UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Implementación de un SOC con la herramienta SIEM Elastic Security

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Martí Pastor, David

Tutor/a: Marco Gisbert, Héctor

CURSO ACADÉMICO: 2021/2022

Resumen

Los ataques informáticos están a la orden del día. Las organizaciones se encuentran continuamente amenazadas tanto desde Internet como desde la propia red interna. Por ello, una de las soluciones más extendidas para detectar y responder a estos ataques es mediante un Centro de Operaciones de Seguridad (SOC). Además de estudiar los problemas de seguridad informática, también se destacarán los desafíos encontrados al crear un SOC desde cero y escoger la tecnología correcta para desplegarlo.

La tecnología principal de un SOC es el gestor de información y eventos de seguridad (SIEM) que permite la monitorización de activos de una organización a través de alarmas. Se estudiarán algunas herramientas SIEM y se profundizará en Elastic Security, un SIEM innovador y *open source* de Elastic, así como en el resto de tecnologías complementarias como Elasticsearch, Kibana, Logstash y Beats.

En lo que se refiere a la parte práctica, se diseñará una arquitectura siguiendo las recomendaciones y buenas prácticas de Elastic y se implementará una solución SIEM de Elastic Security en un entorno virtual para poder experimentar con esta tecnología en un entorno de pruebas. Además, se ejemplificará la creación de reglas con 5 casos de uso reales extraídos del framework Mitre Att&ck.

Finalmente, se analizará el trabajo realizado y se expondrán los resultados obtenidos, valorando los objetivos logrados.

Palabras clave: Ciberseguridad, Defensa, SOC, SIEM, Elastic Security.

Abstract

Cyber attacks are the order of the day. Organizations are under constant threat both from the Internet and from their own internal network. Therefore, one of the most widespread solutions to detect and respond to these attacks is through a Security Operations Center (SOC). In addition to studying IT security issues, the challenges encountered in creating a SOC from scratch and choosing the right technology to deploy it will also be highlighted.

The core technology of a SOC is the Security Information and Event Management (SIEM) that allows the monitoring of an organization's assets through alarms. Some SIEM tools will be studied, especifically Elastic Security, an innovative and open source SIEM from Elastic, as well as the other complementary tools such as Elasticsearch, Kibana, Logstash and Beats.

Regarding the practical part, an architecture will be designed following Elastic's recommendations and best practices and an Elastic Security SIEM solution will be implemented in a virtual environment in order to experiment with this technology in a test environment. In addition, the creation of rules will be exemplified with 5 real use cases extracted from the Mitre Att&ck framework.

Finally, the work done will be analyzed and the results obtained will be presented, evaluating the objectives achieved.

Keywords: Cybersecurity, Defense, SOC, SIEM, Elastic Security.

Índice de contenidos

Índice de contenidos 5
Índice de figuras 7
Índice de tablas 9

1. Introducción 11
   1.1. Motivación 11
   1.2. Objetivos 12
   1.3. Estructura 13
2. Estado del arte 15
   2.1. Generaciones SOC 15
   2.2. Comparativa tecnológica actual 16
      2.2.1. Securonix 18
      2.2.2. Elastic Security 19
      2.2.3. LogRhythm 20
   2.3. Crítica al estado del arte 20
3. Fundamentos teóricos 23
   3.1. El Centro de Operaciones de Seguridad 23
      3.1.1. Componentes 23
      3.1.2. Servicios 24
   3.2. El Gestor de Información y Eventos de Seguridad 25
      3.2.1. Componentes 25
      3.2.2. Funciones 26
   3.3. Productos de Elastic Stack 26
      3.3.1. Elasticsearch 27
      3.3.2. Kibana 30
      3.3.3. Logstash 30
      3.3.4. Beats 31
   3.4. IPtables 31
   3.5. Infraestructura de clave pública 32
4. Análisis del problema 35
   4.1. Problemas de seguridad 35
   4.2. Desafíos en la implementación de un SOC 37
   4.3. Desafíos en el despliegue de un SIEM 38

4.4. Solución propuesta 38
5. Diseño de la solución 41
5.1. Presupuesto 41
5.1.1. Recursos humanos 41
5.1.2. Recursos tecnológicos 43
5.1.3. Aspectos de infraestructura 44
5.2. Arquitectura del SIEM 45
5.2.1. Aspectos arquitectónicos de Elastic 45
5.2.2. Arquitectura elegida 53
5.3. Convención de nombres 58
5.4. Casos de uso 59
5.5. Procesos 61
6. Desarrollo de la solución 63
6.1. Instalación de Elasticsearch y Kibana 63
6.2. Instalación y configuración de Logstash 65
6.3. Instalación y configuración de Beats 68
6.4. Creación de reglas 70
7. Pruebas 77
7.1. Pruebas de cifrado 77
7.2. Pruebas de reglas 78
8. Conclusiones 83
8.1. Conclusión 83
8.2. Futuras vías de trabajo 84
9. Glosario 87
10. Bibliografía 95

Anexo A: Objetivos de desarrollo sostenible 103
Anexo B: Configuración de elasticsearch.yml 105
Anexo C: Configuración de kibana.yml 107
Anexo D: Configuración de filebeat.yml 110
Anexo E: Configuración de winlogbeat.yml 117

Índice de figuras

Figura 1.- Generaciones de SOC 16
Figura 2.- Puntuación de soluciones SIEM (Gartner) 17
Figura 3.- Funcionalidades de seguridad incluidas por suscripción 19
Figura 4.- Componentes de un SOC. 24
Figura 5.- Entorno típico de un SIEM. 26
Figura 6.- *Elastic Stack* 27
Figura 7.- Estructura de datos de Elasticsearch (Enfoque ascendente) 28
Figura 8.- Estructuración de los shards en un clúster. 28
Figura 9.- Etapas del ciclo de vida de un índice. 29
Figura 10.- Estructura de datos de Elasticsearch (Enfoque descendente) 30
Figura 11.- Infraestructura de clave pública. 33
Figura 12.- Sueldo medio de los profesionales en ciberseguridad. 42
Figura 13.- Arquitectura básica de un SOC basado en Elastic. 46
Figura 14.- Arquitectura avanzada de un SOC basado en Elastic. 46
Figura 15.- Arquitectura avanzada de un SOC basado en Elastic con colas de mensajería. 47
Figura 16.- Arquitectura avanzada de un SOC basado en Elastic con Elastic APM. 47
Figura 17.- Segregación de clústeres de producción y monitorización. 48
Figura 18.- Implementación del clúster de monitorización en la nube. 48
Figura 19.- Replicación de clústeres por seguridad 49
Figura 20.- Replicación de clústeres para organizaciones internacionales. 49
Figura 21.- Replicación de clústeres para organizaciones centrales. 50
Figura 22.- Administración de los datos por fases. 51
Figura 23.- Arquitectura nivel básico para un entorno de pruebas. 52
Figura 24.- Arquitectura nivel medio-básico para un entorno de pruebas. 52
Figura 25.- Arquitectura nivel medio-avanzado para un entorno de pruebas. 53
Figura 26.- Arquitectura nivel avanzado para un entorno de pruebas. 53
Figura 27.- Arquitectura SIEM básica elegida. 54
Figura 28.- Arquitectura SIEM compleja elegida. 56
Figura 29.- Arquitectura SIEM del entorno de pruebas. 58
Figura 30.- Contraseña del superusuario elastic. 64
Figura 31.- Comprobación de la correcta ejecución de Elasticsearch. 64
Figura 32.- Generación del token de Kibana. 64
Figura 33.- Regla de IPtables para registrar cualquier conexión de red. 68
Figura 34.- Índices creados en el SIEM. 69
Figura 35.- *data streams* creados en el SIEM. 70
Figura 36.- Configuración de la regla de Escáner de puertos 71
Figura 37.- Definición de la regla de Escáner de puertos. 71
Figura 38.- Configuración de la regla de Ataque de fuerza bruta. 72
Figura 39.- Definición de la regla de Ataque de fuerza bruta. 72
Figura 40.- Configuración de la regla de monitorización de acceso a Kibana. 73
Figura 41.- Definición de la regla de monitorización de acceso a Kibana. 74
Figura 42.- Configuración de la regla de creación de cuentas en Windows. 74
Figura 43.- Definición de la regla de creación de cuentas en Windows. 75
Figura 44.- Configuración de la regla de detección de ataques DoS 75
Figura 45.- Definición de la regla de detección de ataques DoS 76

Figura 46.- Conjunto de alarmas creadas en Elastic Security. 76
Figura 47.- Cifrado Winlogbeat-Logstash..... 77
Figura 48.- Cifrado Filebeat-Logstash en logstash-VM..... 77
Figura 49.- Cifrado Filebeat-Logstash en elastic-VM..... 78
Figura 50.- Cifrado Logstash-Elasticsearch. 78
Figura 51.- Prueba de la regla TFG - SSH Brute Force Attack Detected. 79
Figura 52.- Prueba de la regla CST - Windows Account Created..... 80
Figura 53.- Prueba de la regla TFG - Volumetric DoS Attack Detected. 80
Figura 54.- Resultados de las pruebas realizadas. 81

Índice de tablas

Tabla 1.- Selección del modelo de suscripción de un SIEM. 18
Tabla 2.- Campos de los registros de IPtables 32
Tabla 3.- Presupuesto salarial anual de un SOC. 43
Tabla 4.- Fichero de configuración de *pipeline*s. 65
Tabla 5.- Configuración del *pipeline* Winlogbeat. 67
Tabla 6.- Configuración del *pipeline* Filebeat. 67
Tabla 7.- Expresiones temporales en Elastic. 73
Tabla 8.- Cálculos temporales con Elastic. 74
Tabla 9.- Prueba de la regla TFG - Port Scanning Detected. 78

1. Introducción

Actualmente, la ciberseguridad es una de las mayores inquietudes en las empresas e instituciones de todo el mundo, ya que un ataque exitoso, o la omisión de una normativa de seguridad/privacidad como la GDPR o la PCI-DSS, las puede llevar a grandes pérdidas económicas e incluso a la bancarrota (1).

Los cibercriminales utilizan técnicas de reconocimiento y escáneres para obtener la mayor superficie de ataque posible. En cualquier organización media se pueden encontrar varios vectores de ataque como páginas web y otros servidores accesibles al público, redes inalámbricas, dispositivos IoT y OT, móviles y otros dispositivos conectados a la red interna que no pertenecen a la propia organización, sino a sus empleados (denominados *shadow IT* en inglés). Así, con un vector de entrada y ayudados por vulnerabilidades *0-day* y *1-day*, ingeniería social y/o diversas técnicas de hacking, los atacantes consiguen entrar en los activos más críticos de las organizaciones (2).

Para frustrar los ataques o mitigar su impacto cuando han tenido éxito, se han inventado numerosas técnicas de defensa, tanto a nivel de red como a nivel de sistema:

Por un lado, se ha desarrollado software dedicado a defender los equipos de las amenazas de Internet, como los antivirus, los firewalls de host y los *endpoints*. También, se han desarrollado numerosas técnicas de mitigación y prevención de vulnerabilidades en el software ya existente, a nivel de sistema operativo como el NX, SSP o el ASLR (3).

Por otro lado, a nivel de red se ha desarrollado hardware y software dedicado a monitorizar y controlar el tráfico que fluye por la red. Elementos como los IDS, IPS y los firewalls son muy eficientes en alertar de tráfico sospechoso o bloquear tráfico malicioso.

Sin embargo, manejar la seguridad de todos los elementos de una empresa no es sencillo, existen muchas fuentes de información que hay que tener en cuenta y estas se multiplican exponencialmente con la cantidad de dispositivos que contiene la organización. Además, las herramientas y técnicas de seguridad tradicionales no son suficientes, ya que los cibercriminales inventan nuevas técnicas para evadirlas. Por ello, para correlacionar toda la información y administrar la seguridad de las organizaciones de forma eficiente, centralizada y organizada, se han establecido los Centros de Operaciones de Seguridad (SOC por sus siglas en inglés, *Security Operation Center*) (4).

A pesar de toda la tecnología y las estrategias desarrolladas para defender la seguridad de una organización, esta no estará completamente segura. Aunque se utilice la mejor tecnología, un SOC muy eficiente con un equipo experimentado y unas políticas de seguridad muy rigurosas, con el tiempo y los recursos suficientes siempre existirá la posibilidad de un ataque exitoso. No obstante, cuanto más difícil sea para un atacante comprometer la seguridad de la empresa o institución, habrá una mayor probabilidad de que este cambie de objetivo (2).

1.1. Motivación

La seguridad informática es uno de los temas más preocupantes en la actualidad. A menudo aparecen en las noticias ataques informáticos a empresas importantes que generan grandes pérdidas económicas y atentan contra la privacidad de la gente. Por ello, tenía dos proyectos de TFG en mente:

Por un lado, uno titulado “Privacidad y Anonimato en Internet”, enfocado sobre todo en la parte de anonimato, comentando técnicas y herramientas que se pueden utilizar para pasar desapercibido en Internet. Este proyecto no salió adelante porque el tutor que quería que me supervisara, por su experiencia en el campo y su usual disponibilidad a ayudar, tenía demasiados proyectos asignados este año. Y, por otro lado, el proyecto que se ha llevado a cabo, “Implementación de un SOC con la herramienta SIEM Elastic Security”.

Uno de los motivos esenciales que incentivaron la propuesta de este TFG es que muchos de los libros de la biblioteca del departamento de informática y plataformas online describen los conceptos clave de la seguridad informática, la metodología hacker o las herramientas más usadas por estos, pero no hay tanta información sobre cómo defenderse ante los ataques, ya que en general genera mucha menos curiosidad entre los estudiantes.

Hay que añadir que en la empresa donde trabajo están creando un SOC desde cero y a petición propia he sido involucrado desde la fase de diseño, ya que la fase de planificación fue llevada a cabo mediante reuniones de los directores de la empresa. Esta oportunidad fue decisiva a la hora de elegir el tema del TFG, puesto que podía ayudar en mi empresa y realizar un Trabajo de Fin de Grado de calidad y aplicable a la realidad.

Por último, Héctor Marco accedió a supervisar este trabajo, lo cual confirmó finalmente el tema, puesto que es uno de los pocos profesores que me ha impartido una asignatura dedicada exclusivamente a la seguridad en el grado y tiene mucha experiencia en el campo.

1.2. Objetivos

Este trabajo persigue los siguientes objetivos:

- OBJETIVO 1: Divulgar información de defensa informática.
- OBJETIVO 2: Documentar la creación de un prototipo de SOC de forma reproducible y asequible.
- OBJETIVO 3: Aprender a crear reglas en un SIEM.
- OBJETIVO 4: Mostrar las dificultades encontradas durante la implementación de un SOC.

El primer objetivo pretende despertar la curiosidad e informar al lector sobre conceptos básicos de seguridad informática, relacionados concretamente con la defensa. Para ello, conceptos como SOC, SIEM, firewall, IPS/IDS, etc. serán explicados en distintos grados de profundidad.

A través del segundo objetivo se desea crear un entorno de pruebas que permita experimentar con la herramienta SIEM de Elastic para distintos casos de uso de un SOC. Para que este objetivo sea cumplido, el TFG debe incluir el proceso de creación y que este sea repetible por el lector con ayuda de la documentación de Elastic.

El tercer objetivo persigue mostrar una de las funcionalidades más básicas de un SIEM, necesaria para el control de amenazas. Para lograr el cumplimiento de este objetivo se elegirán 5 casos de uso estándar y se crearán reglas que permitan monitorizarlos.

Finalmente, con el cuarto objetivo se busca documentar los desafíos que las organizaciones deben superar para llevar a cabo este proyecto en la vida real y poner en valor el trabajo del estudiante.

1.3. Estructura

El capítulo 1. Introducción ofrece una visión general de la seguridad y los mecanismos de defensa desarrollados para la detección de ataques. También, hay una sección de motivación donde se explican los posibles proyectos de TFG que se han propuesto y se justifica porque se ha escogido el actual. A continuación, se nombran y explican los objetivos que persigue este documento y la estructura del mismo.

