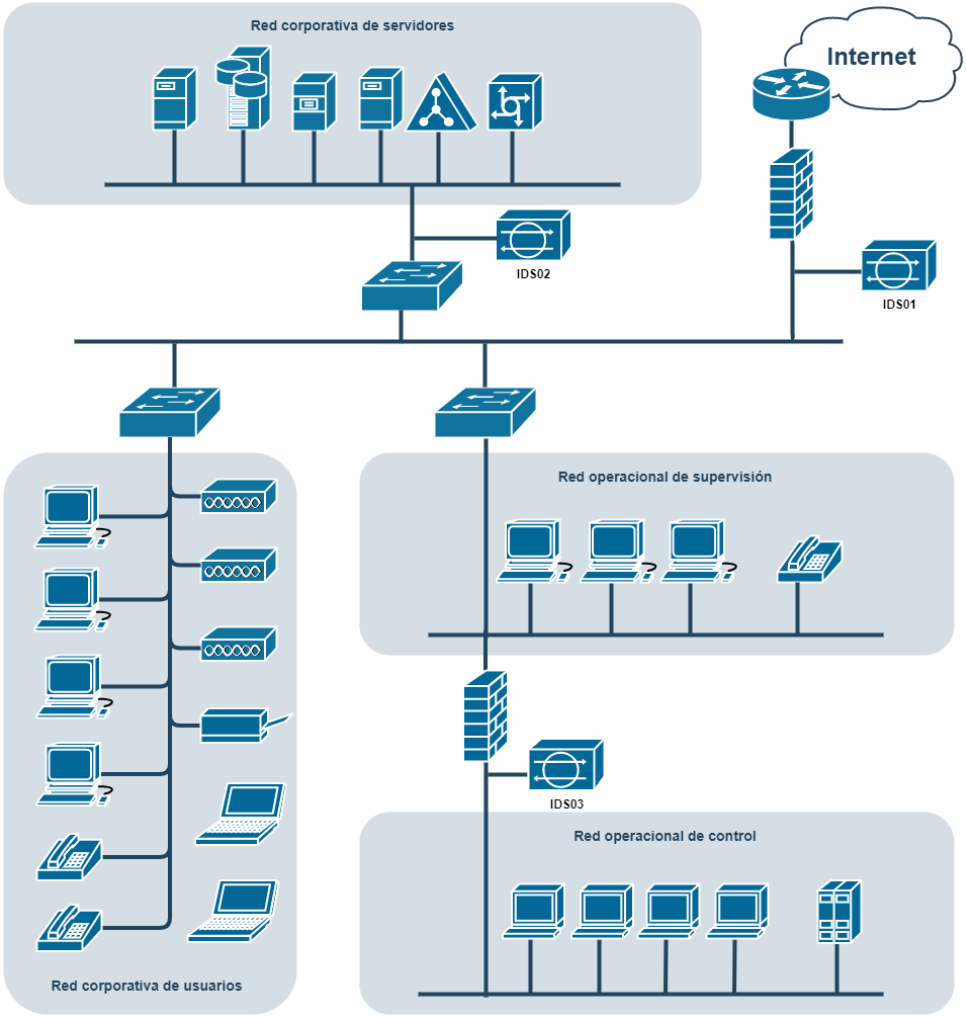

Figura 3. Diseño seguro de red operacional

Los componentes necesarios y segmentaciones de la red industrial de la Figura 3. Diseño seguro de red operacional se detallan a continuación:

- **Perímetro**: rúter de entrada y salida a Internet y firewall para la segmentación de las redes en la organización
- **Red corporativa de servidores**: servicios de la empresa como directorio, servidores de ficheros, ERP, servidor de comunicaciones. Contiene toda la información sensible para la empresa
- **Red corporativa de usuarios**: red de uso habitual para empleados de la empresa que no tengan un rol operacional, por ejemplo: marketing, finanzas...
- **Red operacional de supervisión**: red de uso habitual para empleados de la empresa que tengan un rol operacional, por ejemplo: ingenieros, operadores... proporciona conectividad con el entorno OT aislado del resto de redes
- **Red operacional de control:** red industrial desde la que se operan los PLC, HMIs...
- **IDS01**: monitorización de entrada y salida a Internet y monitorización inter-VLANs en toda la red
- **IDS02**: monitorización intra-VLAN de la red de servidores
- **IDS03**: monitorización de la conexión entre la red IT y la red OT

3.3.2 Diseño de la centralización, visualización y monitorización de logs

Una solución de centralización de logs aporta la visión centralizada de todo lo que está ocurriendo en los sistemas de la organización, y va a permitir tanto generar alertas automáticas basadas en una sucesión de eventos considerados anómalos, como tener disponible toda la información pasada para poder llevar a cabo investigaciones futuras.

Esta solución debe cumplir los siguientes requisitos para ser confiable:

- Garantizar la disponibilidad de los datos en todo momento.
- Ser escalable a cualquier tamaño de organización.
- Restringir el acceso a los datos únicamente a personal autorizado.
- Cifrar los datos *at-rest* y *on-transit* para evitar acceso ilegítimo a la información.
- Establecer diferentes periodos de retención de los datos y tipos de almacenamiento para los mismos.
- Generar alertas en pseudo tiempo real basadas en reglas previamente establecidas.
- Normalizar los datos de las diferentes fuentes en un único esquema.

En base a estos puntos se considera Elastic [13], también conocido como ELK Stack como herramienta de centralización, gestión, visualización y monitorización de eventos. Esta solución es de código abierto y ofrece licencias libres para su uso gratuito, además de ofrecer también licencias de pago con funcionalidades adicionales y soporte a sus usuarios.

La utilización de este paquete de herramientas supone los siguientes elementos:

- Base de datos de centralización de logs Elastic Search [14]
- Sistema de gestión centralizada de agentes de punto final Elastic Agent [15]
- Herramienta de detección y respuesta de puntos finales EDR Endpoint security [16]
- Herramienta de visualización y análisis de eventos Kibana [17]
- Reglas definidas para la detección de anomalías del paquete Security [18]

La solución de monitorización de red se basa en un software IDS o Sistema de Detección de Intrusiones, con la capacidad de analizar el tráfico de red diseccionando los diferentes protocolos de las capas 2 a 7 del modelo OSI.

Este software permite generar en pseudo tiempo real alertas indicando anomalías detectadas en las transmisiones de red o basadas en patrones conocidos de comportamientos maliciosos. El software elegido es Suricata IDS [19] al ser de licencia libre y código abierto.

El software Suricata IDS dispone de múltiples repositorios de código abierto actualizados diariamente con reglas de seguridad tanto de las nuevas amenazas publicadas como de los patrones de comportamientos asociados a actividades maliciosas.

3.3.3 Especificaciones técnicas de las herramientas

En este apartado se aborda el diseño y despliegue de las diferentes soluciones de seguridad para un entorno de demostración que sea representativo de sus capacidades de gestión de información aplicada a la seguridad informática. Este entorno mínimo no es recomendado para su aplicación directa en organizaciones ya que, aunque se cumplen las medidas básicas de seguridad en el entorno, no se garantiza la disponibilidad de los datos al estar almacenados en un único nodo de datos. Tampoco es intención del documento el dimensionamiento del entorno en función del tamaño de la organización, aunque se proveerán estimaciones para poder escalar el entorno a uno aplicable en un entorno real.

Para ambos tipos de hardware se ha elegido el sistema operativo Ubuntu 20.04(LTS) basado en Linux 5.4, en sus dos variantes para arquitecturas x86_64(amd64) y arm64. Ni Suricata IDS ni la máquina para el despliegue del paquete Elastic tiene ningún requisito adicional a la instalación estándar de Ubuntu, por lo que la instalación base es suficiente para ejecutar el software.

3.3.3.1 Dimensionamiento paquete Elastic

La solución Elastic se compone de distintas piezas de software tal y como ya se ha comentado anteriormente, de los cuales en este apartado se va a cubrir la implementación de Elasticsearch, Kibana y Fleet (Elastic Agent). Estos componentes cumplen las siguientes funciones:

- Elasticsearch: motor de base de datos donde se va a almacenar la información.
- Kibana: interfaz gráfica sobre la que realizar las consultas a la base de datos, generar alertas y reportes.
- Fleet: gestión centralizada de los colectores de información.

Para la preparación del hardware se deberá tener en cuenta los siguientes puntos:

- Almacenamiento y gestión de la información requiere hardware que nos permita ser ágiles al consultar la información y que tenga la capacidad de procesar los datos en pseudo tiempo real para evitar la pérdida de información. El software soporta las arquitecturas x86_64(amd64) y arm64. Se ha elegido la arquitectura x86_64(amd64) por su mayor disponibilidad en el mercado y su asequibilidad. El dimensionamiento de recursos para el entorno utilizado es el siguiente:
    - CPU: Procesador x86_64 de 4 núcleos a 3.5GHz.
    - Memoria RAM: 16GB.
    - Almacenamiento: 128GB de almacenamiento rápido SSD.
    - Red: tarjeta de red Ethernet de un puerto Gigabit.

3.3.3.2 Dimensionamiento de Suricata

Para la monitorización de la red se detallará la implementación del software Suricata para la agregación de IDS dentro del entorno siguiendo las recomendaciones detalladas a continuación:

- Escenario 1: Raspberry Pi 4 Model B (arm64)
  - CPU: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz.
  - Memoria RAM: 8GB LPDDR4-3200 SDRAM.
  - Almacenamiento: tarjeta microSD 64GB.
  - Red:
    - Gestión: tarjeta de red WiFi de 2.4 GHz y 5.0 GHz IEEE 802.11ac o tarjeta de red Ethernet de un puerto Gigabit externa conectada por USB3.
    - Datos: tarjeta de red Ethernet de un puerto Gigabit.
  - Capacidad de análisis de tráfico: 500 Mbps

- Escenario 2: Servidor x86_64 (amd64) para 1-4 Gbps
  - CPU: Procesador x86_64 de 4 núcleos y 8 hilos a 3.5GHz.
  - Memoria RAM: 32GB
  - Almacenamiento: 128GB de almacenamiento rápido SSD.
  - Red:
    - Gestión: tarjeta de red Ethernet de un puerto Gigabit.
    - Datos: tarjeta de red Ethernet de cuatro puertos Gigabit.
  - Capacidad teórica de análisis de tráfico: 5 Gbps

- Escenario 3: Servidor x86_64 (amd64) para 10 Gbps
  - CPU: Procesador x86_64 de 20 núcleos y 40 hilos a 3.5GHz
  - Memoria RAM: 128GB
  - Almacenamiento: 1TB de almacenamiento rápido SSD.
  - Red:
    - Gestión: tarjeta de red Ethernet de un puerto Gigabit.
    - Datos: tarjeta de red Ethernet un puerto 10 Gigabit.
  - Capacidad teórica de análisis de tráfico: 12 Gbps

3.4 Herramientas utilizadas

A continuación, se detallarán en los siguientes apartados las herramientas escogidas durante la fase de diseño del proyecto.

3.4.1 Docker

Docker [20] se trata de una plataforma que permite el desarrollo y ejecución de múltiples aplicaciones separadas en una infraestructura para la distribución rápida de software. Esta separación es posible gracias a los contenedores que permiten aislar de manera segura estas aplicaciones.

Para la utilización de esta plataforma se requieren los siguientes objetos, siendo estos:

- **Imágenes:** se trata de plantillas con instrucciones para la creación de contenedores. En ocasiones, las imágenes utilizadas constituyen la personalización de otras imágenes ya existentes. Existen imágenes predefinidas, no obstante, es posible la creación de estas a partir de ficheros denominados *Dockerfiles*.
- ***Dockerfiles:*** son ficheros que con un lenguaje simple permiten la definición de una imagen, así como su ejecución. Estas instrucciones crean capas sobre esta imagen de manera que se pueden aplicar cambios constantes realizando un *rebuild* de la imagen.

- **Contenedores:** son instancias ejecutables de una imagen, independientemente de si estas se tratan de imágenes predefinidas o creadas a partir de un *Dockerfile*. Estos pueden ser creados, iniciados, detenidos o borrados de manera sencilla mediante la línea de comandos. Tal y como se ha explicado previamente, los contenedores se encuentran aislados en un único host. Estos contenedores a su vez pueden estar conectados a una o múltiples redes, así como a unidades de almacenamiento.

3.4.2 Elastic Search

ElasticSearch [14] es un motor de búsqueda *full-text* distribuido, escrito en Java y basado en Lucene. Es capaz de almacenar, buscar y analizar grandes volúmenes de datos en tiempo casi real. Está orientado a ofrecer búsquedas rápidas y complejas sobre datos que cambian poco una vez escritos, como por ejemplo logs. Se la considera como una base de datos NoSQL al almacenar sus datos como documentos JSON.

A bajo nivel, un servidor ElasticSearch es un sistema de gestión de varias bases de datos Apache Lucene a las que se añade replicabilidad, búsquedas complejas y orquestación.

ElasticSearch [21] es un servicio API REST, y por tanto no está diseñado para ser accedido directamente por un usuario. La interfaz gráfica para interaccionar con ElasticSearch es Kibana, que presenta los datos de forma óptima para el análisis y permite la realización de gráficas.