El capítulo 2. Estado del arte explica la evolución de los Centros de Operaciones de Seguridad desde el inicio de Internet y el malware simple hasta la actualidad y muestra algunas de las tecnologías SIEM más usadas en el mercado. Además, analiza los trabajos realizados hasta la fecha en la UPV relacionados con el tema del TFG.

El capítulo 3. Fundamentos teóricos alinea los conocimientos del lector con los conceptos específicos del campo de la seguridad informática necesarios para entender el TFG correctamente y presenta las tecnologías que se usarán en la parte práctica del mismo.

El capítulo 4. Análisis del problema se explican algunos de los problemas de seguridad a los que se enfrentan las empresas hoy en día, y las tácticas y técnicas utilizadas por los atacantes para irrumpir en la infraestructura TI de las organizaciones. A continuación, se comentan los desafíos que se plantean al desarrollar un SOC y desplegar su herramienta principal, el SIEM. Finalmente, se propone una solución y se explican los pasos a realizar para llevarla a cabo.

El capítulo 5. Diseño de la solución comienza con el estudio del presupuesto necesario para arrancar el proyecto de implementar un Centro de Operaciones de Seguridad y desplegarlo durante alrededor de un año. Asimismo, se muestran los componentes y arquitecturas más recomendados de la herramienta SIEM elegida y se diseña la convención de nombres que se seguirá a lo largo del proyecto y los casos de uso derivados de las tácticas y técnicas de ataques explicadas en el capítulo 4, que se implementarán en el prototipo de SOC. Para acabar el capítulo, se nombrarán los procesos que debería tener un SOC maduro.

El capítulo 6. Desarrollo de la solución explica el proceso de instalación y configuración del SIEM en un entorno virtual, así como la creación de las reglas explicadas en los casos de uso del capítulo 5.

El capítulo 7. Pruebas explica las diferentes pruebas que se han realizado para generar alarmas de todas las reglas creadas en el capítulo anterior para demostrar que todas ellas funcionan correctamente y se muestran los resultados. También, se muestran evidencias de que el cifrado de las comunicaciones está activo.

El capítulo 8. Conclusiones analiza el trabajo realizado a lo largo del documento, se exponen los resultados obtenidos y se valoran los objetivos logrados. También, se sugieren varias líneas de investigación para complementar el presente documento.

El capítulo 9. Glosario define brevemente los términos que el lector pueda encontrar desconocidos debido al enfoque del TFG sobre un campo específico de la informática, la ciberseguridad.

El capítulo 10. Bibliografía expone las fuentes utilizadas para dar soporte a las definiciones, afirmaciones y análisis de todo el TFG.

Para terminar, al final de documento se encuentran los anexos, que poseen información que puede resultar de interés para aquellos lectores que deseen entrar en detalles en algunas secciones específicas.

2. Estado del arte

En este capítulo se pretende mostrar una visión general de los conocimientos actuales sobre el tema propuesto, profundizando además en la historia de los SOC. También, se estudiarán las diferentes alternativas de SIEM para ofrecer una base en la que apoyarse a la hora de elegir la solución adecuada.

En la sección 2.1. Generaciones SOC se explicará la evolución de los Centros de Operaciones de Seguridad, desde el inicio de Internet hasta la actualidad.

En la sección 2.2. Comparativa tecnológica actual se escogerán tres soluciones SIEM líderes en el mercado y se analizarán sus cualidades. Además, se comentarán algunos de los planes de suscripción más utilizados por este tipo de herramientas y su idoneidad según el modelo de negocio de la organización que pretenda obtenerlas.

En la sección 2.3. Crítica al estado del arte se comentan los diferentes trabajos relacionados con el tema del presente documento y en qué se diferencia este del resto.

2.1. Generaciones SOC

La primera generación de Centros de Operaciones de Seguridad comprende desde 1975 hasta 1995, es la más larga puesto que se inicia en los primeros años de Internet y en aquella época los ataques eran simples en comparación con la actualidad. Los SOC se centraban en proteger a las organizaciones y agencias gubernamentales de ataques de código malicioso de bajo impacto.

A medida que Internet se extendía, del mismo modo lo hacían las amenazas. Así nació la segunda generación de SOC, que abarca desde 1996 hasta 2001. En esa época, los ataques mejoraron en complejidad y fue necesaria la implementación de un SOC que permitiese detección de intrusiones. Para ello se desarrollaron los primeros SIEM y empezó la oferta de servicios MSSP.

Los ataques evolucionaron y surgió el uso de *bots* para robar identidades e información bancaria. Por ello, se desarrolló una nueva generación de SOC que comprendió desde 2002 hasta 2006. Los SOC comenzaron a realizar tareas para administrar las vulnerabilidades de sus clientes y responder a incidentes de seguridad.

La cuarta generación de SOC comprendió desde 2007 hasta 2012 e introdujo varias mejoras para manejar nuevas amenazas como el *hacktivismo*, el robo de propiedad intelectual y los APT. Entre estas mejoras se pueden observar el uso de Big Data para análisis de seguridad, el enriquecimiento de la información mediante fuentes externas y la monitorización continua.

Finalmente, a partir de 2013 comenzó la automatización de tareas con herramientas como los SOAR, la compartición de información y la generación de inteligencia para conseguir una mayor eficiencia y mejores resultados que en la generación anterior (4).

En la Figura 1, se puede observar la evolución de los SOC con el paso del tiempo:

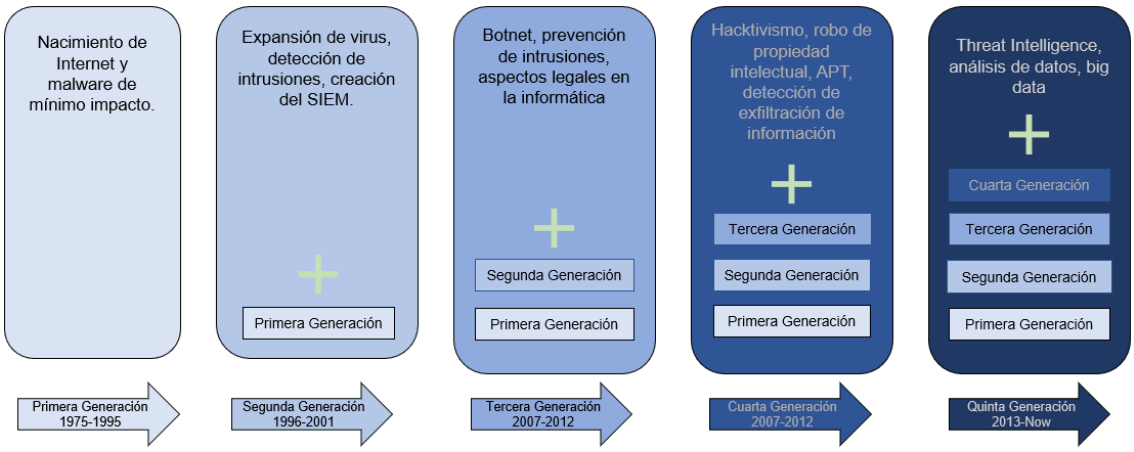

Figura 1.- Generaciones de SOC

2.2. Comparativa tecnológica actual

Las soluciones SIEM se adaptan y evolucionan constantemente según las necesidades del mercado para detectar, investigar y responder a amenazas y ataques y para cumplir con las leyes de seguridad informática. Estas tecnologías son complejas y requieren expertos en la materia y muchos otros recursos a la hora de evaluarlas, seleccionarlas, implementarlas y utilizarlas.

Existen muchas soluciones diferentes en el mercado y seleccionar la correcta para un negocio resulta muy complicado incluso para profesionales experimentados. En la Figura 2 podemos encontrar un gráfico de un estudio de la compañía Gartner donde puntúa las diferentes soluciones según un análisis realizado en abril de 2021.

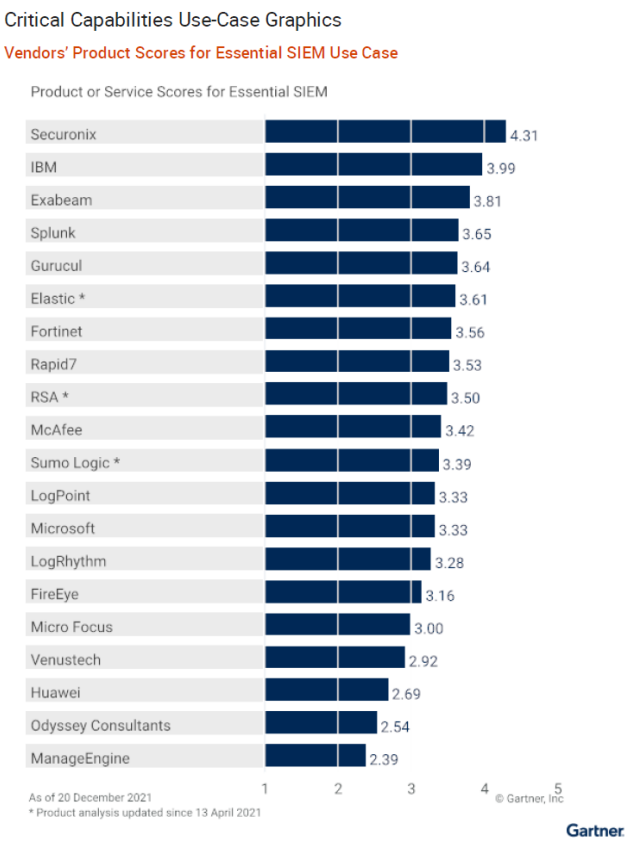

Figura 2.- Puntuación de soluciones SIEM (Gartner)

Se ha escogido el gráfico de la Figura 2 porque el objetivo del SIEM que se desea implementar en este TFG es detectar casos de uso esenciales para cualquier empresa u organización. A lo largo de todo el análisis de Gartner, la herramienta que más destaca es Securonix, de la que se hablará a continuación. Otras dos herramientas que pueden resultar interesantes son Elastic Security de Elastic, que es conocida por basarse en software libre y mantenerse en el top 10, y LogRhythm que lleva casi 20 años en el mercado y se encuentra en el top 20 (5).

En lo que se refiere al precio de los SIEM depende del proyecto, la empresa y el modelo de suscripción del SIEM. Los dos modelos de subscripción de SIEM más comunes son el basado por capacidad y el basado por usuario.

El primero es el modelo más clásico, se basa en pagar por cada cierta cantidad de recursos utilizados por el cliente (almacenamiento, eventos por segundo, etc.), el segundo es un modelo más reciente donde los clientes pagan por usuarios en el sistema por año, sin tener en cuenta los datos generados por cada usuario. Este último modelo es más sencillo de entender, pues es más fácil de relacionar con el crecimiento de la empresa.

Sin embargo, los dos modelos explicados anteriormente son inestables para empresas que esperan un crecimiento rápido o impredecible, porque a mayor crecimiento en la empresa, se necesitará mayor almacenamiento y datos procesados, así como un mayor número de empleados trabajando en el SIEM.

Por ello, se ha creado un tercer modelo que está tomando fuerza entre los gerentes de seguridad (*Chief Information Security Officer*, CISO) de las empresas, este modelo se conoce como modelo de procesamiento ilimitado. Elimina las restricciones de ingestión y procesamiento de datos o de usuarios en el sistema, configurando un precio predecible y estable a pesar del crecimiento experimentado en la empresa.

Otros modelos de suscripciones se han desarrollado en los últimos tiempos basados en las funcionalidades incluidas en la suscripción o el lugar donde se despliega la aplicación (en local o en la nube).

Como conclusión, a la hora de elegir el modelo de suscripción, hay que conocer las características de la empresa donde se desea implementar y seguir las recomendaciones de la Tabla 1 para poder contar con un precio estable de SIEM a lo largo del tiempo (6).

	Capacidad	Usuario	Ilimitado	Funcionalidad
Transformación Digital en proceso	☺	☺☺	☺☺☺	☺☺
Negocios de temporada	☺	☺	☺☺☺	☺
Infraestructura TI fija	☺☺☺	☺☺	☺	☺☺☺
Mejora en cumplimiento de leyes	☺	☺☺	☺☺☺	☺
Nº de empleados estable	☺☺	☺☺☺	☺	☺☺
Negocio en crecimiento	☺	☺	☺☺☺	☺

Tabla 1.- Selección del modelo de suscripción de un SIEM.

A continuación, se explicarán con más detalle las soluciones SIEM mencionadas al inicio de esta sección: Securonix, Elastic Security y LogRhythm.

2.2.1. Securonix

Securonix es una solución flexible, con un enfoque basado en el análisis de amenazas. Reduce los falsos positivos y detecta nuevas amenazas en tiempo real gracias a su producto Securonix Threat Lab.

Esta solución SIEM ofrece un gran control sobre la privacidad de los datos y los privilegios de usuarios, posee un gran número de *partners* que la apoyan con expertos en la materia y ofrece soporte a *Threat Hunting*. Sin embargo, es difícil de mantener y puede dar problemas de escalabilidad cuando se despliega en local (7).

Securonix también ofrece una amplia gama de productos para mejorar y ampliar el servicio como el análisis de comportamiento de usuarios (UEBA, por sus siglas en inglés *User and Entity Behavior Analytics*), una tecnología de detección y respuesta basada en *endpoints* con Open XDR y una herramienta de automatización de respuesta a incidentes (SOAR, por sus siglas en inglés *Security Orchestration, Automation and Response*)[1]. Además, Securonix también ofrece servicios gestionados y en la nube. Todo esto lo convierte en el SIEM mejor valorado por Gartner en 2021.

[1] https://www.securonix.com/es/

El precio de esta herramienta no se ha podido verificar desde la propia web del fabricante, puesto que normalmente depende del proyecto del cliente. Sin embargo, para tener una idea, Amazon Web Service (AWS) ofrece el servicio en la nube de Securonix desde $67.331 hasta $91.378, dependiendo del tipo y tiempo de almacenaje y de los productos adicionales contratados.

Como conclusión, se puede afirmar que Securonix es una solución estable y potente que posee múltiples características interesantes para un SOC.

2.2.2. Elastic Security

La solución de Elastic, conocida como Elastic Security, está compuesta por un conjunto de productos de código abierto que permiten obtener información de forma segura para buscar, visualizar y analizar en tiempo real (4). Ofrece también seguridad basada en *endpoints* con el producto Elastic Endgame y permite el despliegue en la nube a través de Elastic Cloud.

Entre sus ventajas se encuentra la oportunidad de comenzar de forma gratuita con funciones básicas de seguridad y mejorar la suscripción al cabo del tiempo si se desea. Además, a través de Kibana facilita el servicio de *Threat Hunting*. Sin embargo, es una herramienta con una curva de aprendizaje acentuada (7).

El precio discurre desde $1.140 al año con la suscripción estándar hasta $2.100 con la suscripción Enterprise. La suscripción más interesante para un SOC es la Platino por $1.500, ya que incluye protección contra *ransomware* basada en el comportamiento, detección de anomalías con *machine learning* y respuesta remota en hosts distribuidos en caso de que estos usen Elastic Endgame (8). En la Figura 3, se pueden observar las características de seguridad incluidas en cada plan de suscripciones.