3.4.3 Kibana

Kibana [17] se trata de la interfaz gráfica utilizada para la interacción con los datos tratados con ElasticSearch. Presenta los datos de manera óptima, de manera que facilita la interpretación de los mismos, así como la realización de gráficas a partir de estos.

Tal y como se ha mencionado previamente en este documento, la unidad principal en una base de datos como Elastic es un documento que a su vez se encuentra asociado a un objeto JSON que representa un *item* dentro de un conjunto, como por ejemplo, una entrada en un log.

Cada documento debe tener al menos los siguientes campos de metadatos para su correcto procesado.

- _id: identificador del documento
- _index: índice al que pertenece
- _type: tipo de documento

Los índices de ElasticSearch son conjuntos lógicos en los que se organizan documentos que comparten características comunes. Son el equivalente a una base de datos en SQL y es la unidad sobre la que se ejecutan las búsquedas. Elastic permite hacer búsquedas simultaneas en varios índices mediante la utilización de *index patterns*.

3.4.3.1 Discover

La visualización de eventos se realiza a partir de **Analytics > Discover** tal y como se puede observar en la siguiente figura:

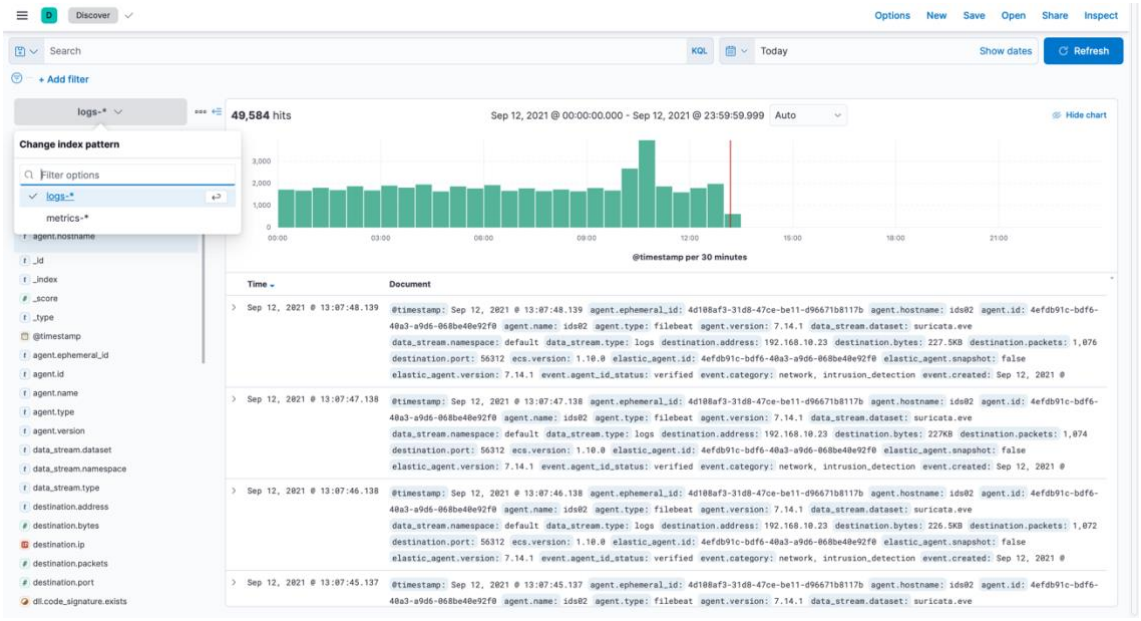

Figura 4. Visualización de eventos en Discover

Este es el panel principal de búsqueda en Kibana. Las funcionalidades a destacar son:

- **Search:** Aquí se introduce la query para obtener los resultados. Puede alternarse entre KBL y Lucene como sintaxis de búsqueda
- Ventana temporal: El filtro de tiempo para el que se muestran los datos. Aplica al campo que se haya definido en el index pattern como valor por defecto (normalmente @timestamp). Solo se mostraran los datos que esten dentro de la ventana temporal seleccionada.
- **Index Pattern:** Aquí se selecciona sobre qué datos se desea realizar las búsquedas.
- **Selected Fields**: Estos son los campos destacados que se muestran en la tabla principal.
- **Available Fields:** El resto de campos presentes en los documentos seleccionados.
- **Date histogram**: El resto de campos presentes en los documentos seleccionados.
- **Results:** El resto de campos presentes en los documentos seleccionados.
- **Filter:** El resto de campos presentes en los documentos seleccionados.
- **Management:** El resto de campos presentes en los documentos seleccionados.

3.4.3.2 Dashboards

La visualización de los eventos recibidos se obtiene en **Analytics > Dashboards**. Es posible realizar visualizaciones a partir de uno o múltiples index patterns. Estos paneles de visualización pueden incorporar elementos como tablas, mapas, gráficos entre otros.

Existen dashboards predeterminados, como el que se puede observar en la siguiente figura, no obstante, es posible crear dashboards personalizados con queries definidas por el usuario.

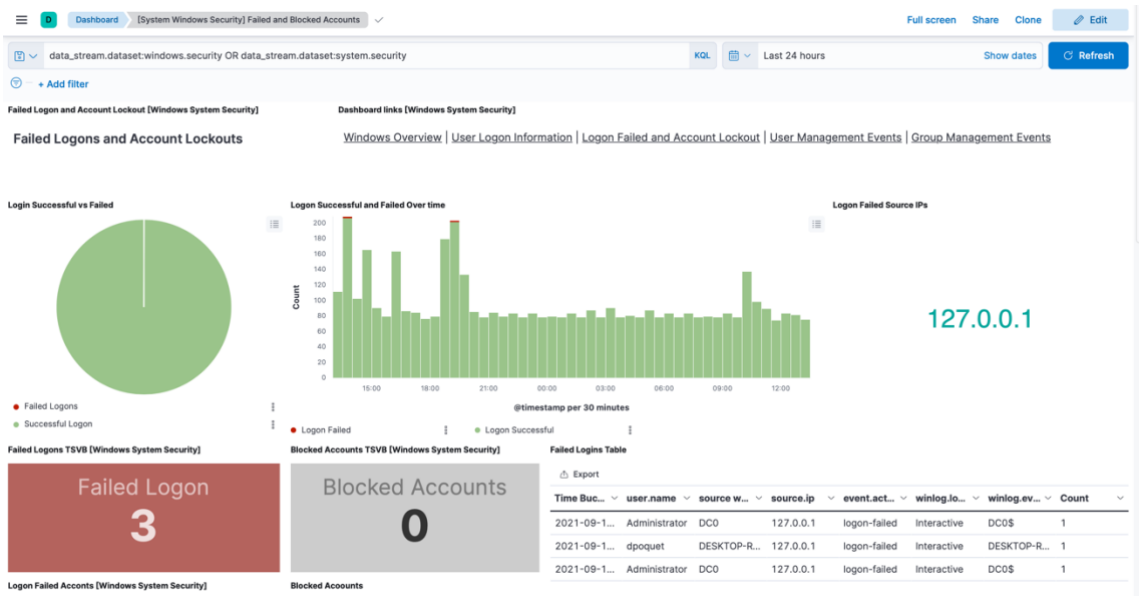

Figura 5. Dashboard "Failed Logon and Acccount Lockout"

3.4.3.3 Rules

Kibana permite la creación de reglas basadas en condiciones específicas para la activación de la alerta asociada correspondiente. Estas condiciones pueden ser tan complejas como se requieran para satisfacer los requisitos definidos [22]. Con la aplicación de **Security** tenemos la posibilidad de definir estas reglas para su posterior detección y responder a las amenazas si fuera necesario.

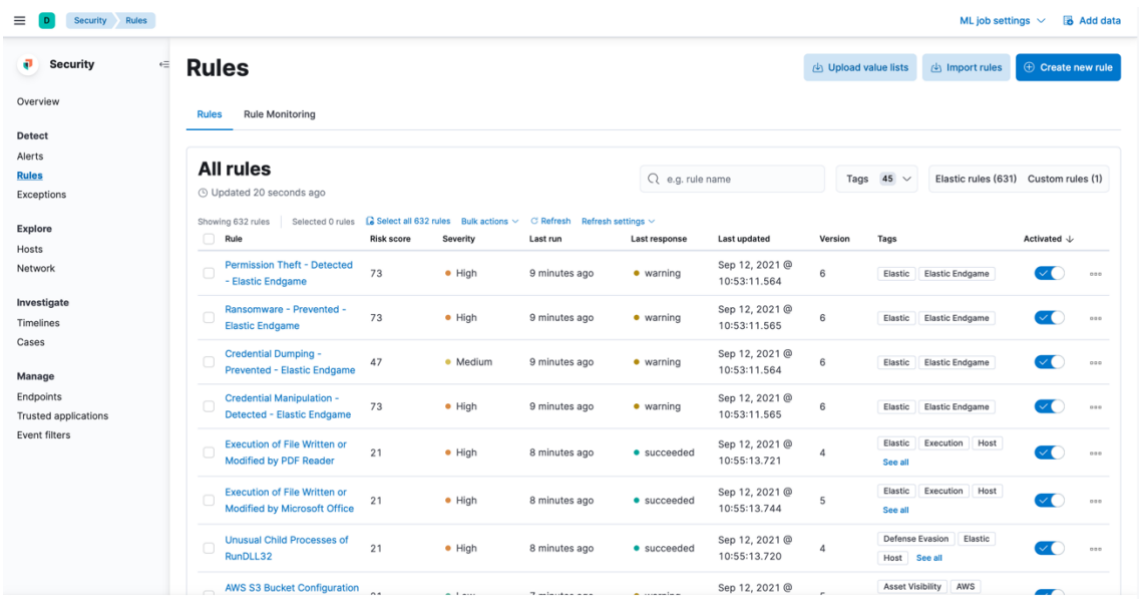

Figura 6. Reglas definidas por defecto en aplicación Seguridad de Kibana

Existen reglas predeterminadas, no obstante, es posible la definición de reglas nuevas bien importando reglas o creando una nueva con una *query* personalizada.

A continuación, se detallan las características de la regla **Ransomware - Detected - Elastic Endgame.** Esta regla se trata de una regla predefinida que se basa en los datos obtenidos por la herramienta de detección y respuesta en puntos finales (EDR) de Kibana.

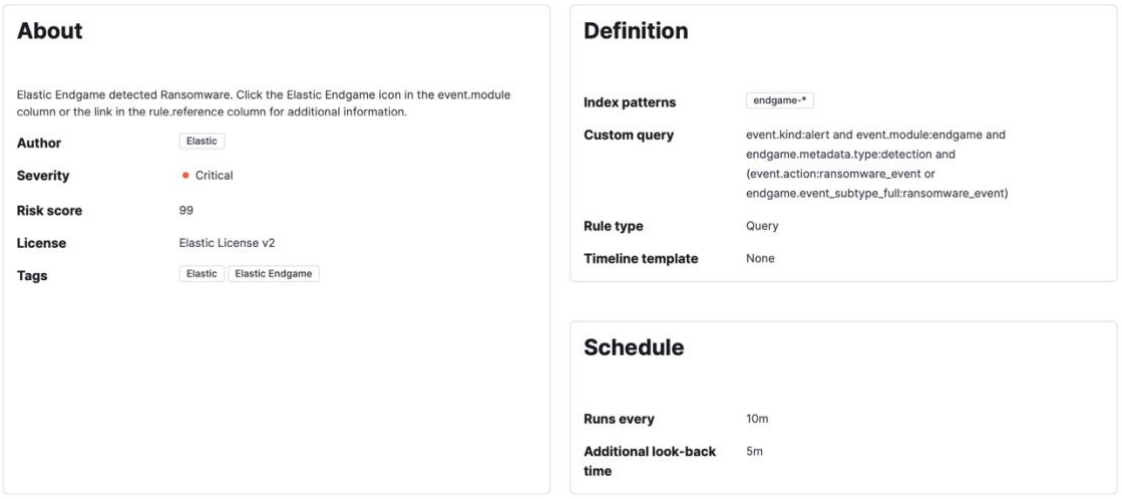

Figura 7. Características de regla "Ransomware - Detected - Elastic Endgame"

Cada alerta tiene los siguientes elementos relevantes:

- **Author:** autor de la regla, en este caso, Elastic al tratarse de una alerta predefinida.
- **Severity:** el grado de gravedad de la alerta cuyos valores pueden ser *Low*, *Medium*, *High* y *Critical*.
- **Risk score:** puntación que va en relación al grado de gravedad definido para la alerta.
- **Index patterns:** indica el *index pattern* a partir del cual se analizan los datos obtenidos, en este caso, dado que se tratan de los datos procesados por el EDR se utiliza el *index pattern endgame-**
- **Custom query:** *query* empleada para la detección de ejecución de *ransomware* en un host
- **Runs every:** las reglas se ejecutan cada cierto tiempo para la generación de alertas en caso de que se cumpla la condición especificada en **Custom query**.

3.4.3.4 Alerts

Una vez tiene lugar la activación de una alerta basada en una regla definida y habilitada, estas pueden ser visualizadas en la pestaña de **Alerts**, dentro de la aplicación de **Security** de Kibana.

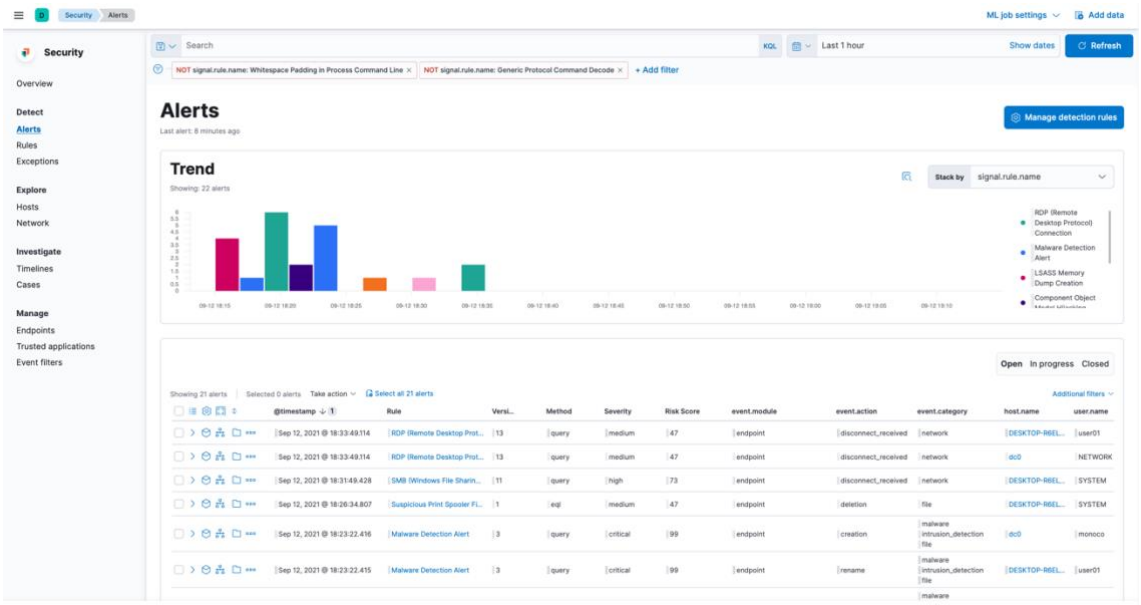

Figura 8. Visualización de alertas en pestaña Alerts de aplicación Security en Kibana

Cuando tiene lugar la ejecución de software potencialmente malicioso o malware, se observa una línea de tiempos de la creación de procesos que han tenido lugar para la detonación de dicha alerta.

En el caso de la detonación de la alerta ***Malware Prevention Alert***, se observan los procesos creados hasta la ejecución de `mimikatz.exe` tal y como se puede observar en la siguiente figura.

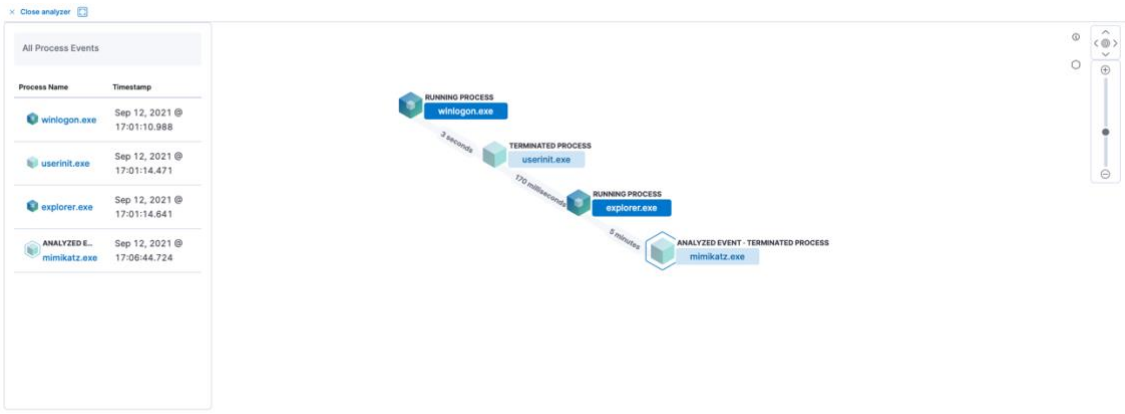

Figura 9. Creación de procesos para alerta "Malware Prevention Alert"

Adicionalmente, se permite su añadido a una línea de tiempo para su posterior análisis.

3.4.4 Fleet

Fleet proporciona una interfaz gráfica a la que se accede a través de la sección de **Management** de Kibana que permite añadir y gestionar integraciones para servicios o plataformas. Esto es posible mediante un servidor Fleet que permite conectar los agentes de Elastic a Fleet consiguiendo así una centralización de los agentes. Se destaca, además, que permite la centralización de datos provenientes de agentes con distintos formatos y fuentes. [23]

Las integraciones permiten la utilización de política a utilizar para la configuración de fuentes de entrada para registros y métricas como, por ejemplo, la ruta al fichero de eventos Sysmon con la

integración **Windows**. Estas integraciones facilitan el despliegue de actualización de agentes, así como de políticas de manera rápida y sencilla.

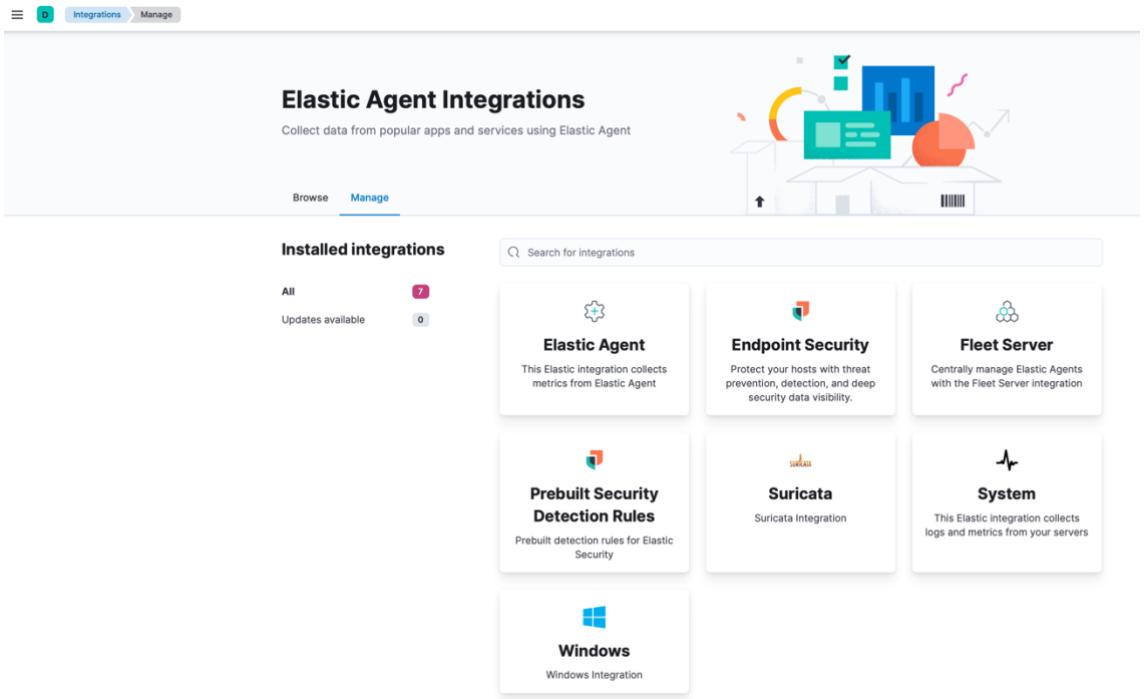

Figura 10. Administración de integraciones de Elastic Agent

A continuación, se describen brevemente las funciones desempeñadas por las integraciones relevaciones para el desarrollo de este proyecto:

- **Endpoint Security:** configura plantillas predeterminadas e index patterns necesarios para la monitorización de seguridad en los puntos finales.
- **Fleet Server:** componente del stack de Elastic que permite la centralización de la gestión de agentes de Elastic. Se despliega en un host que actuará como servidor Fleet.
- **Suricata:** lee a partir del fichero de salida JSON EVE. Este escribe alertas, anomalías, metadatos e información de ficheros recogidos a partir del tráfico analizado por Suricata.
- **System:** permite la monitorización de servidores con información referente a la CPU, memoria, procesos entre otros.
- **Windows:** permite la monitorización de sistemas operativos Windows, servicios, aplicaciones, etc. A partir de esta integración, se permite la recogida de distintos ficheros de registros de eventos de Windows tales como los eventos Sysmon o PowerShell.

Dado que se obtiene una visibilidad completa de los agentes que se comunican con el servidor Fleet, es posible obtener el estado de cada uno de los agentes, así como aquellos que tienen errores, último registro de actividad entre otros como se observa en la siguiente figura.

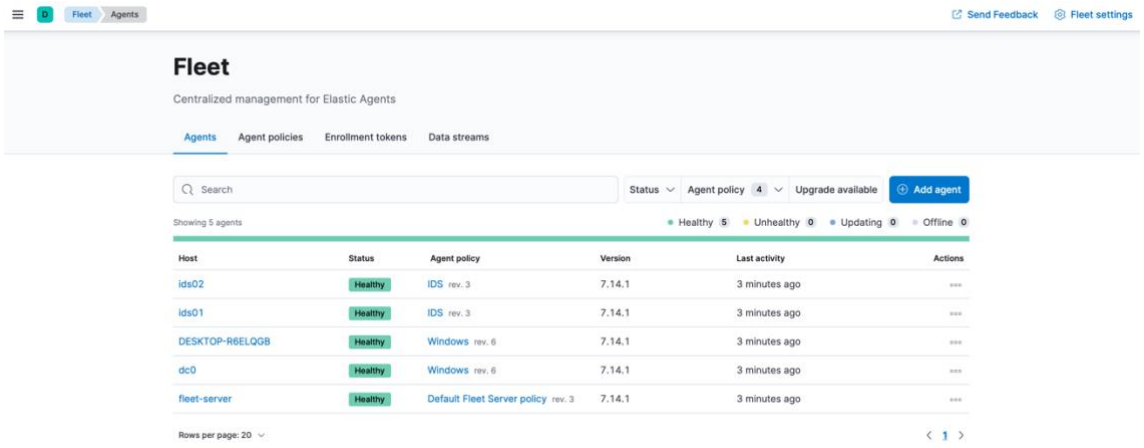

Figura 11. Listado de agentes de Elastic en Fleet

En la pestaña **Agent Policies** es posible observar las distintas políticas aplicadas a cada agente, donde cada agente consta de las siguientes integraciones previamente definidas en este documento:

Agentes	Integraciones
dc0 DESKTOP-R6ELQGB	Endpoint Security System Windows
ids01 ids02	• Suricata • System
fleet_server-1	Fleet Server

Tabla 1. Integraciones aplicadas a los agentes de Elastic

3.4.5 Endpoint Security

Esta integración constituye la herramienta de detección y respuesta de puntos finales EDR. Proporciona prevención ante amenazas de *malware* y actores maliciosos a la vez que permite el bloqueo de técnicas de ataques en el punto final.

Entre la información recopilada por el EDR se encuentran las ejecuciones de procesos, las búsquedas de dominios, los detalles de las conexiones TCP, el acceso a archivos, la comunicación de red, etc. Esta recogida se realiza de forma segura y permite la correcta contextualización de la telemetría de eventos de los puntos finales.

Lo explicado previamente es posible gracias a la integración de múltiples tecnologías que se aúnan en un único agente ligero resistente a manipulaciones que trabajan de manera independiente para cubrir funciones de prevención, detección y respuesta. [24]

3.4.6 Sysmon

System Monitor (Sysmon) es un servicio del sistema de Windows y *driver* de dispositivo que persiste a reinicios y permite la monitorización de la actividad de un sistema para su posterior

recogida en el registro de eventos de Windows Microsoft-Windows-Sysmon/Operational.evtx. Proporciona información sobre la creación de procesos, conexiones de red y operaciones referentes a la creación, modificación o borrado de ficheros. [25]

Entre las capacidades de monitorización de Sysmon se encuentran las siguientes:

- Registro de creación de procesos con inclusión de la línea de comando empleada
- Registro de hashes de imágenes creadoras de procesos
- Registro de carga de DLLs con sus firmas y hashes correspondientes.
- Registro de conexiones de red, incluyendo información sobre las direcciones IP y puertos de origen y destino, así como el proceso que ha iniciado la conexión.

Entre otros.

Para su correcta instalación se ha utilizado el fichero de configuración proporcionado por SwiftOnSecurity. [26]

3.4.7 Suricata

Suricata es un motor de monitorización de seguridad de red, IDS e IPS de alto rendimiento. El funcionamiento de la detección de anomalías en la red se basa en la aplicación de reglas preconfiguradas o manuales creadas por el propio usuario. [19]

Se ha elegido esta herramienta debido a la cantidad de reglas ya predefinidas en repositorios que se actualizan a diario. Al tratarse de una herramienta de código abierto, cuando se revela la existencia de vulnerabilidades o se publica un *exploit* de una prueba de concepto, investigadores en el campo de la ciberseguridad puede publicar una regla creada que se encuentra disponible para todos los usuarios de la comunidad. [27]

Adicionalmente, se trata de una herramienta cuya integración y automatización resulta muy sencilla.

Tal y como se ha expuesto previamente en este documento en las secciones 3.3.1 y 3.4.4, la política de integración de Suricata se aplica a los agentes IDS01, IDS02 y IDS03 para la monitorización del tráfico en distintos puntos de la infraestructura.