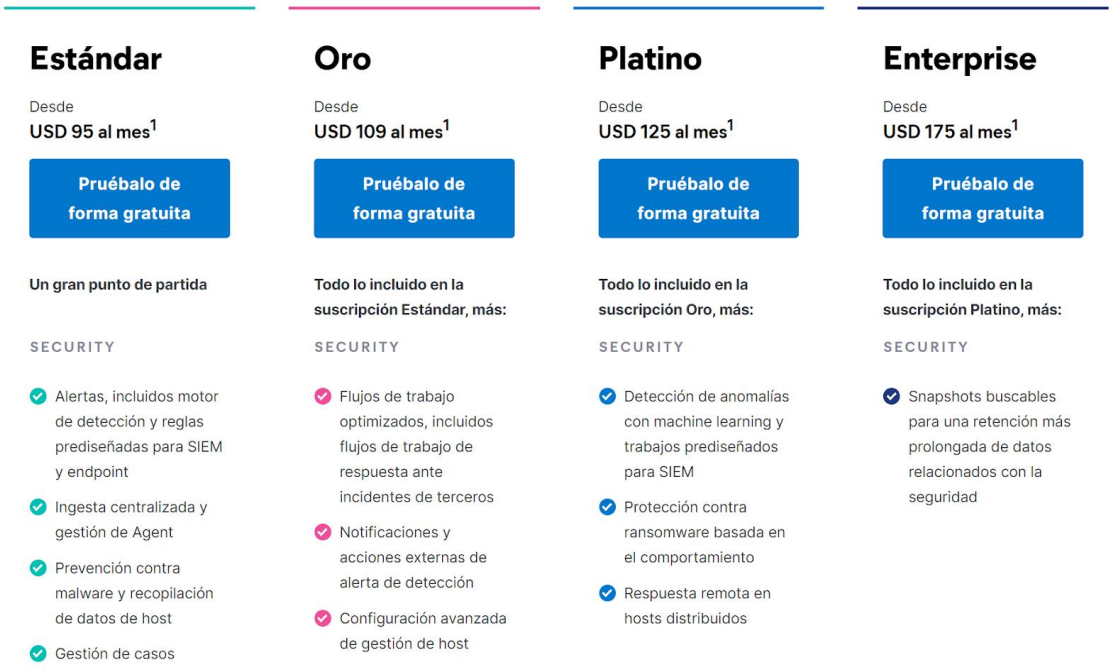

Figura 3.- Funcionalidades de seguridad incluidas por suscripción

En conclusión, Elastic Security es una herramienta con un balance calidad/precio muy bueno, de hecho, es de las pocas herramientas SIEM que disponen de un precio transparente y predefinido,

y se encuentra en la sexta posición del reporte de Gartner, por lo que sin duda es una solución a tener en cuenta.

2.2.3. LogRhythm

El SIEM de LogRhythm es una solución unificada y escalable diseñada para ayudar a las organizaciones a monitorizar y analizar su infraestructura. Es una solución integral que identifica y mitiga las amenazas y se recupera rápidamente de incidentes de seguridad (4).

LogRhythm posee una gran red de distribuidores que aumentan el soporte técnico de los clientes. Además, ofrece varias opciones para probarlo antes de comprarlo y un sistema de control y manejo casos con años de experiencia que permite realizar investigaciones siguiendo un flujo de trabajo inteligente. Sin embargo, es criticado por el sistema que utiliza al nombrar sus productos, el cual parece ser un tanto confuso (7).

En lo que se refiere al precio, LogRhythm ofrece el modelo de procesamiento ilimitado, pero no muestra ningún precio fijo en un web, por lo que para obtener presupuesto hay que contactar con el fabricante.

En conclusión, LogRhythm es una herramienta robusta con años de experiencia y un sistema de análisis de casos potente, lo que mejora la experiencia de usuario de los analistas. El mayor motivo por el que esta herramienta no está en el top 10 del análisis de Gartner, es la falta de soporte en la nube, que actualmente ya existe. LogRhythm ofrece una solución muy competente y un modelo de suscripciones adecuado para empresas con expectativas de crecimiento, pero el precio no es transparente y depende del proyecto a llevar a cabo.

2.3. Crítica al estado del arte

Para la investigación del estado del arte del tema presente en este TFG se ha hecho uso de la base de datos RiuNet[2], que es la base de datos que almacena los Trabajos de Fin de Grado y Master de la Universidad Politécnica de Valencia, ya que esta universidad posee no solo un grado dedicado a la ingeniería informática, sino también un master reciente enfocado específicamente a la ciberseguridad. Es de suponer pues, que los trabajos de investigación de sus estudiantes sean punteros y profesionales en materia de seguridad informática.

Tras analizar los trabajos de la plataforma RiuNet, se han encontrado tres tipos de TFG/TFM relacionados con la temática de este documento: obras sobre la implementación de un SOC, muy similares a este TFG, obras sobre el despliegue de un SIEM, una de las partes más importantes de este proyecto, y obras sobre Elasticsearch en otros ámbitos de uso, destacable puesto que es la tecnología principal de este trabajo.

En primer lugar, la obra “Implementación de un centro de operaciones de seguridad (SOC) de código abierto con elementos de red para sistemas industriales” de Mònica Martínez Gómez está muy relacionada con el trabajo realizado en este documento. Sin embargo, esta obra se centra en Elasticsearch, Kibana y Fleet y despliega dicha tecnología en contenedores Docker, mientras que este proyecto se enfoca en las herramientas del *ELK Stack* (Elasticsearch, Logstash, Kibana y Beats) y se despliega en máquinas virtuales de VirtualBox. Además, en lo

[2] https://riunet.upv.es/

que se refiere a la implementación de un SOC, la obra mencionada deja de lado los componentes clave de personal y procesos y se centra únicamente en la tecnología. Una diferencia fundamental entre ambos proyectos de temática similar es que en la obra de Mònica Martínez Gómez se explica el análisis de una alarma tras el despliegue del SIEM, mientras que en este trabajo no se entra en detalle en las investigaciones, haciéndose hincapié en la creación de las reglas (9).

En segundo lugar, se han encontrado varios trabajos dedicados únicamente a una tecnología SIEM, en concreto Splunk y Qradar. Ninguno de estos trabajos entra en detalles sobre el despliegue de la herramienta y se enfocan en el uso de la herramienta una vez está en funcionamiento, creando reglas y probándolas como en este TFG o realizando un pentest y analizando la información obtenida a través del SIEM (10). Un aspecto interesante encontrado en la obra "Implantación de Qradar en un entorno genérico multi cliente para SOC" por Alexis Sánchez Sanz es la referencia a la convención de nombres, a la que se refiere como taxonomía. Este es un punto que cubre este trabajo también en distintos aspectos y que se considera primordial para mejorar la eficiencia de los SIEM (11).

En tercer y último lugar, algunos estudiantes han utilizado la herramienta Elasticsearch con otros fines distintos a la seguridad informática como la monitorización del rendimiento de sistemas, o también, se han centrado en la migración de esta herramienta de local a en la nube. Estos trabajos no tienen tanto que ver con el tema del TFG sino con la tecnología utilizada en el mismo, y ninguno de ellos entra en profundidad en la estructura de almacenamiento de logs de Elasticsearch, al contrario que en el presente documento.

Finalmente, cabe destacar que en ningún otro trabajo relacionado se detallan los gastos que estas herramientas y los equipos que las gestionan pueden suponer para una empresa o se mencionan las alternativas de infraestructura que hay que tener en cuenta según el presupuesto. Sin embargo, este TFG supone que esta información es esencial para la dirección de una empresa antes incluso de que el proyecto arranque.

3. Fundamentos teóricos

La ciberseguridad es un campo específico de la informática que posee su propia literatura, jerga, metodologías y tecnologías. Para aquellos lectores que no estén familiarizados con estos conceptos, este documento posee un capítulo dedicado al glosario (Capítulo 11) donde se explican de forma muy breve todos los conceptos y acrónimos que pueden resultar confusos para un profesional TIC fuera del área de la seguridad informática.

Sin embargo, algunos conceptos son tan importantes para seguir correctamente los objetivos del TFG y el trabajo realizado que merecen una explicación más detallada. Por ello, este apartado pretende cubrir los conocimientos básicos necesarios para que un profesional con conocimientos técnicos generales propios del Grado pueda entender el resto del documento con mayor facilidad.

En la sección 3.1. El Centro de Operaciones de Seguridad se explicará que es un SOC, cuáles son sus componentes y que servicios puede ofrecer.

En la sección 3.2. El Gestor de Información y Eventos de Seguridad se explicará que es un SIEM, cuáles son sus componentes y que funciones puede desempeñar.

En la sección 3.3. Productos de *Elastic Stack* se mencionan y detallan los productos de Elastic que se van a trabajar en este documento: Elasticsearch, Kibana, Logstash y Beats.

3.1. El Centro de Operaciones de Seguridad

El SOC es una central de seguridad informática que monitoriza, administra y analiza constantemente las actividades en curso en los sistemas de información de una organización, como redes, servidores, sistemas, bases de datos, aplicaciones y sitios web.

Proporciona un punto único de control a través del cual se supervisan, evalúan y defienden los activos de la organización. Recopila la información de los registros obtenidos de los IDS/IPS, firewalls, *endpoints*, etc. y facilita la detección, investigación y respuesta a incidentes.

Su objetivo final es mantener la continuidad de una organización previniendo, detectando, identificando y respondiendo a las amenazas de seguridad antes de que afecten al negocio (4).

3.1.1. Componentes

Un Centro de Operaciones de seguridad está compuesto por Personas, Procesos y Tecnología:

Las personas son individuos especializados en las distintas funciones del SOC. Deben tener un gran conocimiento técnico, un amplio rango de habilidades y experiencia en distintos sectores de la seguridad.

Cada persona puede desempeñar uno o varios roles. Por ello, sus responsabilidades para con el SOC deben quedar claramente definidas. Su objetivo principal es vigilar y analizar la información que llega al SOC y comunicarse con el equipo de respuesta a incidentes en caso necesario.

Algunos de los puestos desempeñados en un SOC son: analista de seguridad, responsable de respuesta a incidentes, jefe de equipo, gerente y CISO. Aunque estos puestos varían según los servicios que se ofrecen.

Los procesos deben de actuar como un enlace entre las personas y la tecnología. Permiten que las personas del equipo correcto realicen las tareas correctas de forma eficiente. Sin procesos bien definidos, el SOC está sujeto al conocimiento de sus empleados y en caso de que estos no estén disponibles, podría llegar a parar la operatividad del SOC.

La tecnología amplía o limita las capacidades del SOC dependiendo de si facilita la automatización, la detección y prevención de amenazas, la clasificación de eventos, etc. Su objetivo principal es recopilar, almacenar, correlacionar y alertar sobre incidentes de seguridad.

Dentro de la tecnología utilizada en el SOC se incluyen elementos como el SIEM, los IDS/IPS, firewalls, sistemas de monitorización de actividades de bases de datos (DAM) o sistemas de tickets (4). En la Figura 4 se pueden observar los componentes de un SOC de forma gráfica:

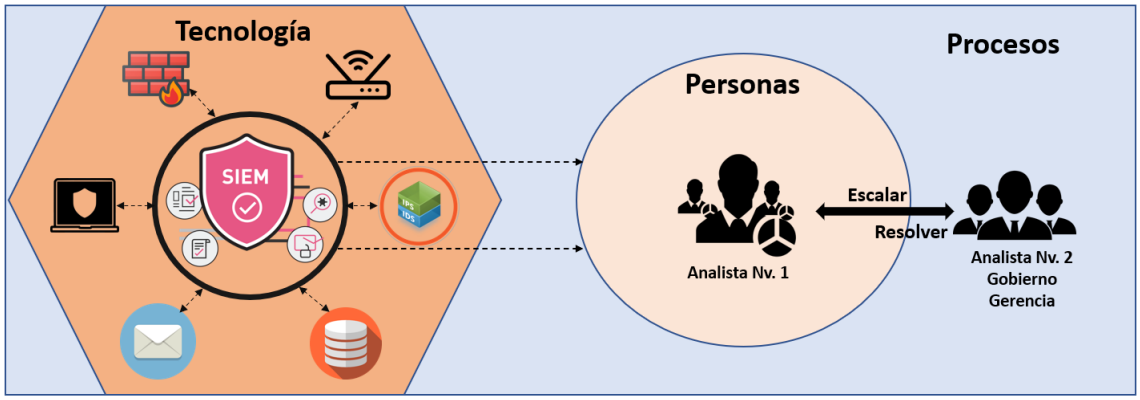

Figura 4.- Componentes de un SOC.

3.1.2. Servicios

Los servicios ofrecidos por un SOC maduro son muy diversos. Aquellos SOC que no ofrezcan uno de los servicios explicados a continuación, deberían estar preparados y tener presupuestos para subcontratar el servicio en caso necesario:

El servicio de gestión de riesgo identifica y decide cómo lidiar con los riesgos de la organización, desde la seguridad física hasta la educación de los empleados en el ámbito de la ciberseguridad.

El servicio de gestión de vulnerabilidades se encarga de los riesgos procedentes de vulnerabilidades técnicas como software no actualizado o errores de configuración.

El servició de respuesta a incidentes toma las medidas necesarias para mitigar el impacto de los eventos de seguridad que acontecen en los activos del cliente, es decir, decide qué acciones debe realizar el SOC cuando ocurre un evento determinado.

El servicio de análisis de malware realiza ingeniería inversa y análisis de vulnerabilidades y *exploits*. También analiza la causa, el remedio y la mitigación del malware.

El servicio de cumplimiento de la ley evalúa y mantiene los requisitos necesarios para cumplir con leyes y estándares, tales como la GDPR o ISO27001 respectivamente.

El servicio de análisis forense recoge pruebas tras un incidente para determinar su causa y tomar acciones legales.

El servicio de concienciación en ciberseguridad ofrece educación a los empleados del cliente del SOC relacionada con las amenazas potenciales de su entorno profesional.

El servicio de *threat hunting* investiga el panorama de las amenazas en constante evolución, desarrolla nuevas herramientas y técnicas y modifica las ya existentes para mejorar su eficacia (1).

3.2. El Gestor de Información y Eventos de Seguridad

El Gestor de Información y Eventos de Seguridad (SIEM por sus siglas en inglés, *Security Information and Events Management*) es un elemento básico del SOC. Ofrece monitorización y análisis de eventos en tiempo real y seguimiento de registros para propósitos de cumplimiento y auditoría.

El SIEM ayuda a las organizaciones a reconocer posibles amenazas y vulnerabilidades de seguridad de forma automatizada antes de que tengan la oportunidad de interrumpir las operaciones empresariales, revelando anomalías en el comportamiento de entidades y usuarios (UEBA) gracias al potencial de la Inteligencia Artificial y del *machine learning*.

Es un sistema de orquestación de datos altamente eficiente para administrar amenazas en constante evolución, así como para el cumplimiento normativo y la generación de informes (12).

3.2.1. Componentes

El entorno típico de un SIEM está compuesto de 4 componentes distintos: Las fuentes de datos, los agentes, el motor central y la base de datos.

Las fuentes de datos (*Log sources*) son las aplicaciones de los distintos dispositivos de una organización. La mayoría del software que corre en las aplicaciones puede generar registros y, en caso de que estos sean relevantes, son enviados al agente.

Hay cuatro tipos de fuentes de datos que suelen resultar interesantes: Los dispositivos de red como los routers, switches u ordenadores, las herramientas de seguridad como los firewalls, IPS/IDS o antivirus, los servidores como los servidores web, de correo o proxys y las aplicaciones como los buscadores.

Los agentes (*Log collectors*/*Agents*) son los dispositivos que reciben la información de los registros desde las fuentes de datos. Su objetivo es recolectar y normalizar dicha información antes de reenviarla al motor central.

El motor central (*Central engine*) es el lugar donde se correlacionan y analizan los datos y se alerta en caso de que ocurra alguna actividad sospechosa.

La correlación de los datos es un proceso basado en reglas, estadísticas o algoritmos que permite relacionar los eventos unos con otros. Por otro lado, el análisis de datos se refiere al proceso a

través del cual se identifican patrones y anomalías, que indican una intrusión o una violación de las políticas de la empresa, en los datos correlacionados.

La base de datos (*Database*) es el lugar donde se almacenan los registros de datos durante el periodo de tiempo especificado por la política de retención (4).

En la Figura 5 se muestran gráficamente los componentes del SIEM y su interacción.