3.5 Implementación del centro de operaciones de seguridad (SOC)

Una vez definidas las herramientas de seguridad necesarias para la creación de un centro de operaciones descritas en el apartado Herramientas utilizadas y siguiendo las especificaciones técnicas descritas en el apartado Especificaciones técnicas de las herramientas se procede a detallar el proceso de configuración de las herramientas empleadas.

3.5.1 Docker

Siguiendo la documentación oficial de Docker [28], para su instalación se ejecutarán los siguientes comandos en la máquina donde se desplegará los diferentes componentes del paquete Elastic:

- Instalación de herramientas necesarias para el correcto funcionamiento de la herramienta

```
sudo apt-get update
    sudo apt-get install \
        apt-transport-https \
        ca-certificates \
        curl \
        gnupg \
        lsb-release
```

- Añadir la clave GPG de Docker para comprobar la integridad del software descargado a través de su firma:

```
    curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-
keyring.gpg
```

- Añadir el repositorio de paquetes de Docker al gestor de paquetes de Ubuntu:

```
    echo \
        "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-
archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
        $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null
```

- Instalar Docker:

```
    sudo apt-get update
        sudo apt-get install docker-ce docker-ce-cli containerd.io
```

3.5.2 Implementación Elastic

- Preparación del hardware

El almacenamiento y gestión de la información requiere hardware que nos permita ser ágiles al consultar la información y que tenga la capacidad de procesar los datos en pseudo tiempo real para evitar la pérdida de información. El software soporta las arquitecturas x86_64(amd64) y arm64. Se ha elegido la arquitectura x86_64(amd64) por su mayor disponibilidad en el mercado y su asequibilidad.

El dimensionamiento de recursos para el entorno utilizado es el siguiente:

- CPU: Procesador x86_64 de 4 núcleos a 3.5GHz.
- Memoria RAM: 16GB.
- Almacenamiento: 128GB de almacenamiento rápido SSD.
- Red: tarjeta de red Ethernet de un puerto Gigabit.

- Preparación del sistema operativo

Se ha elegido el sistema operativo de código abierto Ubuntu 20.04(LTS) basado en Linux 5.4 para el despliegue de los distintos componentes de la solución de Elastic. Con la finalidad de gestionar el software utilizado, el entorno estará basado en contenedores Docker y se utilizará el sistema de orquestación Docker Compose.

En el apartado Docker se detalla el proceso de instalación del software necesario y las configuraciones específicas para Elasticsearch basadas en las recomendaciones de Elastic [29] para el correcto aprovechamiento de los recursos del sistema.

Para el correcto funcionamiento de Elasticsearch será necesario la aplicación previa a su despliegue de los siguientes comandos:

- Crear el fichero 20-elasticsearch.conf en el directorio /etc/sysctl.d con el siguiente contenido:

```
vm.swappiness=1
    vm.max_map_count=262144
    net.ipv4.tcp_retries2=5
```

- Ejecutar el comando para aplicar los cambios.

```
sysctl -p
```

- Despliegue de Elastic

Con el objetivo de demostrar las capacidades de la solución se va a desplegar Elasticsearch en un único nodo sobre Docker, siguiendo las especificaciones de Elasticsearch [30], Kibana [31]y Fleet [32]. En este despliegue se cumplen las medidas mínimas de seguridad para Elastic [33], no siendo suficientes para un entorno productivo en una organización considerados en la guía de securización [34].

Para el despliegue del paquete Elastic será necesario el siguiente fichero `docker-compose.yml` que recoge las diferentes configuraciones propias para un entorno seguro. Será necesario generar claves aleatorias para el cifrado de los datos e introducirlas en la configuración:

```
version: "3.9"
services:

  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:7.14.1
    hostname: elasticsearch
    environment:
      - discovery.type=single-node
      - cluster.name=cluster0
      - node.name=es0
      - bootstrap.memory_lock=true
      - xpack.security.enabled=true
      - xpack.security.authc.api_key.enabled=true
    networks:
      - elastic
    ports:
      - "9200:9200"
    restart: unless-stopped
    ulimits:
      memlock:
        soft: -1
        hard: -1
    volumes:
      - es01-data:/usr/share/elasticsearch/

  kibana:
    image: docker.elastic.co/kibana/kibana:7.14.1
    hostname: kibana
    environment:
      - SERVER_NAME=kibana.dominio.tld
      - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
      - ELASTICSEARCH_USERNAME=kibana_system
      - ELASTICSEARCH_PASSWORD=
      - XPACK_SECURITY_ENCRYPTIONKEY=
```

```
      - XPACK_ENCRYPTEDSAVEDOBJECTS_ENCRYPTIONKEY=
      - XPACK.REPORTING.ENCRYPTIONKEY=
      - XPACK_SECURITY_SESSION.IDLETIMEOUT=1h
      - XPACK_SECURITY_SESSION.LIFESPAN=30d
    networks:
      - elastic
    ports:
      - "5601:5601"
    restart: unless-stopped


  fleet-server:
    image: docker.elastic.co/beats/elastic-agent:7.14.1
    hostname: fleet-server
    environment:
      - FLEET_SERVER_ENABLE=true
      - FLEET_SERVER_ELASTICSEARCH_HOST=http://elasticsearch:9200
      - FLEET_SERVER_SERVICE_TOKEN=
    networks:
      - elastic
    ports:
      - "8220:8220"
    restart: unless-stopped

networks:
  elastic:

volumes:
  es01-data:
```

- Despliegue de los agentes Elastic Agent

Previamente a instalar los Elastic Agent, se deben configurar las políticas que van a ser aplicadas a los mismos para que se configuren automáticamente con las funciones deseadas. La interfaz de Fleet nos permite crear las distintas políticas y, posteriormente, añadir las integraciones a cada política. Para crear una nueva política habrá que pulsar el botón “Create agent policy”.

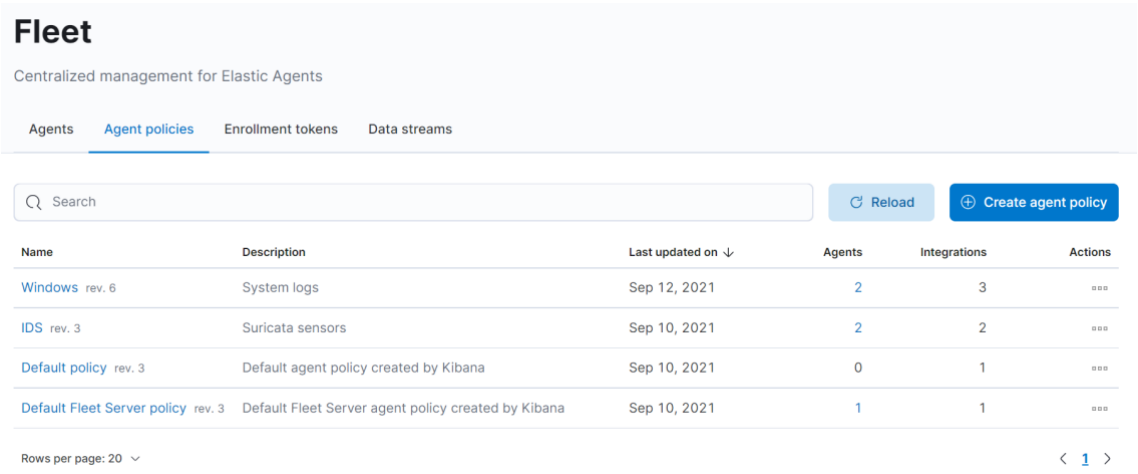

Figura 12. Gestión de políticas de Fleet

Las políticas que crear se basan en el rol de los sistemas en los que se instala el Elastic Agent:

1. Windows: recogida de logs del sistema operativo, logs de seguridad de Windows e instalación del módulo Endpoint Security. Para añadir estar integraciones habrá que pulsar en el botón "Add integration". Esta política se utilizará para los equipos de usuario y los servidores Windows.

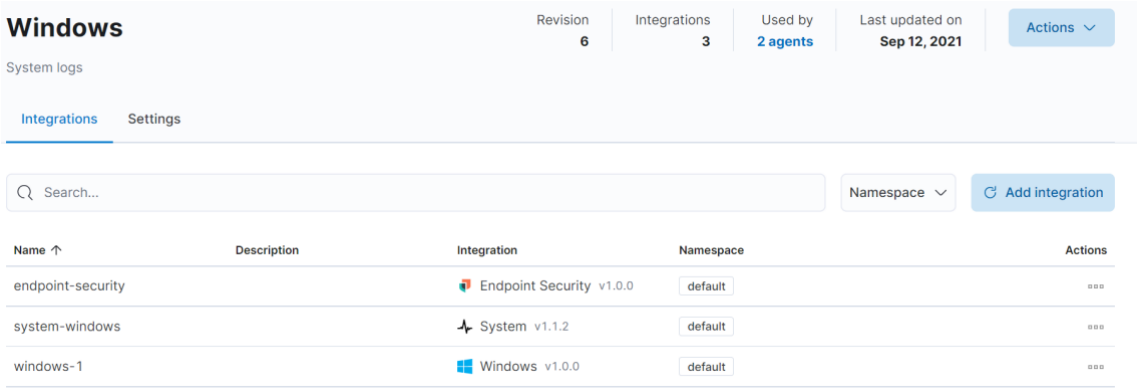

Figura 13. Política para Windows en la interfaz Fleet

2. IDS: recogida de los logs de Suricata y los logs del sistema operativo. Para añadir estar integraciones habrá que pulsar en el botón "Add integration". Esta política se utilizará para los IDS desplegados en la red.

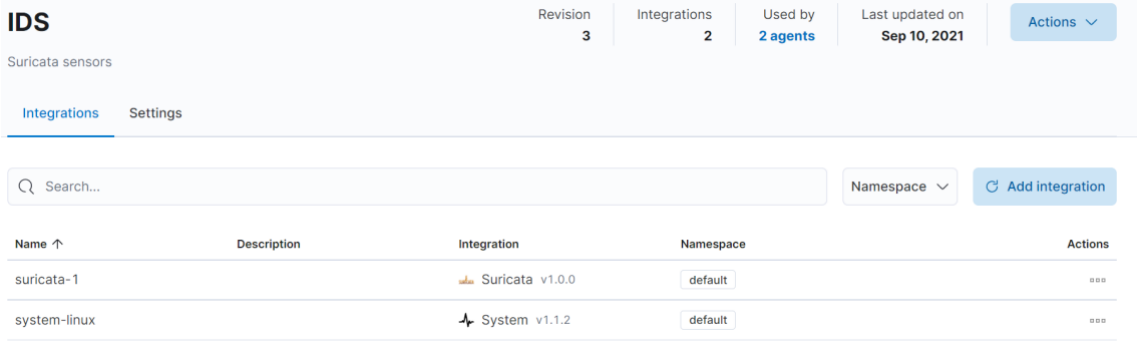

Figura 14. Política para IDS en la interfaz Fleet

Una vez creadas las políticas, se procederá con la instalación de los agentes en las diferentes máquinas de la red; se deberá realizar mediante herramientas de despliegue y ejecución remota como SCCM o políticas de grupo.

- Instalación de Elastic Agent en Linux x86_64
  - Descargar y descomprimir el software necesario:

```
curl -L -O
https://artifacts.elastic.co/downloads/beats/elastic-
agent/elastic-agent-7.14.1-linux-x86_64.tar.gz
tar -xzvf elastic-agent-7.14.1-linux-x86_64.tar.gz
```

  - Instalar Elastic Agent gestionado por Fleet:

```
cd elastic-agent-7.14.1-linux-x86_64
```

```
sudo ./elastic-agent install -f --
url=<url_servidor_fleet> --enrollment-
token=<token_fleet_server>
```

- Instalación de Elastic Agent en Linux arm64
    - Descargar y descomprimir el software necesario:

```
curl -L -O
https://artifacts.elastic.co/downloads/beats/elastic-
agent/elastic-agent-7.14.1-linux-arm64.tar.gz
ar -xzvf elastic-agent-7.14.1-linux-arm64.tar.gz
```

    - Instalar Elastic Agent gestionado por Fleet:

```
cd elastic-agent-7.14.1-linux-arm64
sudo ./elastic-agent install -f --
url=<url_servidor_fleet> --enrollment-
token=<token_fleet>
```

- Instalación de Elastic Agent en Windows
    - Descargar y descomprimir el software necesario:

```
Invoke-WebRequest
"https://artifacts.elastic.co/downloads/beats/elastic-
agent/elastic-agent-7.14.1-windows-x86_64.zip" -OutFile
"C:\Windows\Temp\elastic-agent-7.14.1-windows-x86_64.zip"
Expand-Archive C:\Windows\Temp\elastic-agent-7.14.1-
windows-x86_64.zip -DestinationPath
C:\Windows\Temp\elastic-agent-7.14.1-windows-x86_64
```

    - Instalar Elastic Agent gestionado por Fleet:

```
cd C:\Windows\Temp\elastic-agent-7.14.1-windows-x86_64
.\elastic-agent.exe install -f --url=<url_servidor_fleet> -
-enrollment-token=<token_fleet> (Ejecutar como
Administrador)
```

- Obtención del token de registro de Elastic Agent desde Fleet:
    - Registrar un nuevo agente desde Kibana en el apartado Fleet.

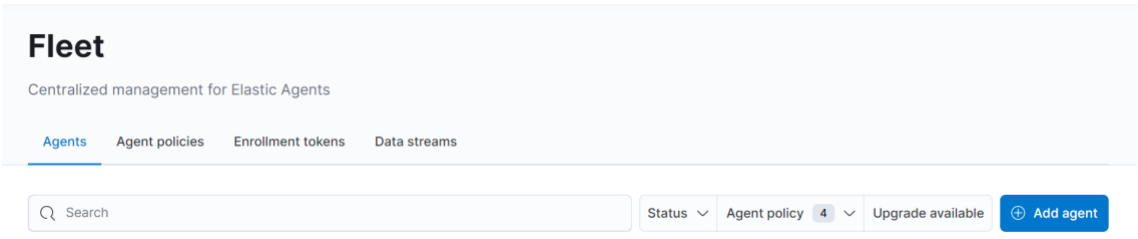

Figura 15. Interfaz de Fleet

- Seleccionar la política a desplegar en el nuevo Elastic Agent.

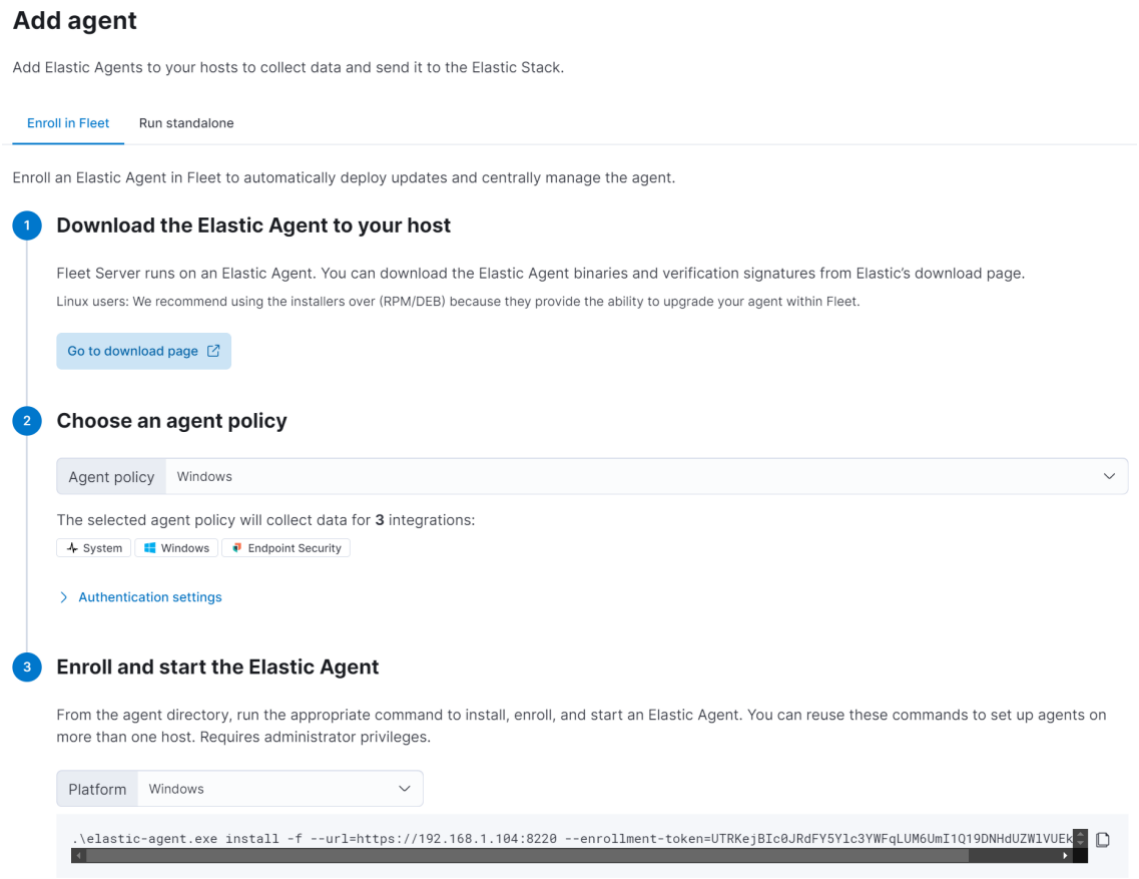

Figura 16. Interfaz para añadir nuevo agente a Fleet

- El comando generado por Fleet para la instalación del agente contiene la URL del servidor Fleet y el token de registro para el nuevo Elastic Agent.

Una vez instalado el Elastic Agent, este se sincronizará automáticamente con el servidor de Fleet para recibir la política y las integraciones asociadas, configurándose automáticamente e iniciando el envío de logs hacia Elasticsearch.

3.5.3 Implementación de Suricata

Las sondas IDS utilizadas durante el diseño de la red en el apartado Diseño de la segmentación de una red industrial llevan el software de detección de amenazas de red Suricata. Esta guía deberá repetirse cada vez que se desee implementar un nuevo IDS.

- Preparación de la red

En un entorno IT/OT como el descrito anteriormente, se pretende ganar la máxima visibilidad de la red mediante la implementación de sistemas IDS en puntos estratégicos. Estos IDS recibirán una copia del tráfico de red, por lo que será necesario configurar la electrónica de red para copiar el tráfico conmutado hacia el puerto de red en el que Suricata va a recibir los datos.

- Instalación de Suricata

La instalación del software Suricata IDS se ha realizado siguiendo las instrucciones de los desarrolladores [35]

- Instalación de herramientas necesarias:

```
sudo apt-get install software-properties-common
```

- Añadir el repositorio de software en el gestor de paquetes de Ubuntu:

```
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update
```

- Instalación de Suricata IDS:

```
sudo apt-get install suricata
```

- Configuración de Suricata IDS:

- Modificar en el fichero /etc/suricara/suricata.yaml con el fichero de configuración del 0 Anexo I – Archivo de configuración suricata.yaml

El archivo de configuración del Anexo I necesita la modificación de las siguientes líneas para el correcto funcionamiento dentro de la red implementada por el usuario.

```
            HOME_NET:
"[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
            EXTERNAL_NET: "!$HOME_NET"

            af-packet:
                - interface: eth0
```

- Descarga de las reglas de detección:

```
suricata-update update
```

- Instalación de Elastic Agent

Para el envío de los logs generados por Suricata IDS se va a utilizar el software Elastic Agent [36], que recoge estos ficheros mediante Filebeat y los procesa para normalizar los datos antes de enviarlos a Elasticsearch.

La instalación de Elastic Agent en Linux está detallada en el apartado Implementación Elastic siguiendo la documentación proporcionada por Elastic [37].

Capítulo 4. Análisis de los resultados

4.1 Introducción

En este capítulo se pretende evaluar las medidas de seguridad indicadas en el capítulo 3 Capítulo Diseño e implementación del centro de operaciones de seguridad (SOC) para ello se han escogido vulnerabilidades recientes de entornos Windows como PrintNightmare, actualmente este tipo de ataques se encuentran durante los ejercicios de respuesta de incidentes y análisis forense.

De igual manera se detonará un *ramsonware* con la finalidad de determinar si nuestro sistema sería capaz de monitorizar los principales puntos durante las investigaciones de respuesta de incidentes.

1. Vector de entrada y primera máquina comprometida
2. Acceso a credenciales con permisos privilegiados
3. Posibles movimientos laterales
4. Despliegue del *ransomware* entre las diferentes máquinas de la red
5. Detonación del *ramsonware*.

Para demostrar que una correcta segmentación de la red podría evitar que la detonación del *ramsonware* afectase a todos los equipos de esta.

4.2 Simulación del ataque

Antes de comenzar con la descripción del ataque se ha de mencionar que para poder realizar la parte de la simulación del incidente se ha tenido que modificar el EDR Endopint security del modo de prevención a detección ya que si no el propio EDR bloqueaba e impedía el desarrollo del ejercicio.

El vector de entrada del ataque llevado a cabo para el ámbito práctico de este documento se presupone que se trata de un *phishing*. Esto se considera que se encuentra fuera del marco de este documento.

Una vez el atacante obtiene las credenciales del usuario de una red corporativa user01 mediante técnicas de ingeniería social, se utilizan para la obtención de acceso a la red.

Se parte de la situación en la que las credenciales del usuario obtenidas no poseen privilegios administrativos en la máquina dentro de la red. Es por ello que se busca aprovechar una vulnerabilidad que permita a un atacante elevar privilegios. Se obtienen privilegios administrativos en la máquina local con la vulnerabilidad de Windows PrintNightmare. PrintNightmare es una vulnerabilidad en el servicio de impresión de Microsoft Windows (Print Spooler) que permite la ejecución de código con privilegios de usuario SYSTEM. Esta vulnerabilidad se puede utilizar para la escalada de privilegios de forma local en el equipo en que se explota. [38]

Tras la obtención de privilegios administrativos en la máquina, al poder ejecutar procesos con el usuario SYSTEM, que es el de mayor privilegio en sistemas Windows, podemos acceder al contenido de la memoria RAM. Esto se puede aprovechar para buscar las credenciales de usuarios que hayan iniciado sesión anteriormente mediante el volcado de la memoria del proceso del sistema de autenticación de Windows (LSASS).

Tras la obtención de volcado de LSASS, se emplea el software mimikatz.exe que permite leer e interpretar el formato de los archivos de volcado de memoria de Windows, consiguiendo así la lectura de las credenciales de otros usuarios del sistema [39]. En este volcado se encuentran las credenciales del administrador del dominio. Esto se puede emplear para realizar un movimiento lateral desde la estación de trabajo donde el atacante posee privilegios administrativos hacia el controlador de dominio con las credenciales del administrador de dominio.

Esto supone el compromiso del dominio entero, por lo que se puede proceder con la descarga del fichero que será el detonante del *ransomware* en la estación de trabajo y su transferencia al controlador de dominio para su posterior ejecución.

Pese a haberse controlado el dominio si la red se encuentra segmentada tal y como se detalla en el apartado *Diseño de la segmentación de una red industrial* el atacante no podría moverse lateralmente hacia fuera de la red corporativa (usuarios y servidores), donde se encuentra ubicado el ataque, ya que no se permitiría ese tipo de tráfico (*remote desktop protocol*) hacia máquinas fuera de esta red. Esta configuración protegería del ataque de *ransomware* a la red operacional manteniendo activo el servicio de producción de la compañía.

4.3 Investigación del ataque

A continuación, se detallará el proceso de análisis por parte del equipo de seguridad una vez han recibido las alertas relacionadas con el incidente. En este apartado se guiará al lector durante el proceso de investigación ya concluido puesto que en un entorno en producción se generan un gran volumen de alertas. Los investigadores de seguridad deben de ser capaces de filtrar y extraer la información relevante almacenada en el SIEM de cara a una investigación.

El primer comportamiento anómalo observado en la red corporativa corresponde a la máquina DESKTOP-R6ELQGB. En este equipo se generan una serie de alertas en Kibana correspondientes a la explotación de la vulnerabilidad PrintNightmare.

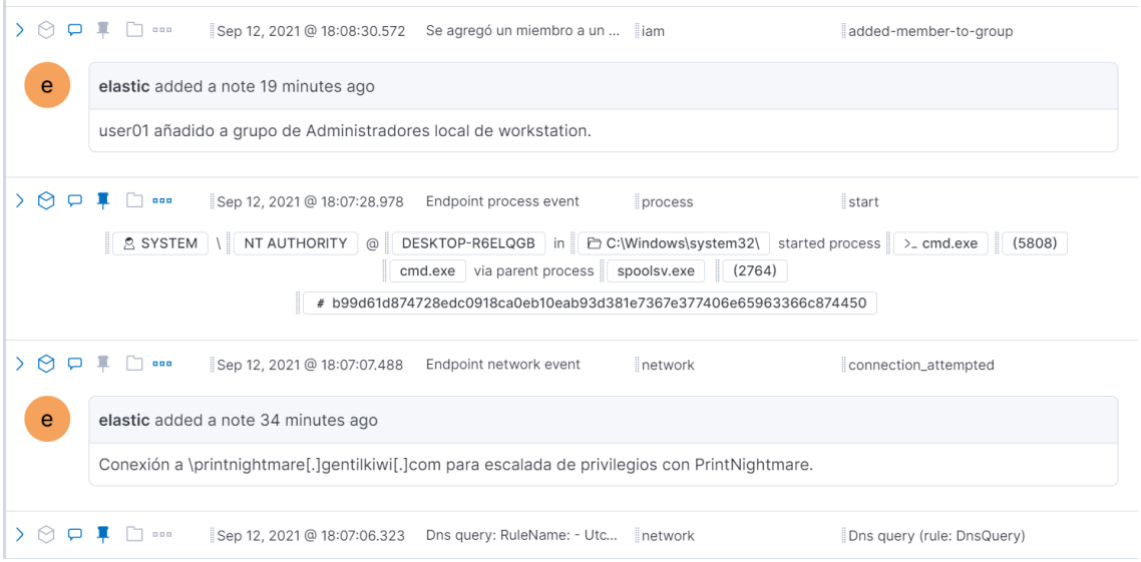

Figura 17. Explotación de PrintNightmare

En las alertas que se muestran en la Figura 17. Explotación de PrintNightmare se puede observar los diferentes pasos seguidos por el atacante para dotar al usuario user01 de privilegios de Administrador local. En primer lugar, el atacante realiza una conexión con un servidor de impresión remota \printnightmare[.]gentilkiwi[.]com previamente modificado por el usuario malicioso para explotar la vulnerabilidad de PrintNightmare.