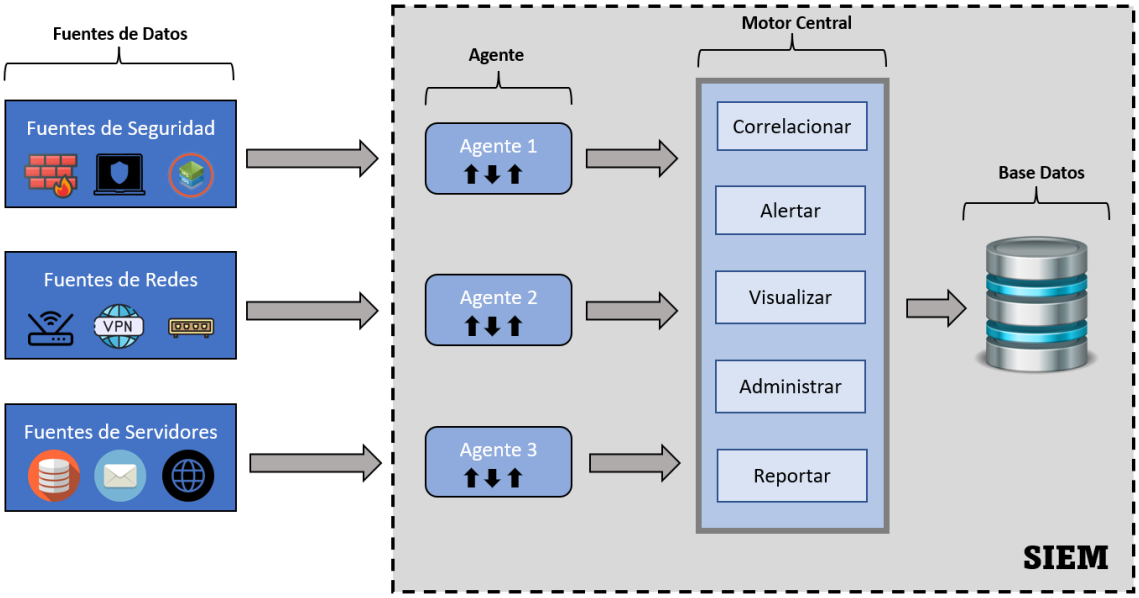

Figura 5.- Entorno típico de un SIEM.

3.2.2. Funciones

La herramienta SIEM facilita los servicios que ofrece el SOC. Entre las funciones del SIEM se puede encontrar la recolección de registros de datos, el análisis automático de los datos, la correlación de eventos, el análisis forense, las alertas en tiempo real, los *dashboards*, la generación automática de informes (generales y relativos al cumplimiento de una norma particular), la retención de los registros y la monitorización de sistemas, aplicaciones, actividades de usuario y accesos a ficheros y carpetas, así como de su integridad (4).

3.3. Productos de Elastic Stack

Elastic Stack es una colección de productos diseñada para ser confiable e indexar de forma segura los datos de cualquier fuente, en cualquier formato. Permitiendo búsquedas, análisis y visualizaciones de esos datos en tiempo casi real (13).

Esta tecnología permite ser desplegada en la nube mediante la tecnología Elastic Cloud, en servidores propios o en contenedores a través de Kubernetes. Además, su uso es muy variable, desde la búsqueda de documentos y el monitoreo de la infraestructura hasta la protección contra amenazas de seguridad (14).

Elastic Stack está compuesto por cuatro aplicaciones distintas: Elasticsearch, Kibana, Logstash y Beats, tal y como se puede apreciar en la Figura 6.

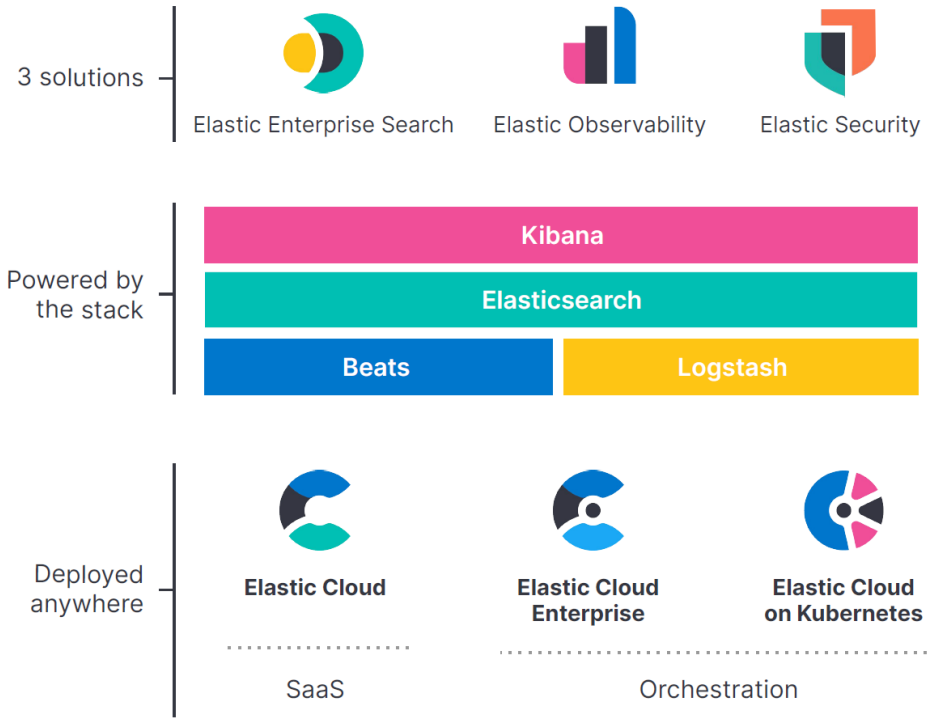

Figura 6.- *Elastic Stack*

3.3.1. Elasticsearch

Elasticsearch es el motor de análisis y búsqueda distribuida de *Elastic Stack*. Esta aplicación permite indexar datos de forma eficiente permitiendo rápidas búsquedas y análisis en tiempo casi real independientemente del tipo de estructura de los datos (15).

Elasticsearch se diseñó con dos objetivos principales, que fuese altamente escalable y usable:

Por un lado, la aplicación permite escalar horizontalmente, es decir, añadir tantos nodos como sea necesario a un clúster, ofreciendo el máximo rendimiento de los recursos puesto que es distribuida por diseño.

Por otro lado, permite que el cliente esté programado en cualquier lenguaje de programación debido a que ofrece API REST para comunicarse con el clúster a través del protocolo HTTP/S.

Elasticsearch no es una base de datos relacional y tiene una estructura para almacenar registros de datos propia que se explicará a continuación:

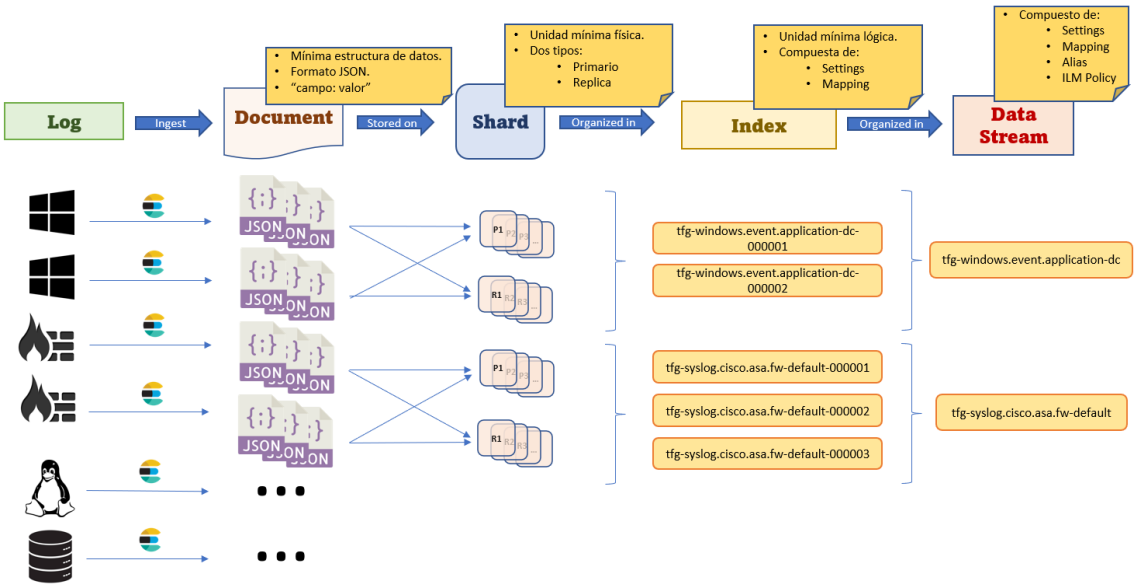

Figura 7.- Estructura de datos de Elasticsearch (Enfoque ascendente)

Como se puede ver en la Figura 7, un registro procedente de una fuente de datos como un firewall, una base de datos o un sistema Linux o Windows es enviado al clúster de Elastic. Para indexar el registro, Elastic crea un documento JSON con los campos del registro y algunos metadatos.

Estos documentos se almacenan dentro del clúster en *shards*, que es la unidad de almacenamiento física mínima, en caso de la existencia de más de un nodo con Elasticsearch en el clúster, si la configuración lo especifica, el documento se replica y se reparte en tantos nodos como lo exija la configuración, creándose así un *shard* primario y uno o varios *shards* replica. Es importante destacar que los *shards* primarios y réplica nunca estarán en un mismo nodo. Esta característica se puede apreciar en la Figura 8.

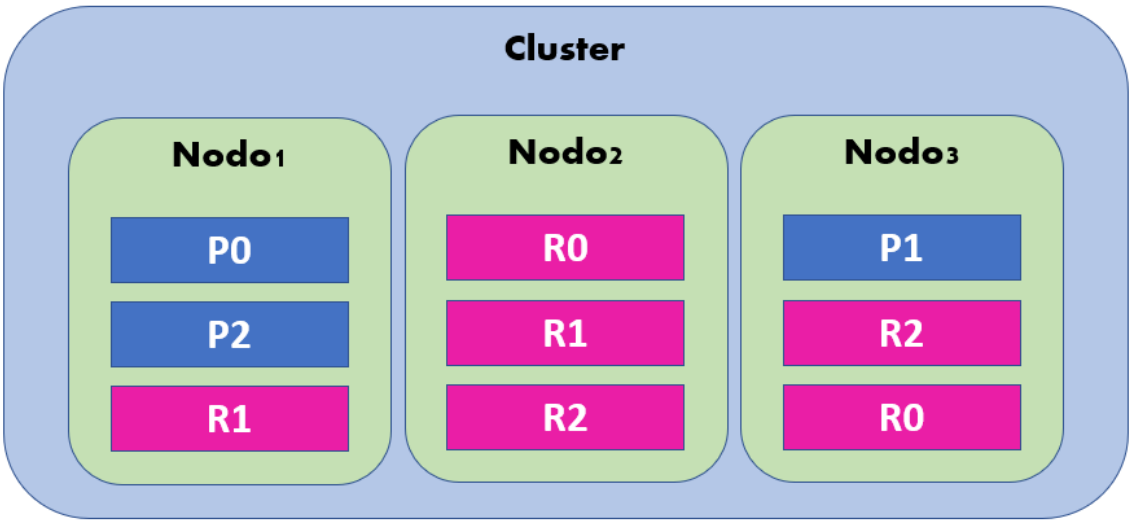

Figura 8.- Estructuración de los shards en un clúster.

Siguiendo la Figura 7, uno o varios shards primarios y sus respectivas réplicas son apuntados por un índice, que es la unidad lógica mínima de Elasticsearch. Los índices son las estructuras que se suelen utilizar para operar con la información del clúster y se definen por parámetros de

configuración en el campo *settings*, como por ejemplo el número de réplica *shards* que tendrá cada primario del índice, los campos de cada documento y el tipo de datos de sus valores en el campo *mappings* y un posible alias que agrupe a más de un índice en el campo *alias*.

Finalmente, hay una estructura adicional llamada *data stream*, la cual es una abstracción de los índices. Permite automatizar algunas funciones de los índices como la política de control del ciclo de vida de los índices (ILM, por sus siglas en inglés *Index Lifecycle Management*) y los alias. Al usar *data streams*, se crean nuevos índices con la misma configuración bajo un mismo alias periódicamente, o cuando el índice almacena cierta cantidad de documentos, o cuando llega a cierta cantidad de bytes. Entonces, al crearse un nuevo índice, este se nombra como el *data stream* seguido de un número y se convierte en el único índice de escritura de entre todos los que forman el *data stream*. De esta forma, todos los índices del *data stream* son de lectura y solo hay un índice de lectura y escritura donde se añaden los nuevos documentos.

Las políticas ILM automatizan el proceso de cambio de estado de un índice. Dependiendo del uso que se les dé a los datos, sería interesante almacenarlos de la forma más eficiente posible. Elasticsearch ofrece 5 etapas para los datos, que se explicarán a continuación tomando como referencia la Figura 9:

- *Content Data*: Almacena aquellos datos que forman parte del sistema, como las reglas, las alarmas, las políticas ILM, etc.
- *Hot Data*: Almacena los datos más nuevos y aquellos para los que se necesita un tiempo de búsqueda mínimo (de unos segundos). Aquellos nodos que contengan este tipo de datos necesitarán de recursos más caros como CPUs, memoria y disco más rápidos. En un SOC serían los datos más actuales de uno a cuatro días en el pasado.
- *Warm Data*: Almacena datos que tienen mucha probabilidad de ser consultados pero la velocidad es relativamente importante (de unos segundos a unos pocos minutos). En un SOC serían datos que pueden resultar de interés en algunas investigaciones de cuatro días a una semana en el pasado.
- *Cold Data*: Almacena los datos que tienen poca probabilidad de ser consultados y su tiempo de búsqueda no es muy importante (de unos pocos minutos a unas horas). En un SOC serían datos que puedan resultar de interés en casos poco comunes de una semana a uno o tres meses en el pasado.
- *Frozen Data*: Almacena el resto de los datos, información que raramente será consultada y el tiempo de búsqueda es irrelevante (de unas horas a días). En un SOC serían los registros almacenados por contrato durante meses y años con el fin de cumplir con ciertas leyes y tener pruebas en caso de litigio.

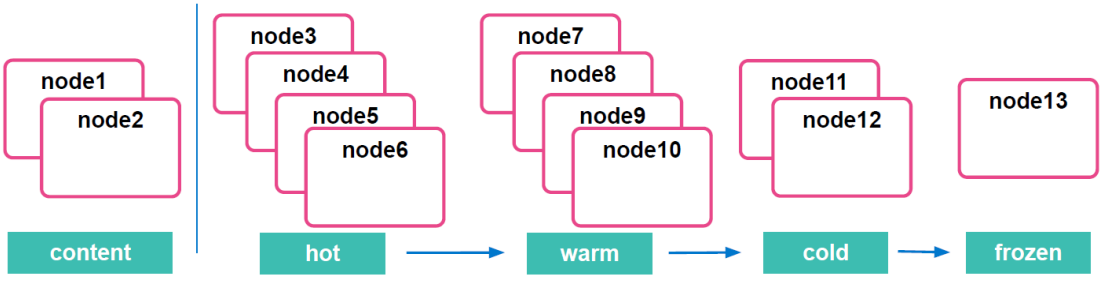

Figura 9.- Etapas del ciclo de vida de un índice.

Con el objetivo de afianzar lo anteriormente explicado, en la Figura 10 se puede observar la estructura desde un enfoque distinto.

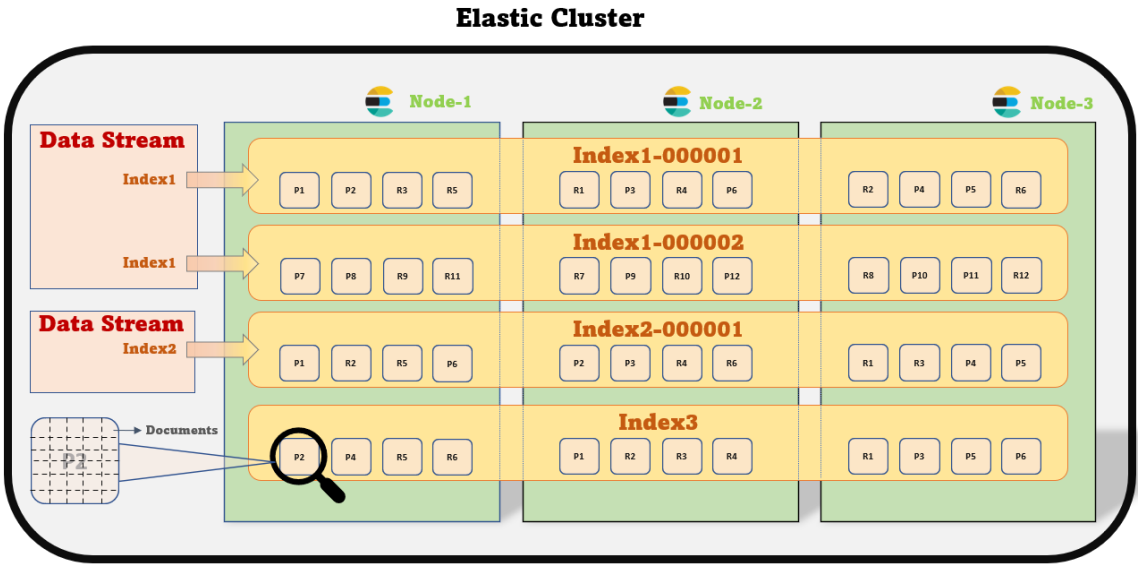

Figura 10.- Estructura de datos de Elasticsearch (Enfoque descendente)

Para finalizar este apartado sobre la tecnología Elasticsearch, es necesario mencionar los distintos roles que un nodo puede desempeñar en el clúster:

- *Master*: Es obligatorio y se encarga de las operaciones CRUD de índices, nodos del clúster, shards y del estado del clúster. Para una plataforma tolerante a fallos, es necesario tener tres nodos que puedan desempeñar este role.
- *Data*: Permite indexar y buscar información en el clúster.
- *Ingest*: Permite normalizar, enriquecer y preprocesar los datos ejecutando *pipelines*.
- *Machine Learning*: Permite realizar trabajos pesados de *machine learning*.
- *Transform*: Permite hacer uso de las funcionalidades de los *transforms* de Elastic que permiten enriquecer los documentos con información. Esta función es la recomendada por Elastic para mantener la relación entre documentos a pesar de no ser una base de datos relacional.
- *Remote cluster client*: Permite la conexión con clientes remotos para que hagan uso de sus APIs.

3.3.2. Kibana

Kibana es la interfaz de usuario del *Elastic Stack*. Permite crear gráficos para visualizar los datos, navegar a través de ellos y administrar la plataforma.

Kibana ofrece la posibilidad de ver los datos sin formato, mediante gráficos, mapas o *canvas*, analizar los datos de forma automática mediante reglas o *machine learning*, y administrar una gran cantidad de configuraciones del clúster como los índices, *data streams*, los ciclos de vida de índices (ILM), usuarios, roles, *pipelines*, etc (16).

Una de las estructuras más importantes de Kibana son los *data views*, antes conocidos como *index pattern*. Los *data views* son filtros de índices, al buscar en Kibana en los mapas, gráficos o los registros directamente, se usan *data views* para filtrar aquellos índices a los que se desea aplicar la consulta.

3.3.3. Logstash

Logstash es un agente o *Log Collector* que permite la recolección de datos de fuentes dispares, la normalización y enriquecimiento de la información recogida y el envío de esta al destino elegido a través de unos ficheros de configuración denominados *pipelines* (17).

Los *pipelines* están formados por tres partes, la entrada o *input*, es donde se especifica el puerto, *pipeline* o servicio por el que Logstash debe estar escuchando en espera de nuevos registros que vayan a ser procesados por el *pipeline*. El filtro o *filter*, es donde ocurre todo el procesamiento, la normalización y enriquecimiento de los registros que pasan por el *pipeline*. Y la salida u *output*, que especifica el *endpoint* al que se va a enviar el registro procesado, normalmente un nodo de Elasticsearch o Logstash (18).

3.3.4. Beats

Los Beats son transportadores de datos en forma de agentes que se instalan en los servidores clientes para enviar registros a Logstash o directamente a Elasticsearch (19). Hay siete tipos de beats dependiendo del tipo de información que se desee obtener:

- **Auditbeat**: Recopila los datos de auditoría de Linux y controla la integridad de sus archivos. Auditbeat envía estos eventos en tiempo real al resto del *Elastic Stack* para su posterior análisis (20).
- **Filebeat**: Envía los registros almacenados en un archivo a Logstash o directamente a Elasticsearch. Si se interrumpe la conexión de red, Filebeat recuerda el último registro enviado y sigue a partir de ahí, evitando pérdidas de registros. Filebeat posee módulos para múltiples tecnologías que permiten la creación automática de gráficos y normalizadores. Además, posee un mecanismo de *back-pressure*, por el cual no sobrecarga al servidor en caso de grandes volúmenes de datos (21).
- **Heartbeat**: Monitoriza la disponibilidad de un servicio mediante sondeos periódicos a través de ICMP, TCP o HTTP/S (22).
- **Metricbeat**: Recopila métricas de sistemas y servicios. Estas métricas abarcan uso de CPU, memoria, entrada/salida de disco y de red y uso del sistema de archivos. Además, posee numerosos módulos para obtener métricas de distintos servicios (23).
- **Packetbeat**: Monitoriza el flujo de tráfico de red de forma pasiva controlado la latencia y los errores de las aplicaciones, los tiempos de respuesta y los patrones y tendencias de acceso de los usuarios (24).
- **Winlogbeat**: Transmite los registros de eventos de Windows a Logstash o Elasticsearch ofreciendo una amplia visión de lo que sucede en una infraestructura basada en Windows (25).
- **Functionbeat**: Desplegado como una función en un proveedor en la nube FaaS, recopila y monitoriza datos del servicio en la nube (26).

3.4. IPtables

IPtables es un software de firewall basado en reglas que permite aceptar, rechazar, registrar y modificar el tráfico entrante y saliente de la red. Contiene un conjunto de tablas que ayudan al procesamiento de los paquetes a través de múltiples cadenas que describen las acciones a realizar sobre el paquete (4).

Al activar la función de registro de paquetes se debe definir una regla cuya tabla sea INPUT y la acción (*target*) sea LOG. El formato de un registro de IPtables es el siguiente:

<Fecha> <Tiempo HH:MM:SS> <Nombre de la máquina> <Acción> <Entrada> <Salida> <IP fuente><IP destino> <Longitud> <TOS> <PREC> <TTL> <ID> <Frag> <PROTO> <SPT> <DPT> <Ventana> <RES> <SYN> <URGP>

Los campos de los registros de IPtables se explican en la Tabla X.

Campo	Descripción	Campo	Descripción
Fecha	Fecha de envío del log	PREC	3 primeros bits del TOS
Tiempo	Tiempo de envío del log	TTL	*Time to live*
Nombre de la máquina	Nombre de la máquina donde se generó el log	Acción	(Permitir, Denegar, Bloquear, etc.)
ID	ID del paquete	FRAG	Fragmentación
Entrada	Interfaz de red de entrada	PROTO	Tipo de protocolo
Salida	Interfaz de red de salida	SPT	Puerto fuente
IP fuente	Dirección IP fuente	DPT	Puerto destino
IP destino	Dirección IP destino	Ventana	Tamaño de ventana
Longitud	Longitud del paquete	RES	Reservado
TOS	Tipo de Servicio	URGP	Urgente

Tabla 2.- Campos de los registros de IPtables.

3.5. Infraestructura de clave pública

La infraestructura de clave pública (PKI, por sus siglas en inglés *Public Key Infrastructure*) es una arquitectura de seguridad diseñada para aumentar la confidencialidad de la información que circula por Internet. Esta arquitectura ayuda a relacionar las claves públicas con la identidad de usuario o entidad a la que corresponde.

En la arquitectura PKI, un sujeto (empresa, usuario o sistema) solicita un certificado a una autoridad de registro (RA). Tras recibir la petición, la RA verifica la identidad del sujeto y solicita el certificado a una autoridad de certificación (CA). La CA genera una clave pública y otra privada con el mismo algoritmo. La clave privada se la entrega únicamente al sujeto que ha solicitado el certificado y la clave pública se publica para que el resto de usuarios puedan cifrar mensajes con ella. Aquellos mensajes cifrados con la clave pública sólo podrán ser descifrados mediante la clave privada, proporcionando la confidencialidad deseada.

Además, el sujeto puede autenticarse firmando sus mensajes con su certificado de clave pública. Entonces, el receptor puede comprobar la validez del certificado y la identidad del emisor a través de una autoridad de verificación (VA), que almacena los certificados con sus claves públicas, y compara el certificado obtenido con el almacenado (2). En la Figura 11 se muestra la arquitectura PKI mediante un diagrama.

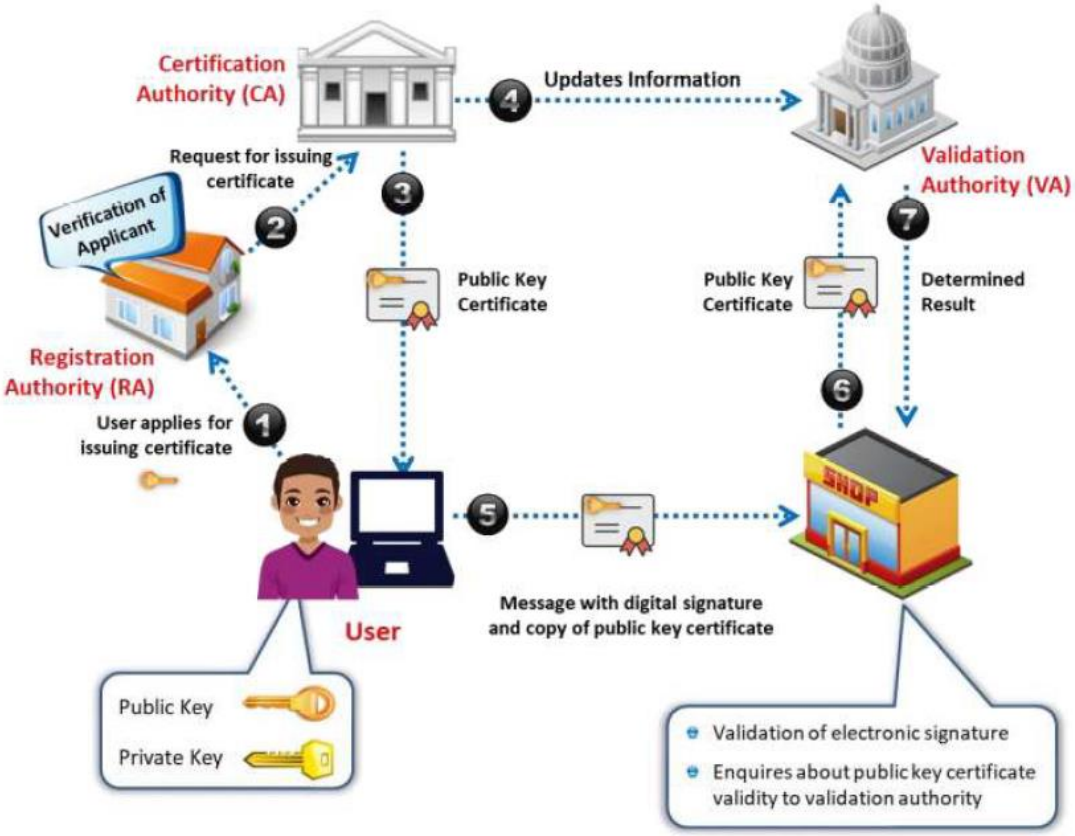

Figura 11.- Infraestructura de clave pública.

Un certificado auto-firmado, por el contrario, no necesita ninguna autoridad verificadora. Se puede generar con herramientas como OpenSSL de forma gratuita, pero ningún usuario de Internet reconocerá el certificado CA la primera vez que lo utilice y desconfiará del mismo.

4. Análisis del problema

En este capítulo se pretende explicar la necesidad de una solución de seguridad robusta y eficiente que mitigue el impacto de las amenazas informáticas a las que esté expuesta una empresa y las dificultades encontradas a la hora de implementar dicha solución.

En la sección 4.1. Problemas de seguridad se explicará la importancia de la seguridad informática a través del impacto de una brecha de seguridad y la creciente frecuencia de ataques como el ataque de Denegación de Servicio. También se estudiarán algunas técnicas que usan los atacantes para vulnerar y aprovechar la intrusión en las redes y sistemas de las víctimas a través del framework de Mitre Att&ck.

En la sección 4.2. Desafíos en la implementación de un SOC se comentarán los retos que han de superar los directivos para iniciar el proyecto de creación de un SOC.

En la sección 4.3. Desafíos en el despliegue de un SIEM se nombran los problemas que se deben evitar al desplegar un SIEM.

En la sección 4.4. Solución propuesta se mencionarán a grandes rasgos los pasos a seguir para llevar a cabo este proyecto.

4.1. Problemas de seguridad

El impacto de un incidente de seguridad en una organización es muy grande y puede afectar negativamente a sus operaciones de distintas formas.

Por un lado, el impacto económico directo asociado a una brecha de seguridad es potencialmente enorme. Según IBM la media del coste total alcanza los 4.24 millones de dólares, un 10% más que el año pasado (27). Además, las brechas de seguridad generan desconfianza en los clientes, *partners* y *stakeholders* de la organización y originan mala reputación. En caso de que la brecha conlleve la violación de leyes como la protección de datos personales o de las tarjetas de crédito, la organización responsable puede sufrir grandes multas e incluso la cárcel para el último responsable.

Tras una brecha de seguridad, la organización afectada se puede enfrentar a la pérdida de personal y de datos. Muchas organizaciones responden internamente despidiendo o reubicando a los trabajadores responsables, lo que dificulta la operatividad debido a que se reemplaza personal crítico en una industria con un limitado talento cualificado. La pérdida de datos por su parte puede dar ventaja a los competidores, en caso de que se divulgue información propietaria de la compañía o puede llevar al robo de identidad de los trabajadores, en caso de robo de información personal (1).

Por otro lado, los ataques de denegación de servicio aumentan cada año, especialmente en 2022 con la guerra entre Rusia y Ucrania. Según Kaspersky, a lo largo del primer cuatrimestre del 2022 se han detectado ataques de denegación de servicio distribuidos (DDoS) contra recursos importantes del ministerio de defensa ucraniano y organizaciones rusas, generando confusión y ralentizando las operaciones gubernamentales. También, a raíz de este conflicto, algunos

hacktivistas han desarrollado aplicaciones y páginas web para que cualquier usuario de internet pueda formar parte del ataque.

Además del conflicto ucraniano, otras naciones han sufrido ataques DDoS en lo que va de año, como Corea del Norte (Tras unas pruebas con misiles a mitad de junio), Israel (Tumbando páginas gubernamentales y al mayor proveedor ISP del país, Bezeq), Andorra (Durante el torneo de Minecraft basado en el juego del calamar) y muchas organizaciones internacionales, generando así pérdidas económicas cuantiosas (28).

Para poder exfiltrar datos o llevar a cabo cualquiera de sus objetivos, los cibercriminales siguen diversas técnicas de ataque que les permiten penetrar o paralizar el negocio de su objetivo. Se han invertido muchos esfuerzos en generar modelos y patrones comunes a la mayoría de los ataques, Mitre Att&ck[3] es un proyecto de Mitre que recoge las tácticas y técnicas de los atacantes basándose en observaciones del mundo real. Las 222 técnicas de Mitre distribuidas en 14 tácticas exceden el alcance de este TFG, por lo que nos centraremos en 5 de ellas:

- [*Discovery*] *Network Service Discovery* (T1046): Para obtener los servicios activos en los dispositivos del objetivo, los atacantes usan métodos como el escáner de puertos y vulnerabilidades.

  Esta información es de gran utilidad para un atacante puesto que les permite conocer aquellos servicios vulnerables a explotaciones de código remoto.

- [*Credential Access*] *Brute Force: Password Guessing* (T1110.001): Los atacantes pueden obtener las credenciales de un usuario legítimo de un sistema o aplicación probando sistemáticamente contraseñas. Este tipo de ataques se suelen realizar a servicios como SSH, Telnet, FTP, NetBIOS, Kerberos, etc.

  Una vez con acceso al sistema, los atacantes pueden intentar escalar privilegios, conseguir persistencia o pivotar a otros sistemas mediante movimiento lateral.