Esta acción permite al atacante abrir una terminal de comandos (CMD) con privilegios de usuario SYSTEM, que aprovecha para añadir al user01, usuario comprometido por el atacante durante el *phishing*, al grupo de administradores locales.

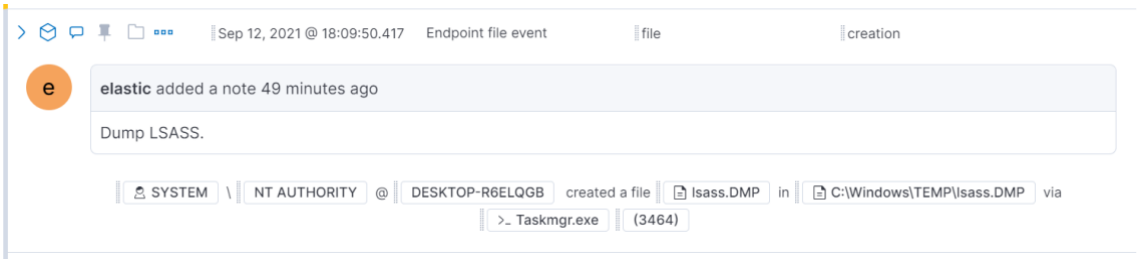

Figura 18. Volcado de LSASS

Empleando la terminal de comandos con permisos de SYSTEM obtenida durante la ejecución de PrintNightmer el atacante obtiene la información relacionada con el proceso de autenticación de usuarios de Windows de la máquina mediante el volcado de la memoria asociada al proceso lsass.exe. Esta acción se puede observar en la Figura 18. Volcado de LSASS.

Una vez se realiza el volcado de la memoria, se ejecuta el software mimikatz que permite al atacante acceder a la información en claro del contenido del fichero lsass.DMP generado en el punto anterior.

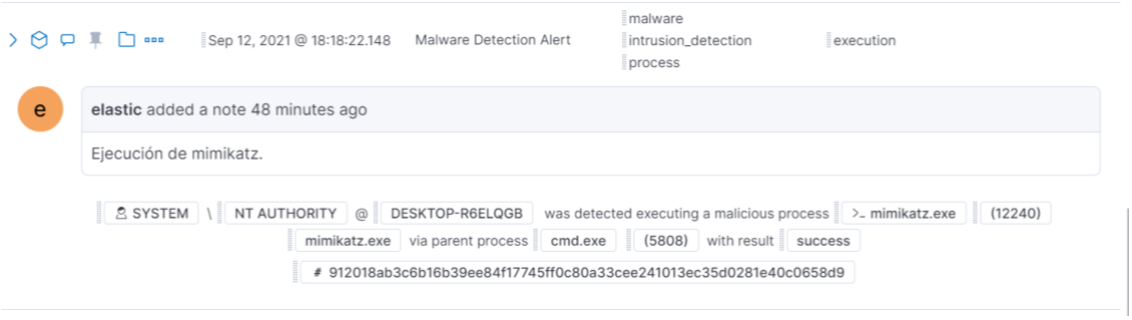

Figura 19. Ejecución de mimikatz.exe

Al acceder al contenido almacenado en el archivo lsass.DMP se obtienen las credenciales de los usuarios que habían iniciado previamente sesión en la máquina DESKTOP-R6ELQGB entre los cuales se encuentran las credenciales del usuario administrador de dominio monoco.

La obtención de estas credenciales permite al atacante el acceso al controlador de dominio dc0. Esto se puede observar en la Figura 20. Movimiento lateral por RDP hacia controlador de dominio donde el usuario monoco realiza una conexión remota desde el equipo comprometido DESKTOP-R6ELQGB hacia el controlador de dominio dc0

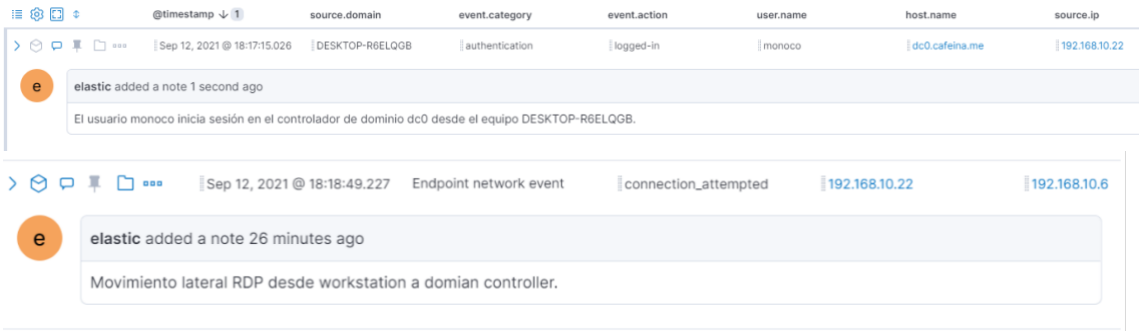

Figura 20. Movimiento lateral por RDP hacia controlador de dominio

El atacante ahora posee privilegios administrativos del dominio, lo que le permite tener control y acceso total a las diferentes máquinas que forman parte de este. Este nivel de privilegios le permitirían realizar cualquier acción sobre el dominio, en este caso se ha utilizado este nivel de permisos para la detonación de un *ransomware*.

El paso siguiente constituye la descarga del binario malicioso detonante del *ransomware* a la estación de trabajo y su transferencia al controlador de dominio tal y como se puede observar en la Figura 21. Descarga de binario malicioso en estación de trabajo y la Figura 22. Transferencia del binario malicioso al controlador de dominio.

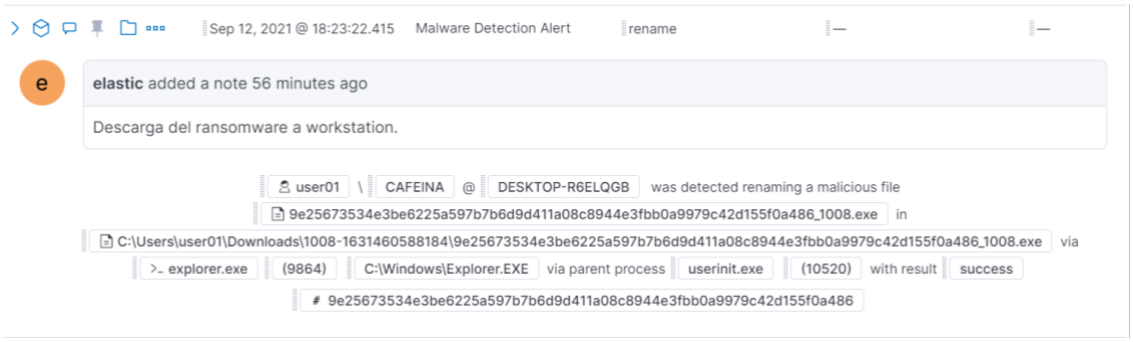

Figura 21. Descarga de binario malicioso en estación de trabajo

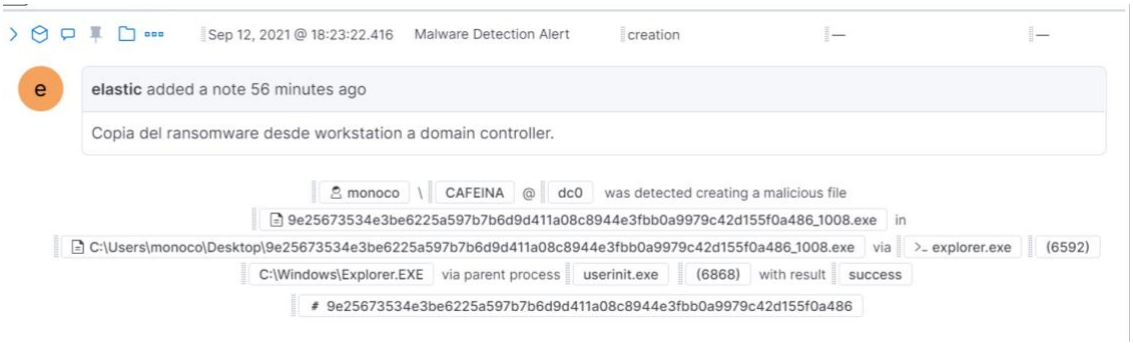

Figura 22. Transferencia del binario malicioso al controlador de dominio

Finalmente, una vez el atacante transfiere el binario malicioso al controlador de dominio dc0 procede a la ejecución del *ransomware* que comprometerá e inhabilitará completamente el dominio corporativo. La ejecución del *ramsomware* en el dc0 con el usuario administrador de dominio comprometido monoco se puede observar en la Figura 23. Ejecución del *ransomware* en el controlador de dominio

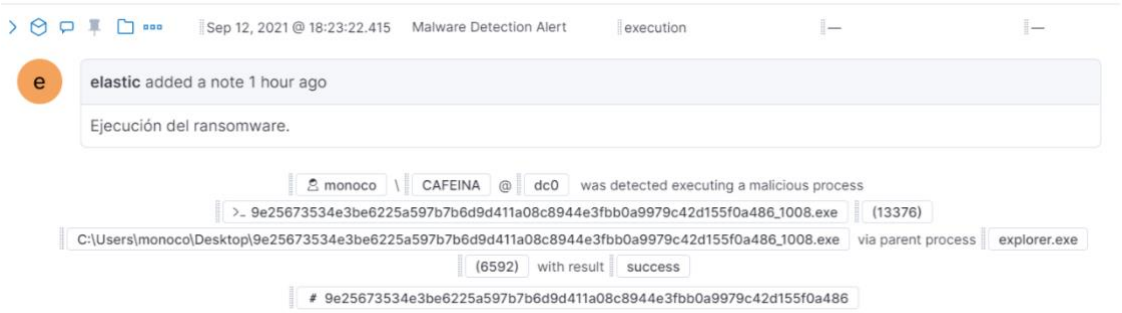

Figura 23. Ejecución del *ransomware* en el controlador de dominio

Capítulo 5. Conclusiones y futuras vías de trabajo

5.1 Introducción

A lo largo de este TFG se ha intentado contemplar la securización de una red industrial sin ningún tipo de monitorización ni de diseño de red seguro aplicado previamente.

En este capítulo se identificarán cuales son las mejoras obtenidas tras la aplicación de todas las indicaciones detalladas durante el proyecto frente al escenario previo sin ningún tipo de medidas de seguridad.

Como es sabido el mundo de la ciberseguridad es extenso y se encuentra en constante cambio es por eso por lo que se dejarán futuras vías de trabajo para ampliar de manera personal si se desea continuar investigando sobre la mejora de las medidas aplicadas durante este proyecto.

5.2 Conclusiones

Como se ha podido observar durante Capítulo 4 Análisis de los resultados la introducción de un sistema de monitorización completo proporciona una visión general de la red y de los equipos. Este control global de las comunicaciones y acciones realizadas por las máquinas de la organización ha permitido la detección e investigación de manera ágil y sencilla a los investigadores de seguridad de cara a un incidente.

Se puede concluir que las mejoras aportadas tras la implementación de este TFG en redes industriales son:

- Diseño de una red industrial segura basándose en el estándar IEC 62443 sobre como segmentar y donde introducir los elementos de seguridad para una correcta aplicación de las diferentes subredes y las comunicaciones entre ellas.
- Visión global de las diferentes subredes, elementos de seguridad y equipos de la red industrial.
- Monitorización y detección de anomalías de seguridad en tiempo real de los diferentes equipos de punto final.
- Bloqueo de actividades maliciosas en los equipos de punto final debido a la introducción de un cliente de detección y respuesta (EDR)
- Posibilidad de introducción de indicadores de compromiso (IOC) propios en los EDR e IDS proporcionando seguridad adicional ante ataques actuales.
- Detección de anomalías en la red en tiempo real mediante la aplicación de reglas Suricata.
- Agregación de todos los logs generados por los diferentes elementos de la red en un SIEM.
- Visualización de los datos mediante paneles de información que proporcionan la información relevante de cara a la monitorización y detección de anomalías
- Introducción de reglas de detección de anomalías basadas en el análisis de los logs recibidos en el SIEM.
- Generación de alertas de seguridad para los analistas informáticos tras la detección de anomalías por parte de las reglas de detección.

5.3 Futuras vías de trabajo

El mundo de la ciberseguridad se encuentra en constante cambio debido a la cantidad de nuevas amenazas y publicaciones de información constantes que afectan sobre los diferentes elementos de las empresas.

La actualización de las herramientas de seguridad y la implementación de nuevas alertas y reglas de monitorización es un trabajo diario del equipo de seguridad. Para la gestión de este tipo de acciones se sugieren las siguientes vías de ampliación de este TFG:

- Automatización de la recolección e introducción de los diferentes indicadores de compromiso de las diferentes fuentes públicas (CSIRT[1], CERT[2] y fabricantes de seguridad informática)
- Desarrollo de nuevas reglas de detección de anomalías específicas para las tecnologías aplicadas dentro de la red industrial.
- Desarrollo de paneles específicos para las diferentes herramientas de monitorización propias dentro de la organización.
- Generación de documentación y planes de acción a seguir por los diferentes miembros de la empresa ante amenazas específicas.
- Generación de planes de continuidad de negocio en caso de disrupción de la actividad de la empresa.