- [*Initial Access*] *Valid Accounts: Local Accounts* (T1078.003): Una vez los atacantes obtienen las credenciales de una cuenta pueden hacer uso de sus privilegios para llevar a cabo sus objetivos.

  En caso de que se filtren las credenciales de una aplicación crítica como la del SIEM, los atacantes podrían llegar a comprometer la confidencialidad, integridad y disponibilidad de los datos de todas aquellas empresas monitorizadas por dicha herramienta, produciendo un resultado catastrófico.

- [*Persistence*] *Create Account: Local Account* (T1136.001): Las cuentas locales se configuran para los usuarios de una organización, ofrecer soporte remoto o la administración del sistema o de uno de sus servicios.

  Los atacantes pueden crear cuentas locales para establecer un acceso remoto persistente sin necesidad de desplegar herramientas específicas en el sistema.

- [*Impact*] *Network Denial of Service: Direct Network Flood* (T1498.001): Al lanzar un ataque de denegación de servicio (DoS), los atacantes pretenden tumbar los sistemas objetivo enviando un gran volumen de tráfico de red, de forma que la víctima no pueda procesar todas las peticiones simultáneamente.

[3] https://attack.mitre.org/

Un ataque DoS puede ocultar diversos motivos: ego, venganza, económicos, políticos y sociales, e incluso para desviar la atención del verdadero ataque subyacente.

Como se ha visto en la introducción, las herramientas y técnicas de seguridad convencionales no son suficientes para defender a una organización de un ataque debido a que los atacantes inventan nuevas técnicas para evadir tales defensas y a la ingente cantidad de dispositivos dispersos por su infraestructura. Por ello, es necesaria una tecnología capaz de mantener el perímetro de la organización siempre actualizado frente a las nuevas y cambiantes amenazas y vulnerabilidades. Esto es posible mediante el SOC.

Antes de comenzar a crear un SOC se deben aclarar algunos datos como el alcance del proyecto o el presupuesto con el que se cuenta (1). Una vez aprobado el proyecto, el gerente del proyecto y el del SOC se enfrentan a varios desafíos, tanto en la implementación del SOC como en el despliegue del SIEM, que deberán tener en cuenta y resolver para evitar que el servicio resultante sea disfuncional.

4.2. Desafíos en la implementación de un SOC

En primer lugar, la falta de profesionales competentes es uno de los mayores desafíos a los que se enfrenta la implementación de un SOC (4). Hoy en día existen más puestos de trabajo que profesionales con habilidades certificadas en la industria de la seguridad y esta situación aumenta continuamente debido al crecimiento de competidores en la industria (1).

En segundo lugar, en el inicio de la vida de un SOC, las reglas y las listas blancas (*whitelists*) se ajustan lentamente y los analistas encargados de administrar las reglas de detección todavía se encuentran adaptándose al entorno de sus clientes. Estos factores conducen a un incremento preocupante de alarmas, en su mayoría falsos positivos, que los analistas tienen que revisar, generando una gran sobrecarga operativa.

En tercer lugar, el SOC debe gestionar los registros de muchas herramientas de seguridad distintas distribuidas por toda la red. Una buena idea para superar este desafío es utilizar un agente que sirva como punto único de recepción de registros, así será más fácil administrar y analizar las operaciones de seguridad.

En cuarto lugar, hay que tener en cuenta la regulación que debe cumplir un SOC. El SOC gestiona la seguridad y los registros de los clientes y debe cumplir con distintas leyes, estándares y buenas prácticas de la industria.

En quinto lugar, la selección correcta de la tecnología que formará parte del SOC es muy importante y una mala configuración puede llevar a una monitorización insuficiente de los incidentes de seguridad.

También, hay que tener en cuenta la integración de los nuevos procesos del SOC con los ya existentes en la empresa, el entrenamiento continuo de los empleados para estar a la altura de las nuevas amenazas y la dificultad de gestionar los ataques avanzados organizados en varias etapas (4).

Finalmente, muchos Centros de Operaciones de Seguridad mantienen dos entornos con el SIEM desplegado. Uno de pruebas para probar tareas de mantenimiento como actualizaciones, la creación de nuevas reglas, nuevas funcionalidades, etc. Y otro de producción donde se monitoriza el entorno del cliente. La creación del entorno de pruebas es otro problema común en la implementación de los SOC, puesto que llevan tiempo y recursos.

4.3. Desafíos en el despliegue de un SIEM

Se debe tener en cuenta algunas cosas antes de comenzar a desplegar la herramienta SIEM en una organización:

La falta de profesionales a pesar de tener el SIEM completamente operativo hace que los recursos y el dinero invertido en el proyecto sean en vano. Además, contratar nuevos empleados cuando el SIEM está funcionando puede llevar a que estos no exploten todo el potencial que ofrece la herramienta.

La configuración de todas las fuentes de datos a la vez puede dificultar el trabajo de los propios empleados del SIEM. Esta integración debería hacerse de forma escalonada y siempre justificando la agregación de una nueva fuente.

Finalmente, hay que valorar los costes que se derivan de desplegar y mantener el SIEM. Los costes de licencia cuando se instala, los costes de implementarlo, de optimizarlo y de mantenerlo cuando se acaba la licencia, los costes de entrenamiento de los empleados y los costes asociados a la expansión de la herramienta en caso necesario (4).

4.4. Solución propuesta

Como se ha visto al inicio del capítulo, el imparable incremento y la rápida evolución de los ataques informáticos hacen necesarias técnicas de monitorización, detección, mitigación y prevención de amenazas y vulnerabilidades. Para ello existen muchas propuestas que defienden la red y los dispositivos de empresas y particulares desde distintos ángulos.

Una de las soluciones cada vez más extendidas en empresas y organizaciones es el Centro de Operaciones de Seguridad. Esta central de seguridad permite recoger la información de las distintas soluciones de seguridad desplegadas por toda la infraestructura de la empresa, como firewalls, IDS/IPS, *endpoints*, *honeypots*, etc. Y generar inteligencia correlacionando los datos. De esta forma, se pueden detectar ataques informáticos en todas sus etapas para frustrarlos o mitigarlos.

En este TFG se pretende implementar un prototipo de SOC que permita detectar el uso de las técnicas de ataque descritas anteriormente, es decir, escáneres de puertos, ataques de contraseñas por SSH, acceso no autorizado a una aplicación crítica, creación de una cuenta local en un sistema Windows y ataques de denegación de servicio con el fin de ofrecer una base de seguridad basada en la monitorización, un entorno de pruebas seguro y un punto de partida firme para conocer los recursos necesarios para el despliegue del SIEM en un entorno de producción, pues no es un dato que los fabricantes suelen ofrecer. Para ello, se seguirán las cuatro fases típicas de un proyecto: planificación, diseño, implementación y pruebas. Sin embargo, en el TFG estas cuatro fases estarán contenidas únicamente en tres puntos, ya que planificación y diseño serán explicadas en el mismo capítulo.

Por un lado, se estudiará el presupuesto que un proyecto de este tamaño requiere. Entrando en detalles en los gastos de personal, así como los del SIEM y se seleccionará el más conveniente para el proyecto. Sin embargo, tan solo se mencionarán los gastos de infraestructura, puesto que son muy variables y aplicarlos o no y en qué grado depende completamente de las preferencias y necesidades de la dirección. A continuación, se diseñará la arquitectura de la red, tanto del entorno de producción como del de pruebas. Para ello, será necesario un análisis de las buenas prácticas de Elastic y el estudio de las posibilidades que este ofrece. Además, se explicará la

importancia de una convención de nombres estándar para la jerga del SOC, para la terminología del SIEM y para los campos de los registros. También se detallará el diseño de los casos de uso con las fuentes de datos necesarias y los servidores donde se monitorizarán y se explicará brevemente en que consiste un proceso y que debe contener. Con todo lo anteriormente mencionado, se habrán cubierto las fases de planificación y diseño del proyecto.

Por otro lado, en lo que se refiere a la implementación, se desplegará la herramienta SIEM y todos sus componentes, obteniendo los registros de las fuentes de datos especificadas en los Casos de Uso y se crearán las reglas necesarias para defender una empresa de las técnicas de Mitre Att&ck expuestas al inicio de este capítulo. También se asegurará la confidencialidad de los datos cifrando todas las comunicaciones entre los componentes del SIEM y se automatizará la administración de la información a través de *data streams*. Por último, se comprobará que todo lo desplegado en la fase anterior funcione correctamente: los componentes del SIEM, el cifrado de las comunicaciones, el funcionamiento de las reglas, y los *data streams*. Así se habrán cubierto las fases de implementación y pruebas del proyecto.

5. Diseño de la solución

En este capítulo se completarán las dos primeras fases del proyecto: la fase de planificación y la fase de diseño. Se obtendrá un presupuesto aproximado a la realidad y se diseñará la arquitectura de red, la convención de nombres y los casos de uso. Finalmente, se dará una visión general de los procesos que deberán ser implementados en un SOC.

5.1. Presupuesto

La inversión necesaria si se desea implementar un SOC en una empresa es enorme. Por lo que la organización económica en un proyecto de semejante magnitud es un factor clave para el éxito de la operación. En caso de una planificación del presupuesto negligente, el proyecto puede resultar cancelado debido a la gran inversión económica y temporal inicial.

Para la implementación del SOC se han dividido los gastos en tres tipos distintos: humanos, tecnológicos y de infraestructura.

En la sección 5.1.1. Recursos humanos se planteará el horario y el equipo iniciales idóneos durante los primeros meses de implementación de un SOC. Además, se detallará los gastos anuales derivados del salario del equipo.

En la sección 5.1.2. Recursos tecnológicos se valorarán las alternativas SIEM propuestas, teniendo en cuenta el gasto asociado a cada una de ellas y se seleccionará aquella que se considere más apropiada para el proyecto.

En la sección 5.1.3. Aspectos de infraestructura se comentan diversos factores que impactan en la retención laboral y la disponibilidad, eficiencia y seguridad del SOC que deberá valorarse si se desean implementar y en qué grado con el consecuente aumento de los gastos.

5.1.1. Recursos humanos

En este apartado se pretende seleccionar los recursos humanos necesarios para el proyecto, así como los gastos derivados de estos, y ofrecer una visión más clara de los servicios que se desean implementar al inicio de la vida del nuevo SOC.

Las profesiones necesarias en un SOC dependen de los servicios que este ofrezca. En la dirección se necesita un gerente de ciberseguridad CISO, un gerente del SOC y dependiendo del tamaño del SOC, uno o varios jefes de equipo. En la parte técnica, se pueden encontrar profesionales como analistas, *threat hunters*, forenses, etc. En la Figura 12, obtenida de una encuesta salarial de Exabeam en 2019 (29), se puede observar el salario medio recibido en la mayoría de los empleos mencionados. Teniendo en cuenta que los técnicos en ciberseguridad cobran cifras similares, podremos deducir el coste medio de los salarios anuales una vez se conozca el equipo inicial del SOC.

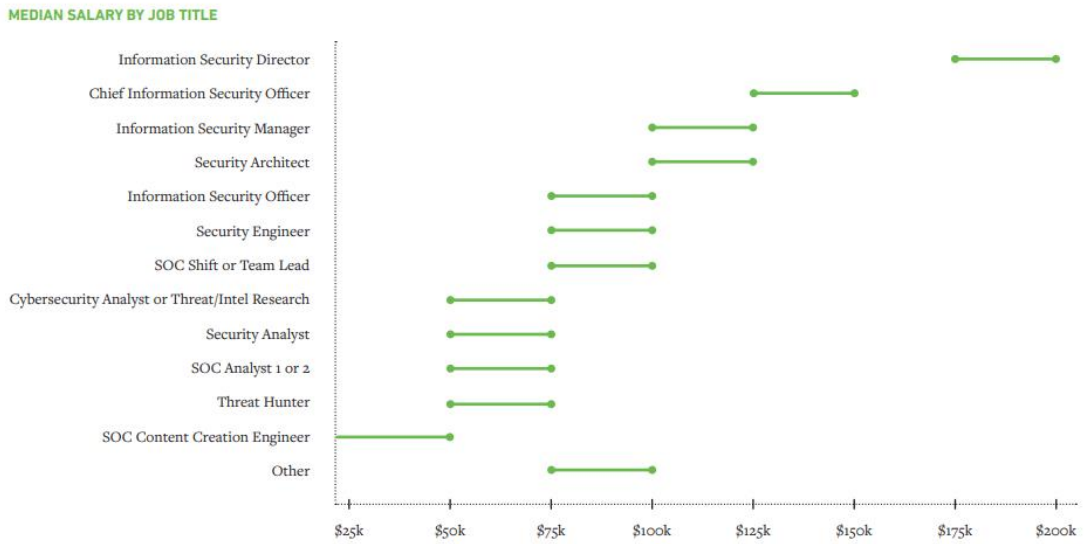

Figura 12.- Sueldo medio de los profesionales en ciberseguridad.

En un SOC en proceso de creación o que acaba de comenzar, el primer servicio que se ofrece es el de monitorización, por lo que los técnicos que se contratan son analistas capaces de administrar el SIEM y monitorizar el entorno de la empresa. Teniendo en cuenta que los servicios ofrecidos por el SOC son críticos, se suelen emplear turnos para dar servicio de monitorización 24/7, lo que implica la adquisición de una plantilla de técnicos de entre seis y doce empleados.

Además, debido a la gran cantidad de trabajo que conlleva el despliegue y la administración del SIEM durante los primeros meses, el SOC se enfrenta a la sobrecarga operativa de sus empleados, que conlleva inevitablemente a su insatisfacción laboral. Para evitar esto, en una industria con escasez de profesionales, se han analizado tres soluciones:

- Subcontratar empleados técnicos que refuercen y apoyen el trabajo de los empleados en nómina hasta que la carga operativa disminuya. La mayor ventaja de la subcontratación es que la empresa obtiene profesionales expertos en la tecnología a implementar en el momento y los nuevos empleados pueden aprender de la mano de estos expertos. La mayor desventaja es que la subcontratación de empleados es más cara y el conocimiento de la infraestructura creada se va con ellos.
- Ampliar el negocio progresivamente aumentando el número de clientes o recursos que monitorizar a medida que la sobrecarga operativa inicial disminuya. La mayor ventaja de esta opción es el aumento de ingresos e influencia en la industria. La mayor desventaja es que se debe contratar una plantilla inicial grande esperando resultados lentos.
- Ampliar los servicios del SOC a medida que los ya existentes se encuentren en un estado estable. La mayor ventaja de esta opción es la mejora progresiva de la madurez del SOC y el aumento de ingresos. La mayor desventaja es que se debe contratar una plantilla inicial grande esperando resultados lentos.

Para la planificación del SOC de este TFG se propone comenzar con el servicio de monitorización doce horas al día durante cinco días a la semana, puesto que el horario 24/7 solo tiene sentido una vez el servicio de monitorización se encuentra en funcionamiento. Por ello, se ha decidido contratar seis técnicos y subcontratar dos durante los primeros cuatro meses, con la finalidad de tener un técnico experto y tres analistas en nómina en cada turno (mañanas y tardes). También se contrarán dos jefes de equipo, uno por turno, y a medida que aumente el número de clientes y servicios a ofrecer, también aumentará el número de empleados en nómina.

A partir de la información anterior se puede obtener la Tabla 3, a través de la cual se puede deducir que el presupuesto mínimo para la contratación de personas al inicio de la vida de un SOC con servicio de monitorización 12/5 debe rondar los 725.000 dólares anuales.

Nº	CARGO	SALARIO
1	Gerente de Ciberseguridad	$125.000
1	Gerente del SOC	$100.000
2	Jefe de Equipo	$75.000
6	Analista de Seguridad	$50.000
2	Analista de Seguridad (Subcontratado)	$25.000[4]
	Total	$725.000[5]

Tabla 3.- Presupuesto salarial anual de un SOC.

Sin embargo, el gasto relacionado con las personas no se reduce a sus salarios. Uno de los mayores problemas analizados en el capítulo anterior es la escasez de personal en la industria y la pérdida de rendimiento entre un profesional que ha desplegado la solución SIEM y conoce cómo esta se adapta al entorno de la empresa y otro que, a pesar de ser un experto en la herramienta, no puede explotar todo su potencial en el entorno donde está desplegada. Por ello, para no perder personal, se ha de invertir en mejorar las condiciones laborales, gastos que se mencionarán en la sección de aspectos de infraestructura, y para mejorar el rendimiento de los nuevos empleados se ha de invertir en su entrenamiento con la herramienta SIEM, gastos que se analizarán en la sección de recursos tecnológicos.

Para finalizar con los recursos humanos, para la implementación del prototipo bastará con una sola persona con conocimientos generales del grado de ingeniería informática que haya leído y entendido el capítulo 3. Fundamentos teóricos.

5.1.2. Recursos tecnológicos

En este apartado se pretende seleccionar la tecnología a utilizar a lo largo del proyecto, en concreto se valorarán las soluciones SIEM propuestas en el capítulo 3. Fundamentos teóricos.

Las tres soluciones SIEM mencionadas son compatibles con los objetivos del SOC que se desea implementar. El SIEM de Securonix es el que ofrece un servicio más completo y variado, Elastic ofrece el mejor balance calidad/precio y LogRhythm ofrece un SIEM robusto con una interfaz y procesos internos fruto de la larga experiencia del producto.

Los cursos de entrenamiento para empleados y el precio del examen es también un factor a tener en cuenta:

Securonix posee cuatro certificados, de los cuales al menos tres son necesarios para un analista de seguridad profesional (CSSA, CSAD, CSDI) para administrar, desplegar y analizar en la herramienta. Cada certificado cuesta $800 y no posee un curso oficial asociado, en total serían $2.400 en certificados por cada empleado, es decir, $14.400 entre los seis empleados

[4] Considerando que un analista de seguridad subcontratado cobra el máximo salario percibido en su profesión según la Figura 12 ($75000/año) durante los primeros cuatro meses.
[5] Calculado a partir de la suma de los salarios por el número de empleados de cada puesto de trabajo.

contratados el primer año. Esta cantidad sumada al precio estimado mínimo de la herramienta se quedaría en $81.731 (30).

Elastic posee tres certificados necesarios para administrar y desplegar la herramienta correctamente. El precio de dos de ellos es de $400 y el tercero $300. Además, dado que Elastic es un conjunto de herramientas complejo, cada certificado tiene un curso asociado que ronda los $2.400, por ello, cada certificado se quedaría en alrededor de los $2.800. Si se desea contratar únicamente los exámenes, entonces se quedaría un precio de $1.100 por empleado, es decir, $6.600 el primer año, en cambio, si se contrata también los cursos, como se recomienda en este TFG, puesto que los exámenes son totalmente prácticos y tienen fama de ser buenos y complicados, el precio se queda en $8.300 por empleado, es decir, $49.800 el primer año. Esta cantidad sumada al precio estimado mínimo de la herramienta se quedaría en $51.300 (31).

LogRhythm posee cuatro certificados, de los cuales tres son necesarios en un SOC y el cuarto únicamente en caso de que se despliegue LogRhythm en la nube. Al igual que con el precio de la herramienta, no hay detalles del precio en la web oficial, por lo que se debe contactar con el fabricante para pedir presupuesto (32).

Como los tres ofrecen la funcionalidad de crear reglas y monitorizar la infraestructura de sus clientes, se ha decidido optar por la solución más barata a corto y largo plazo, es decir, Elastic. Además, el entorno de pruebas o prototipo será completamente gratuito, puesto que no es necesaria la suscripción al producto para utilizar su funcionalidad básica.

5.1.3. Aspectos de infraestructura

Tras concretar los recursos humanos y tecnológicos, el siguiente desafío es definir los requerimientos físicos necesarios, tales como la localización física del SOC, los requisitos energéticos, el espacio para alojar los equipos, la disposición de los asientos, como se maneja la basura, consideraciones de la seguridad física y aspectos importantes de los lugares de trabajo como el tipo de sillas o los monitores, etc.

La localización física del SOC impacta en varios factores, hay que tener en cuenta el barrio donde se encuentra, la distancia a la que se encuentra de restaurantes y del departamento de bomberos, los aparcamientos alrededor, si tiene posibilidad de ampliación. etc. Algunas organizaciones lo sitúan cerca del Centro de Operaciones de Redes (NOC) o del equipo de asistencia técnica, para crear así una sinergia entre ambos equipos. También hay que tener en cuenta las restricciones específicas de cada país en materia de gestión de datos, como leyes o aspectos culturales.

En lo que se refiere al interior, debe ser un lugar que permita la renovación de sus componentes de forma sencilla, que facilite las reuniones tanto internas como de visitantes externos al SOC y que favorezca la salud, seguridad y comodidad de sus empleados con un espacio bien iluminado, una acústica clara que permita conversaciones privadas y concentración en un ambiente de trabajo concurrido, sillas de oficina regulables, monitores grandes y demás consideraciones. Estos requisitos no son obligatorios, pero mejorarán drásticamente la retención de los empleados.

Ciertos tipos de información, como documentos clasificados o evidencias forenses, necesitan unos requisitos de almacenamiento muy específicos, lo que implica la necesidad de una caja fuerte o incluso una sala entera con una protección mayor. La basura también puede contener información confidencial, por lo que la compra de cubos que trituran el papel es otro requisito importante para evitar la filtración de información confidencial.

Para eludir el riesgo de denegación de servicio ante la caída de algún sistema y aumentar el rendimiento, la redundancia de los sistemas críticos es un factor importante que debe ser estudiado y que acarrea un aumento considerable de los gastos.

Finalmente, la seguridad física debe ser parte del diseño de la infraestructura del SOC, permitiendo el control de acceso a sus recursos de forma granular. Algunas de las opciones más comunes son candados, tarjetas de acceso inteligentes y cámaras de seguridad (1).

5.2. Arquitectura del SIEM

A lo largo de este apartado se comentarán y explicarán distintas arquitecturas básicas y recomendaciones de Elastic sobre aspectos como la monitorización, la replicación, la autenticación, la confidencialidad y la administración del ciclo de vida de la información.

Una vez argumentadas las ventajas y desventajas de las arquitecturas ofrecidas por Elastic, se diseñará una arquitectura que comprenda las características evaluadas para el entorno de producción y se seleccionará la arquitectura con la mejor relación fidelidad/recursos para el entorno de pruebas.

5.2.1. Aspectos arquitectónicos de Elastic

A lo largo de esta sección se presentarán los diseños arquitectónicos básicos que ofrece Elastic y algunas características y mejores prácticas del SIEM. Además, se hará hincapié en el entorno de pruebas.

En la sección 5.2.1.1. Arquitecturas simples, se explicarán las posibles combinaciones arquitectónicas con los componentes básicos del *ELK Stack* y otros más avanzados como servidores de colas y agentes de Elastic AMP.

En la sección 5.2.1.2. Monitorización y replicación, se muestran buenas prácticas recomendadas por Elastic y la mejor forma de implementarlas.

En la sección 5.2.1.3. Administración de los datos, cifrado y autenticación, se explican algunas características que Elastic permite para cumplir con la ley y mejorar la seguridad desde el diseño.

En la sección 5.2.1.4. Arquitecturas del entorno de pruebas, se ilustran las arquitecturas que se han tenido en cuenta a la hora de implementar el SIEM del entorno de pruebas.

5.2.1.1. Arquitecturas simples

Un SIEM basado en el modelo *ELK Stack* de Elastic contiene uno o más nodos ejecutando Elasticsearch, Logstash y Kibana, y una o varias fuentes de datos ejecutando Beats. Aunque el componente Logstash añade procesamiento y robustez, no es absolutamente necesario para realizar las tareas básicas de un SOC, por lo que la arquitectura más simple que se puede diseñar

se basa en los componentes de Elasticsearch, Kibana y Beats, tal y como se muestra en la Figura 13.

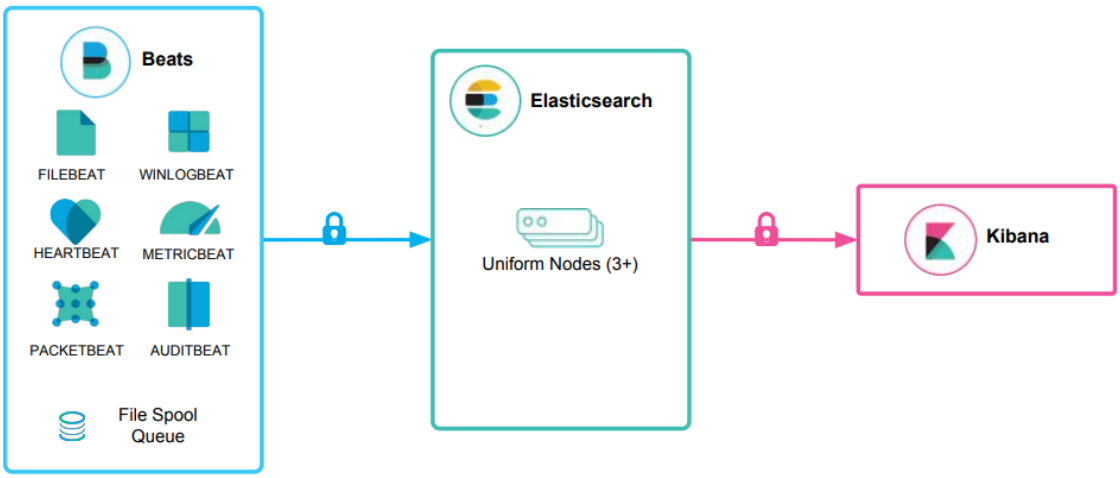

Figura 13.- Arquitectura básica de un SOC basado en Elastic.

Para mejorar el rendimiento de la plataforma, facilitar su administración y cumplir con algunos servicios como el almacenamiento forense, se puede integrar uno o más componentes Logstash que permitan almacenar los registros en caché en caso de fallo de conexión de red o de todas las aplicaciones Elasticsearch y enriquecer y procesar los registros antes de almacenarlos. Esta nueva arquitectura se muestra gráficamente en la Figura 14, donde además de añadir Logstash, se segregan los roles de Elasticsearch entre varios nodos.

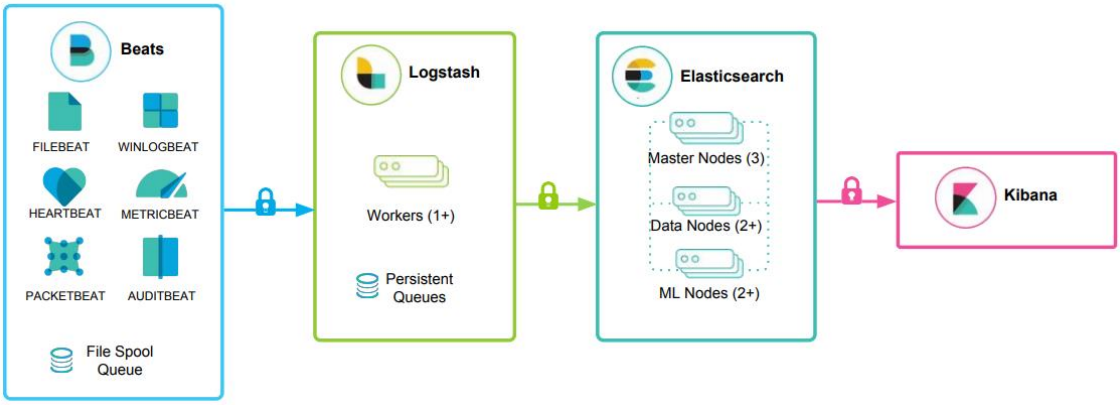

Figura 14.- Arquitectura avanzada de un SOC basado en Elastic.

Otros productos que sería interesante investigar y analizar su aplicación en un entorno de producción serían colas de mensajería como Kafka o Redis (Figura 15), monitorización del rendimiento de aplicaciones con Elastic APM (Figura 16) y administración de fuentes de datos con Fleet (33).

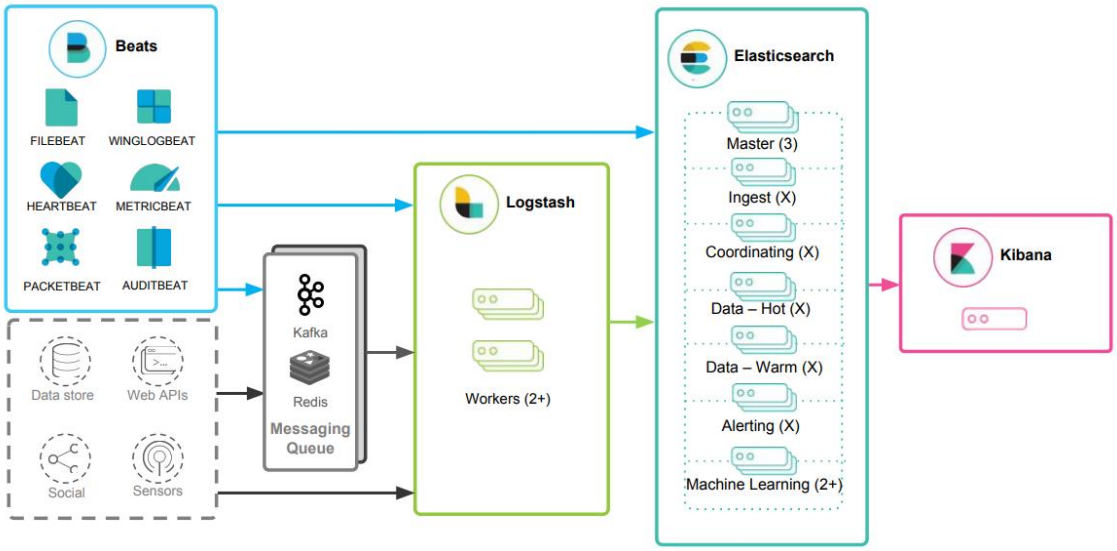

Figura 15.- Arquitectura avanzada de un SOC basado en Elastic con colas de mensajería.

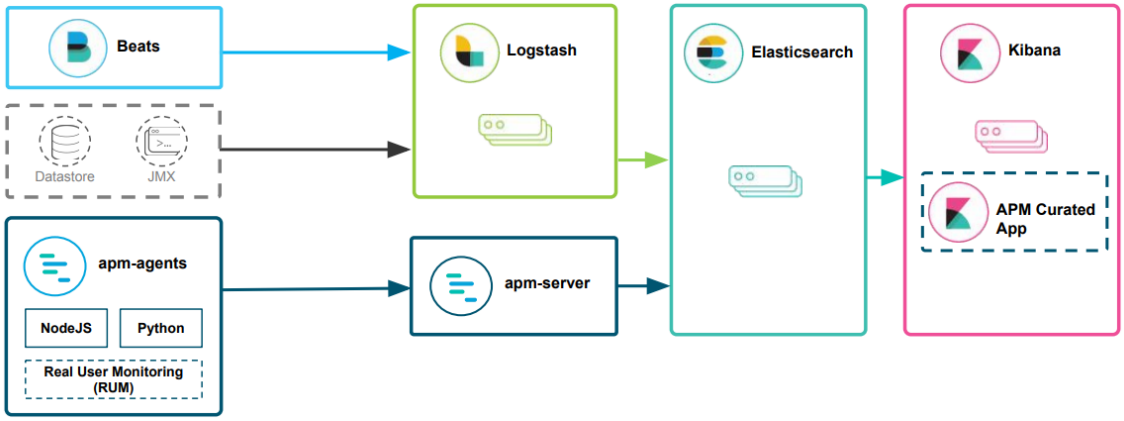

Figura 16.- Arquitectura avanzada de un SOC basado en Elastic con Elastic APM.

5.2.1.2. Monitorización y replicación

En lo que se refiere a la arquitectura del clúster Elasticsearch, se puede crear más de un clúster para mejorar la monitorización y la replicación.