[1] Equipo de Respuesta a Incidentes de Seguridad. Es una organización habitualmente provincial responsable de recibir, revisar, responder y publicar información sobre sobre incidentes de seguridad.
[2] Equipo de respuesta de emergencia. Grupo de expertos encargado a el análisis resolución de incidentes de seguridad y publicación de indicadores de compromiso. Se suele gestionar a nivel gubernamental.

Bibliografía

[1] Cyber Seguridad S.A., «Diferencias entre SOC, CERT y CSIRT,» [En línea]. Available: https://www.cyberseg.com/single-post/2018/09/05/diferencias-entre-soc-cert-y-csirt. [Último acceso: 28 Mayo 2021].

[2] exabeam, «The Modern Security Operations Center, SecOps and SIEM: How They Work Together,» [En línea]. Available: https://www.exabeam.com/siem-guide/the-soc-secops-and-siem/. [Último acceso: 28 Mayo 2021].

[3] Global Technology, «SOC: SECURITY OPERATIONS CENTER ¿Qué hace un SOC o centro de operaciones de seguridad?,» [En línea]. Available: https://www.globalt4e.com/ciberseguridad/soc/?cv=1. [Último acceso: 31 Mayo 2021].

[4] McAfee, «What Is a Security Operations Center (SOC)?,» [En línea]. Available: https://www.mcafee.com/enterprise/en-us/security-awareness/operations/what-is-soc.html. [Último acceso: 2021 Mayo 29].

[5] J. Singh, «SIEM vs SOC: Do You Know the Difference? (Must Know Info),» [En línea]. Available: https://cybersecuritykings.com/2020/07/18/what-is-the-difference-between-siem-and-soc/#:~:text=SIEM%20stands%20for%20Security%20Incident,from%20the%20SIEM%20log%20analysis. [Último acceso: 31 Mayo 2021].

[6] Forcepoint, «What is a Firewall?,» [En línea]. Available: https://www.forcepoint.com/es/cyber-edu/firewall. [Último acceso: 2021 Junio 3].

[7] B. Lutkevich, «intrusion detection system (IDS),» [En línea]. Available: https://searchsecurity.techtarget.com/definition/intrusion-detection-system. [Último acceso: 3 Junio 2021].

[8] Fortinet, «Proxy Server,» [En línea]. Available: https://www.fortinet.com/resources/cyberglossary/proxy-server. [Último acceso: 3 Junio 2021].

[9] Secure & IT, «Ciberseguridad para empresas: prevenir, detectar y detener ataques.,» 2018 Noviembre 2018. [En línea]. Available: https://www.secureit.es/ciberseguridad-para-empresas-prevenir-detectar-y-detener-ataques/. [Último acceso: 3 Junio 2021].

[10] McAfee, «What Is Endpoint Detection and Response (EDR)?,» [En línea]. Available: https://www.mcafee.com/enterprise/es-es/security-awareness/endpoint/what-is-endpoint-detection-and-response.html#:~:text=Endpoint%20detection%20and%20response%20(EDR)%2C%20also%20known%20as%20endpoint,automated%20response%20and%20analysis%20capabilities. [Último acceso: 3 Junio 2021].

[11] IEEE, «Industrial communication networks – Network and system security – Part 1-1: Terminology, concepts and models». Patente IEC/TS 62443-1-1, Julio 2009.

[12] K. Lewotsky, «Automate,» 02 12 2021. [En línea]. Available: https://www.automate.org/industry-insights/cybersecurity-practices-for-the-digital-factory.

[13] elastic, «Elastic,» 2021. [En línea]. Available: https://www.elastic.co/. [Último acceso: 8 Junio 2021].

[14] elastic, "What is Elasticsearch," 2021. [Online]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html. [Accessed 23 Junio 2021].

[15] elastic, «Elastic Agent,» 2021. [En línea]. Available: https://www.elastic.co/es/elastic-agent. [Último acceso: 8 Junio 2021].

[16] elastic, «Endpoint security,» [En línea]. Available: https://www.elastic.co/es/endpoint-security/. [Último acceso: 8 Junio 2021].

[17] elastic, «Kibana - your windows into Elastic,» 2021. [En línea]. Available: https://www.elastic.co/guide/en/kibana/current/introduction.html. [Último acceso: 23 Junio 2021].

[18] elastic, «Elastic Security,» 2021. [En línea]. Available: https://www.elastic.co/es/security. [Último acceso: 8 Junio 2021].

[19] The Open Information Security Foundation, «Suricata,» 2021. [En línea]. Available: https://suricata.io. [Último acceso: 8 Junio 2021].

[20] Docker, «Docker overview,» [En línea]. Available: https://docs.docker.com/get-started/overview/. [Último acceso: 23 Junio 2021].

[21] elastic, «El corazón del Elastic Stack, gratuito y abierto,» 2021. [En línea]. Available: https://www.elastic.co/es/elasticsearch/. [Último acceso: 8 Junio 2021].

[22] elastic, «Alerting,» 2021. [En línea]. Available: https://www.elastic.co/guide/en/kibana/7.14/alerting-getting-started.html. [Último acceso: 25 Junio 2021].

[23] elastic, «Fleet User Guide,» 2021. [En línea]. Available: https://www.elastic.co/guide/en/fleet/current/index.html. [Último acceso: 25 Junio 2021].

[24] Sofecta, «Endpoint Detection & Response (EDR),» [En línea]. Available: https://sofecta.com/edr-endpoint-detection-response/. [Último acceso: 25 Junio 2021].

[25] Microsoft, «Sysmon v.13.24,» 18 Agosto 2021. [En línea]. Available: https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon. [Último acceso: 23 Junio 2021].

[26] SwiftOnSecurity, «https://github.com/SwiftOnSecurity/sysmon-config,» 7 Junio 2021. [En línea]. Available: sysmon-config | A Sysmon configuration file for everybody to fork. [Último acceso: 28 Junio 2021].

[27] Bricata, «What is Suricata? Intro to a Best of Breed Open Source IDS and IPS,» [En línea]. Available: https://bricata.com/blog/what-is-suricata-ids/. [Último acceso: 28 Junio 2021].

[28] Docker, «Install Docker Engine on Ubuntu,» [En línea]. Available: https://docs.docker.com/engine/install/ubuntu/. [Último acceso: 3 Agosto 2021].

[29] elastic, «Important System Configuration,» [En línea]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.14/system-config.html. [Último acceso: 3 Agosto 2021].

[30] elastic, «Install Elasticsearch with Docker,» [En línea]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.14/docker.html. [Último acceso: 3 Agosto 2021].

[31] elastic, «Install Kibana with Docker,» [En línea]. Available: https://www.elastic.co/guide/en/kibana/current/docker.html. [Último acceso: 3 Agosto 2021].

[32] elastic, «Run Elastic Agent in a container,» [En línea]. Available: https://www.elastic.co/guide/en/fleet/current/elastic-agent-container.html. [Último acceso: 3 Agosto 2021].

[33] elastic, «Set up minimal security for Elasticsearch,» [En línea]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.14/security-minimal-setup.html. [Último acceso: 4 Agosto 2021].

[34] elastic, «Set up basic security for the Elastic Stack plus secured HTTPS traffic,» [En línea]. Available: https://www.elastic.co/guide/en/elasticsearch/reference/7.14/security-basic-setup-https.html. [Último acceso: 4 Agosto 2021].

[35] Suricata, «Ubuntu Installation - Personal Package Archives (PPA),» [En línea]. Available: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Ubuntu_Installation_-_Personal_Package_Archives_%28PPA%29. [Último acceso: 4 Agosto 2021].

[36] elastic, «Elastic Agents,» [En línea]. Available: https://www.elastic.co/guide/en/fleet/current/elastic-agent-installation-configuration.html. [Último acceso: 4 Agosto 2021].

[37] elastic, «Install Elastic Agents,» [En línea]. Available: https://www.elastic.co/guide/en/fleet/current/elastic-agent-installation.html#install-fleet-managed-agent. [Último acceso: 4 Agosto 2021].

[38] Microsoft, «Windows Print Spooler Remote Code Execution Vulnerability,» 1 Julio 2021. [En línea]. Available: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-34527 . [Último acceso: 19 Julio 2021].

[39] gentilwiki, «mimikatz,» [En línea]. Available: https://github.com/gentilkiwi/mimikatz. [Último acceso: 7 Agosto 2021].

Anexo I – Archivo de configuración suricata.yaml

```
%YAML 1.1
 ---

 vars:
   address-groups:
     #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
     #HOME_NET: "[192.168.0.0/16]"
     #HOME_NET: "[10.0.0.0/8]"
     #HOME_NET: "[172.16.0.0/12]"
     HOME_NET: "any"

     #EXTERNAL_NET: "!$HOME_NET"
     EXTERNAL_NET: "any"

     HTTP_SERVERS: "$HOME_NET"
     SMTP_SERVERS: "$HOME_NET"
     SQL_SERVERS: "$HOME_NET"
     DNS_SERVERS: "$HOME_NET"
     TELNET_SERVERS: "$HOME_NET"
     AIM_SERVERS: "$EXTERNAL_NET"
     DC_SERVERS: "$HOME_NET"
     DNP3_SERVER: "$HOME_NET"
     DNP3_CLIENT: "$HOME_NET"
     MODBUS_CLIENT: "$HOME_NET"
     MODBUS_SERVER: "$HOME_NET"
     ENIP_CLIENT: "$HOME_NET"
     ENIP_SERVER: "$HOME_NET"

   port-groups:
     HTTP_PORTS: "80"
     SHELLCODE_PORTS: "!80"
     ORACLE_PORTS: 1521
     SSH_PORTS: 22
     DNP3_PORTS: 20000
     MODBUS_PORTS: 502
     FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
     FTP_PORTS: 21
     GENEVE_PORTS: 6081
     VXLAN_PORTS: 4789
     TEREDO_PORTS: 3544

 default-log-dir: /var/log/suricata/

 stats:
   enabled: yes
   interval: 8

 outputs:
   - fast:
       enabled: no
       filename: fast.log
       append: yes
       #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

   - eve-log:
       enabled: yes
       filetype: regular #regular|syslog|unix_dgram|unix_stream|redis
```

```
      filename: eve.json
      #threaded: false
      #prefix: "@cee: " # prefix to prepend to each log entry
      #identity: "suricata"
      #facility: local5
      #level: Info ## possible levels: Emergency, Alert, Critical,
                   ## Error, Warning, Notice, Info, Debug
      #ethernet: no  # log ethernet header in events when available
      #metadata: no
      pcap-file: false
      community-id: false
      community-id-seed: 0
      xff:
        enabled: yes
        mode: extra-data
        deployment: reverse
        header: X-Forwarded-For

      types:
        - alert:
            payload: yes                  # enable dumping payload in
Base64
            # payload-buffer-size: 4kb # max size of payload buffer to
output in eve-log
            # payload-printable: yes    # enable dumping payload in
printable (lossy) format
            # packet: yes                  # enable dumping of packet
(without stream segments)
            # metadata: no             # enable inclusion of app layer
metadata with alert. Default yes
            # http-body: yes              # Requires metadata; enable
dumping of HTTP body in Base64
            # http-body-printable: yes # Requires metadata; enable
dumping of HTTP body in printable format
            tagged-packets: yes
        - anomaly:
            enabled: yes
            types:
              # decode: no
              # stream: no
              # applayer: yes
            #packethdr: no
        #- http:
            #extended: yes       # enable this for extended logging
information
            #custom: [Accept-Encoding, Accept-Language, Authorization]
            # dump-all-headers: none
        #- dns:
            #version: 2
            #enabled: yes
            #requests: no
            #responses: no
            #formats: [detailed, grouped]
            #types: [a, aaaa, cname, mx, ns, ptr, txt]
        #- tls:
            #extended: yes       # enable this for extended logging
information
            #session-resumption: no
```

```
            #custom:  [subject,  issuer,  session_resumed,  serial,
fingerprint, sni, version, not_before, not_after, certificate, chain,
ja3, ja3s]
        #- files:
            #force-magic: no    # force logging magic on all logged
files
            #force-hash: [sha256]
        #- drop:
        #    alerts: yes      # log alerts that caused drops
        #     flows: all         # start or all: 'start' logs only a
single drop
        #- smtp:
            #extended:  yes  #  enable  this  for  extended  logging
information
            #custom: [received, x-mailer, x-originating-ip, relays,
reply-to, bcc]
            #md5: [body, subject]
        #- dnp3
        #- ftp
        - rdp
        #- nfs
        - smb
        #- tftp
        #- ikev2
        - dcerpc
        - krb5
        #- snmp
        #- rfb
        #- sip
        #- dhcp:
            #enabled: yes
            #extended: no
        #- ssh
        #- mqtt:
            #passwords: no
        #- http2
        #- stats:
            #totals: yes       # stats for all threads merged together
            #threads: no       # per thread stats
            #deltas: no        # include delta values
        #- flow
        #- netflow
        #- metadata

  - http-log:
      enabled: no
      filename: http.log
      append: yes
      #extended: yes     # enable this for extended logging information
      #custom: yes        # enable the custom logging format (defined
by customformat)
      #customformat: "%{%D-%H:%M:%S}t.%z %{X-Forwarded-For}i %H %m %h
%u %s %B %a:%p -> %A:%P"
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  - tls-log:
      enabled: no  # Log TLS connections.
      filename: tls.log # File to store TLS logs.
      append: yes
      #extended: yes     # Log extended information like fingerprint
```

```
      #custom: yes       # enabled the custom logging format (defined
by customformat)
      #customformat: "%{%D-%H:%M:%S}t.%z %a:%p -> %A:%P %v %n %d %D"
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'
      #session-resumption: no

  - tls-store:
      enabled: no
      #certs-log-dir: certs # directory to store the certificates files

  - pcap-log:
      enabled: no
      filename: log.pcap
      limit: 1000mb
      max-files: 2000
      compression: lz4
      #lz4-checksum: no
      #lz4-level: 0
      mode: normal # normal, multi or sguil.
      #dir: /nsm_data/
      #ts-format: usec # sec or usec second format (default) is
filename.sec usec is filename.sec.usec
      use-stream-depth: no #If set to "yes" packets seen after reaching
stream inspection depth are ignored. "no" logs all packets
      honor-pass-rules: no # If set to "yes", flows in which a pass
rule matched will stop being logged.

  - alert-debug:
      enabled: no
      filename: alert-debug.log
      append: yes
      #filetype: regular # 'regular', 'unix_stream' or 'unix_dgram'

  - alert-prelude:
      enabled: no
      profile: suricata
      log-packet-content: no
      log-packet-header: yes

  - stats:
      enabled: yes
      filename: stats.log
      append: yes       # append to file (yes) or overwrite it (no)
      totals: yes       # stats for all threads merged together
      threads: no       # per thread stats
      #null-values: yes  # print counters that have value 0. Default:
no

  - syslog:
      enabled: no
      #identity: "suricata"
      facility: local5
      #level: Info

  - file-store:
      version: 2
      enabled: no
      #dir: filestore
      #write-fileinfo: yes
      #force-filestore: yes
```

```
        #stream-depth: 0
        #max-open-files: 1000
        #force-hash: [sha1, md5]
        xff:
          enabled: yes
          mode: extra-data
          deployment: reverse
          header: X-Forwarded-For

    - tcp-data:
        enabled: no
        type: file
        filename: tcp-data.log

    - http-body-data:
        enabled: no
        type: file
        filename: http-data.log

    - lua:
        enabled: no
        #scripts-dir: /etc/suricata/lua-output/
        scripts:
        #   - script1.lua

logging:
  default-log-level: notice
  #default-log-format: "[%i] %t - (%f:%l) <%d> (%n) -- "
  default-output-filter:
  outputs:
  - console:
      enabled: yes
      # type: json
  - file:
      enabled: yes
      level: info
      filename: suricata.log
      # type: json
  - syslog:
      enabled: no
      facility: local5
      format: "[%i] <%d> -- "
      # type: json

af-packet:
  - interface: eth0
    threads: auto
    cluster-id: 99
    cluster-type: cluster_flow
    defrag: yes
    use-mmap: yes
    mmap-locked: yes
    tpacket-v3: yes
    #ring-size: 2048
    #block-size: 32768
    #block-timeout: 10
    #use-emergency-flush: yes
    #buffer-size: 32768
    #disable-promisc: no
    #checksum-checks: kernel
```

```
    #bpf-filter: port 80 or udp
    #copy-mode: ips
    #copy-iface: eth1

 #- interface: default
    #threads: auto
    #use-mmap: no
    #tpacket-v3: yes

#pcap:
  #- interface: eth0
    #buffer-size: 16777216
    #bpf-filter: "tcp and port 25"
    #checksum-checks: auto
    #threads: 16
    #promisc: no
    #snaplen: 1518

  #- interface: default
    #checksum-checks: auto

pcap-file:
  checksum-checks: auto

app-layer:
  protocols:
    rfb:
      enabled: yes
      detection-ports:
        dp: 5900, 5901, 5902, 5903, 5904, 5905, 5906, 5907, 5908, 5909
    mqtt:
      enabled: yes
      # max-msg-length: 1mb
      # subscribe-topic-match-limit: 100
      # unsubscribe-topic-match-limit: 100
    krb5:
      enabled: yes
    snmp:
      enabled: yes
    ikev2:
      enabled: yes
    tls:
      enabled: yes
      detection-ports:
        dp: 443
      ja3-fingerprints: auto
      #encryption-handling: default
    dcerpc:
      enabled: yes
    ftp:
      enabled: yes
      # memcap: 64mb
    rdp:
      enabled: yes
    ssh:
      enabled: yes
      #hassh: yes
    http2:
      enabled: no
    smtp:
```

```
      enabled: yes
      raw-extraction: no
      mime:
        decode-mime: yes
        decode-base64: yes
        decode-quoted-printable: yes
        header-value-depth: 2000
        extract-urls: yes
        body-md5: no
      inspected-tracker:
        content-limit: 100000
        content-inspect-min-size: 32768
        content-inspect-window: 4096
    imap:
      enabled: detection-only
    smb:
      enabled: yes
      detection-ports:
        dp: 139, 445
      #stream-depth: 0
    nfs:
      enabled: yes
    tftp:
      enabled: yes
    dns:
      tcp:
        enabled: yes
        detection-ports:
          dp: 53
      udp:
        enabled: yes
        detection-ports:
          dp: 53
    http:
      enabled: yes
      # memcap:                    Maximum memory capacity for HTTP
      # default-config:            Used when no server-config matches
      #   personality:            List of personalities used by default
      #   request-body-limit:     Limit reassembly of request body for
inspection
      #    response-body-limit:     Limit reassembly of response body
for inspection
      # server-config:            List of server configurations to use
if address matches
      #   address:                List of IP addresses or networks for
this block
      #   personality:              List of personalities used by this
block
      libhtp:
         default-config:
           personality: IDS
           request-body-limit: 100kb
           response-body-limit: 100kb
           request-body-minimal-inspect-size: 32kb
           request-body-inspect-window: 4kb
           response-body-minimal-inspect-size: 40kb
           response-body-inspect-window: 16kb
           response-body-decompress-layer-limit: 2
           http-body-inline: auto
           swf-decompression:
```

```
            enabled: yes
            type: both
            compress-depth: 100kb
            decompress-depth: 100kb
          #randomize-inspection-sizes: yes
          #randomize-inspection-range: 10
          double-decode-path: no
          double-decode-query: no
          #lzma-enabled: false
          #lzma-memlimit: 1mb
          #compression-bomb-limit: 1mb
          #decompression-time-limit: 100000
        server-config:
          #- apache:
          #    address: [192.168.1.0/24, 127.0.0.0/8, "::1"]
          #    personality: Apache_2
          #    request-body-limit: 4096
          #    response-body-limit: 4096
          #    double-decode-path: no
          #    double-decode-query: no
          #- iis7:
          #    address:
          #      - 192.168.0.0/24
          #      - 192.168.10.0/24
          #    personality: IIS_7_0
          #    request-body-limit: 4096
          #    response-body-limit: 4096
          #    double-decode-path: no
          #    double-decode-query: no
    modbus:
      #request-flood: 500
      enabled: yes
      detection-ports:
        dp: 502
      stream-depth: 0
    dnp3:
      enabled: yes
      detection-ports:
        dp: 20000
    enip:
      enabled: yes
      detection-ports:
        dp: 44818
        sp: 44818
    ntp:
      enabled: yes
    dhcp:
      enabled: yes
    sip:
      enabled: yes

asn1-max-frames: 256

# datasets:
#   defaults:
#     memcap: 100mb
#     hashsize: 2048

#run-as:
#  user: suri
```

```
#  group: suri
#sensor-name: suricata
#pid-file: /var/run/suricata.pid
#daemon-directory: "/"
#umask: 022

coredump:
  max-dump: unlimited

host-mode: auto
#max-pending-packets: 1024
#runmode: autofp
#autofp-scheduler: hash
#default-packet-size: 1514

unix-command:
  enabled: auto
  #filename: custom.socket

#magic-file: /usr/share/file/magic

#geoip-database: /usr/local/share/GeoLite2/GeoLite2-Country.mmdb

legacy:
  uricontent: enabled

# action-order:
#   - pass
#   - drop
#   - reject
#   - alert

#reputation-categories-file: /etc/suricata/iprep/categories.txt
#default-reputation-path: /etc/suricata/iprep
#reputation-files:
# - reputation.list

engine-analysis:
  rules-fast-pattern: yes
  rules: yes

pcre:
  match-limit: 3500
  match-limit-recursion: 1500

host-os-policy:
  windows: [0.0.0.0/0]
  bsd: []
  bsd-right: []
  old-linux: []
  linux: []
  old-solaris: []
  solaris: []
  hpux10: []
  hpux11: []
  irix: []
  macos: []
  vista: []
  windows2k3: []
```

```
defrag:
  memcap: 32mb
  hash-size: 65536
  trackers: 65535 # number of defragmented flows to follow
  max-frags: 65535 # number of fragments to keep (higher than trackers)
  prealloc: yes
  timeout: 60

#  host-config:
#
#    - dmz:
#        timeout: 30
#       address: [192.168.1.0/24, 127.0.0.0/8, 1.1.1.0/24, 2.2.2.0/24,
"1.1.1.1", "2.2.2.2", "::1"]
#
#    - lan:
#        timeout: 45
#        address:
#          - 192.168.0.0/24
#          - 192.168.10.0/24
#          - 172.16.14.0/24

flow:
  memcap: 128mb
  hash-size: 65536
  prealloc: 10000
  emergency-recovery: 30
  #managers: 1 # default to one flow manager
  #recyclers: 1 # default to one flow recycler thread

vlan:
  use-for-tracking: true

stream:
  memcap: 64mb
  checksum-validation: yes      # reject incorrect csums
  inline: auto                    # auto will use inline mode in IPS
mode, yes or no set it statically
  reassembly:
    memcap: 256mb
    depth: 1mb                  # reassemble 1mb into a stream
    toserver-chunk-size: 2560
    toclient-chunk-size: 2560
    randomize-chunk-size: yes
    #randomize-chunk-range: 10
    #raw: yes
    #segment-prealloc: 2048
    #check-overlap-different-data: true

host:
  hash-size: 4096
  prealloc: 1000
  memcap: 32mb

#ippair:
#  hash-size: 4096
#  prealloc: 1000
#  memcap: 32mb

decoder:
```

```
    teredo:
      enabled: true
      ports: $TEREDO_PORTS # syntax: '[3544, 1234]' or '3533' or 'any'.
    vxlan:
      enabled: true
      ports: $VXLAN_PORTS # syntax: '[8472, 4789]' or '4789'.
    vntag:
      enabled: false
    geneve:
      enabled: true
      ports: $GENEVE_PORTS # syntax: '[6081, 1234]' or '6081'.
    # max-layers: 16

detect:
  profile: medium
  custom-values:
    toclient-groups: 3
    toserver-groups: 25
  sgh-mpm-context: auto
  inspection-recursion-limit: 3000
  #delayed-detect: yes

  prefilter:
    default: mpm
  grouping:
    #tcp-whitelist: 53, 80, 139, 443, 445, 1433, 3306, 3389, 6666,
6667, 8080
    #udp-whitelist: 53, 135, 5060

  profiling:
    #inspect-logging-threshold: 200
    grouping:
      dump-to-disk: false
      include-rules: false      # very verbose
      include-mpm-stats: false

mpm-algo: auto
spm-algo: auto

threading:
  set-cpu-affinity: no
  cpu-affinity:
    - management-cpu-set:
        cpu: [ 0 ]  # include only these CPUs in affinity settings
    - receive-cpu-set:
        cpu: [ 0 ]  # include only these CPUs in affinity settings
    - worker-cpu-set:
        cpu: [ "all" ]
        mode: "exclusive"
        # threads: 3
        prio:
          low: [ 0 ]
          medium: [ "1-2" ]
          high: [ 3 ]
          default: "medium"
    #- verdict-cpu-set:
    #    cpu: [ 0 ]
    #    prio:
    #      default: "high"
  detect-thread-ratio: 1.0
```

```
luajit:
  states: 128

profiling:
  rules:
    enabled: yes
    filename: rule_perf.log
    append: yes
    limit: 10
    json: yes

  keywords:
    enabled: yes
    filename: keyword_perf.log
    append: yes

  prefilter:
    enabled: yes
    filename: prefilter_perf.log
    append: yes

  rulegroups:
    enabled: yes
    filename: rule_group_perf.log
    append: yes

  packets:
    enabled: yes
    filename: packet_stats.log
    append: yes
    csv:
      enabled: no
      filename: packet_stats.csv

  locks:
    enabled: no
    filename: lock_stats.log
    append: yes

  pcap-log:
    enabled: no
    filename: pcaplog_stats.log
    append: yes

capture:
  #disable-offloading: false
  #checksum-validation: none

default-rule-path: /var/lib/suricata/rules

rule-files:
  - suricata.rules

classification-file: /etc/suricata/classification.config
reference-config-file: /etc/suricata/reference.config
# threshold-file: /etc/suricata/threshold.config
```