Por un lado, para monitorizar los componentes propios de la plataforma ELK, es decir, Elasticsearch, Kibana y Beats, y otros productos de Elastic como APM o Enterprise Search, se puede utilizar un único clúster que comprenda los servicios de producción y monitorización a la vez. Sin embargo, esta arquitectura no es recomendada puesto que, en caso de error en el entorno de producción, no se podría acceder al entorno de monitorización para investigar dicho error, ya que ambos comparten clúster. Por ello, una buena práctica es implementar dos clústers, uno de producción y uno de monitorización, Figura 17 (34).

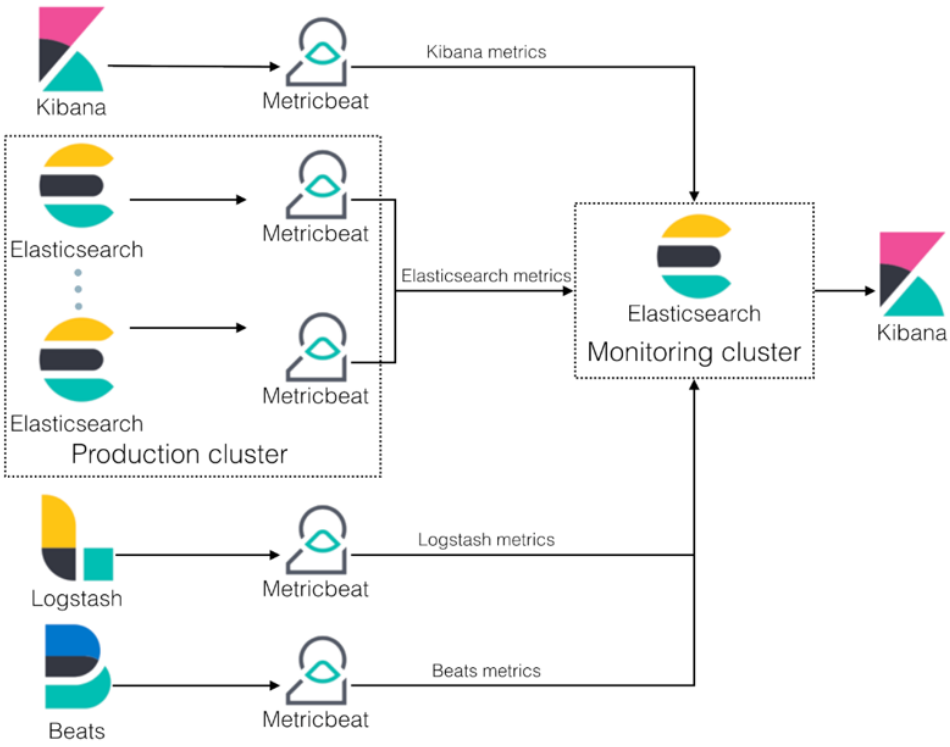

Figura 17.- Segregación de clústeres de producción y monitorización.

Normalmente, aunque se implemente la monitorización en un clúster separado, ambos clústeres suelen coexistir en la misma localización física, por lo que en caso de accidente natural u otros tipos de desastres físicos, ambos clústeres se verían afectados. Para evitar esto, los ingenieros de Elastic recomiendan implementar el clúster de monitorización en la nube, Figura 18 (33).

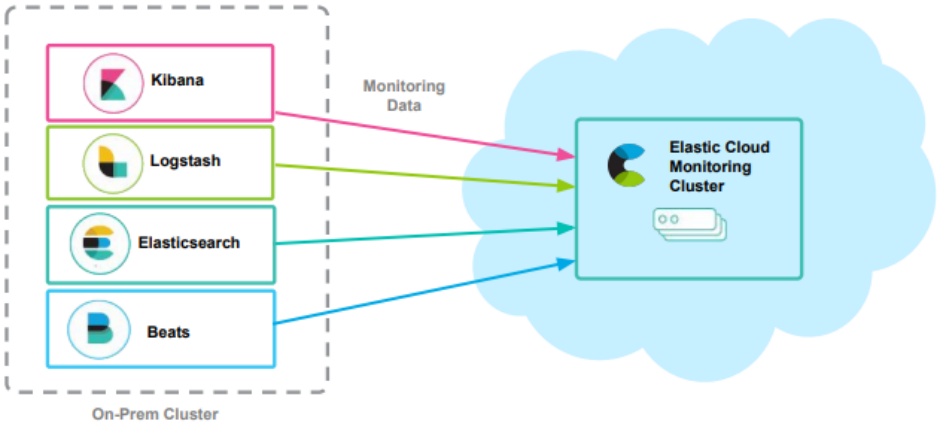

Figura 18.- Implementación del clúster de monitorización en la nube.

Por otro lado, al añadir replicación de clústeres se puede mejorar en seguridad y rendimiento. Es verdad que un clúster de Elasticsearch posee replicación de shards por defecto en sus nodos. Sin embargo, en caso de que fallen todos los nodos del clúster, la información almacenada se perdería y se pararía la operatividad del SOC. Por ello, si se implementan dos clústeres réplica separados físicamente, en caso de que uno de ellos falle, tanto la operatividad del SOC como la integridad de los datos permanecerán garantizados, Figura 19.

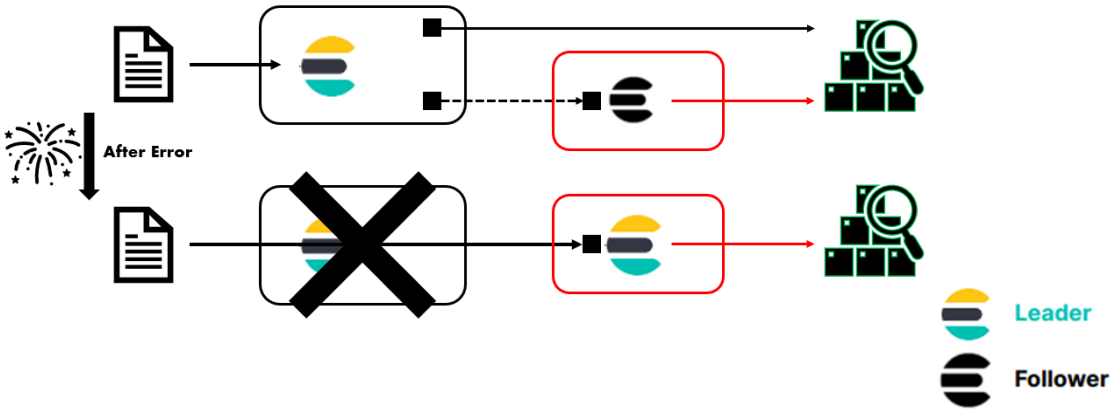

Figura 19.- Replicación de clústeres por seguridad

Para mejorar el rendimiento del clúster mediante la replicación, hay que analizar primero el tipo de infraestructura donde se desea implementar el clúster. Para aquellas organizaciones internacionales con oficinas por todo el mundo, sería interesante disponer de un clúster réplica cerca de cada sucursal para mejorar las operaciones de lectura reduciendo la latencia y el tiempo de respuesta, Figura 20.

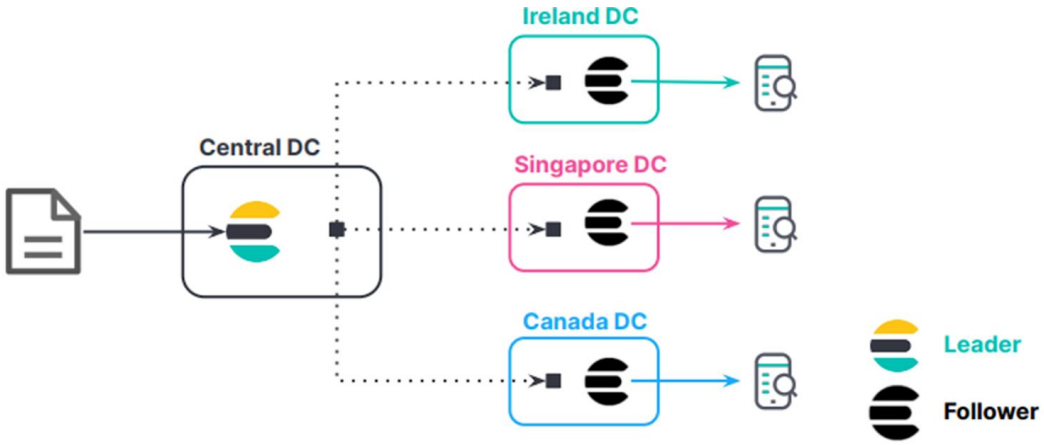

Figura 20.- Replicación de clústeres para organizaciones internacionales.

En cambio, si se dispone de una organización con un SOC centralizado y clientes internacionales, la arquitectura más interesante sería aquella que permitiera realizar operaciones de escritura desde distintos puntos geográficos y operaciones de lectura desde una central de forma eficiente, Figura 21.

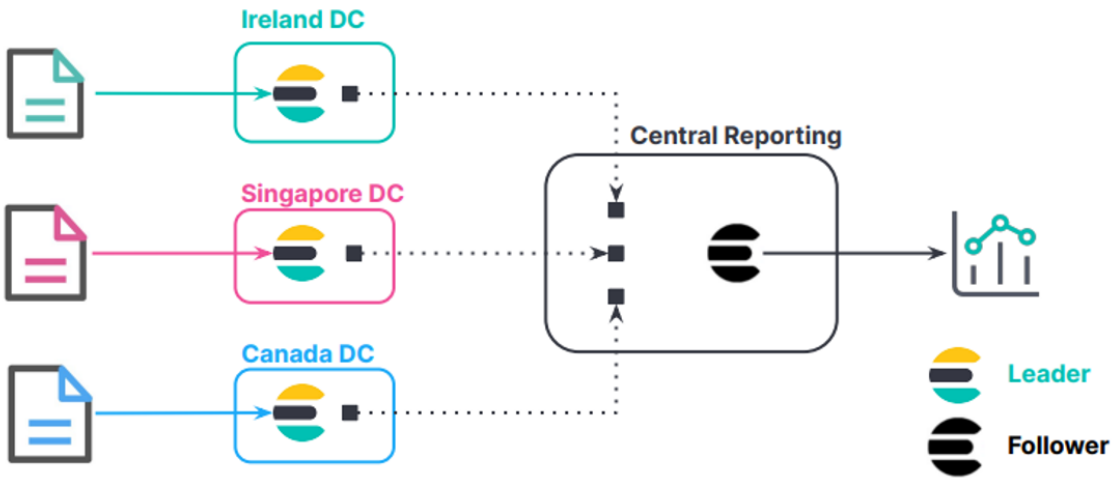

Figura 21.- Replicación de clústeres para organizaciones centrales.

5.2.1.3. Administración de los datos, cifrado y autenticación

La administración de los datos depende directamente del contrato que el SOC tenga con el cliente, de las leyes del país de origen y destino de los datos y de los estándares a los que se adhiera. Existen muchos factores, pero normalmente se suelen mantener los datos entre tres, seis y doce meses para posibles investigaciones forenses. Sin embargo, mantener tantos datos cuesta muchos recursos y algunos de ellos no se suelen utilizar a menos que ocurra algún evento extraño que requiera de análisis forense.

Por ello, Elastic ofrece cinco fases de datos explicadas en el punto 3.3.1. Elasticsearch. Como ya se ha explicado anteriormente, los datos más recientes se almacenan en nodos Elasticsearch de datos de la fase *hot* para mejorar la latencia en las búsquedas más usuales de los analistas. Tres días después, son transferidos a nodos de datos de tipo *warm* hasta que termina la semana, puesto que las búsquedas de datos en la última semana son también muy utilizadas por los analistas. Cuando los registros llevan en el sistema de uno a tres meses, son transferidos a la fase *cold*, donde las búsquedas son mucho más lentas y los nodos necesitan poca CPU y mucho disco. Finalmente, los datos terminan almacenándose en la fase *frozen*, donde quedan comprimidos. Esta fase tiene una gran latencia en las búsquedas y es utilizada para cumplir con la ley de mantener datos durante cierto tiempo y para investigaciones forenses. Los tiempos mencionados son los que se utilizarán en el entorno de pruebas y pueden ser modificados de forma arbitraria según las necesidades de los clientes del SOC. En la Figura 22 se puede observar el proceso de una forma más visual.

Figura 22.- Administración de los datos por fases.

A partir de la versión 6.8 y 7.1, Elastic lanzó varias funcionalidades orientadas a la seguridad como la posibilidad de cifrar el tráfico usando SSL y TLS, crear y administrar usuarios y roles que protegen el acceso a nivel de índice y clúster, y asegurar Kibana por completo (35).

Elasticsearch posee una herramienta conocida como elasticsearch-certutil que permite la generación de certificados. Además, al instalar Elasticsearch se crea automáticamente un certificado CA mediante el cual se pueden firmar el resto de los certificados. Si la empresa donde se instala el SIEM no posee su propia estructura de certificados, el uso de esta herramienta es recomendado (36).

Elasticsearch permite integrar la autenticación de usuarios con distintas tecnologías como LDAP, Active Directory, PKI, Kerberos, SAML, etc. El SIEM es una herramienta crítica en caso de fallos, por lo que, si la autenticación de todos los usuarios del SIEM está integrada con otra tecnología, por ejemplo, Active Directory (AD), entonces en caso de fallo de AD se perderá el acceso al SIEM también. Por ello, se recomienda mantener uno o dos usuarios nativos de la herramienta para casos de emergencia y desastre (37).

5.2.1.4. Arquitecturas del entorno de pruebas

El entorno de pruebas debe simular un entorno de producción evitando el mayor consumo de recursos posible, tanto en la implementación, como en el mantenimiento. En este proyecto se pretende crear un entorno básico, con Elasticsearch, Logstash y Kibana instalados y una o varias fuentes de datos de los dos sistemas operativos principales Windows y Linux.

El diseño más sencillo que se ha analizado para este entorno es un único servidor con Elasticsearch, Logstash y Kibana instalados localmente que reciba datos de uno o varios sistemas Linux y Windows a través de una conexión segura, Figura 23. Este diseño requiere pocos recursos de memoria y disco, puesto que los principales componentes del SIEM se encuentran instalados en una única máquina. Sin embargo, dista mucho de una arquitectura avanzada típicamente implementada en producción y tiene un rendimiento muy limitado.

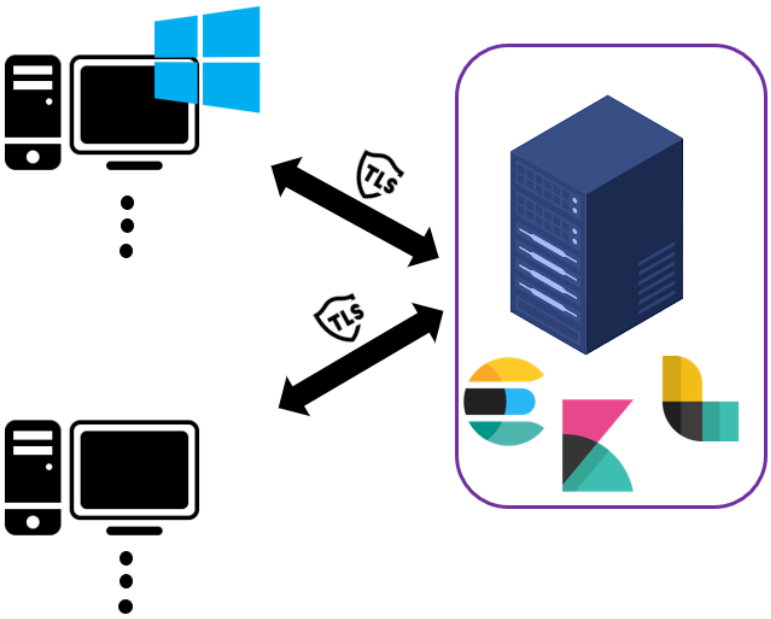

Figura 23.- Arquitectura nivel básico para un entorno de pruebas.

La siguiente arquitectura para analizar posee dos servidores, uno corriendo Logstash y otro corriendo Elasticsearch y Kibana. Además, se reciben registros de sistemas Windows y Linux a través de una conexión segura, Figura 24. Este entorno se acerca a un entorno de producción para empresas pequeñas, cuya monitorización es dedicada a la propia empresa, con pocos recursos y pocos activos que monitorizar.

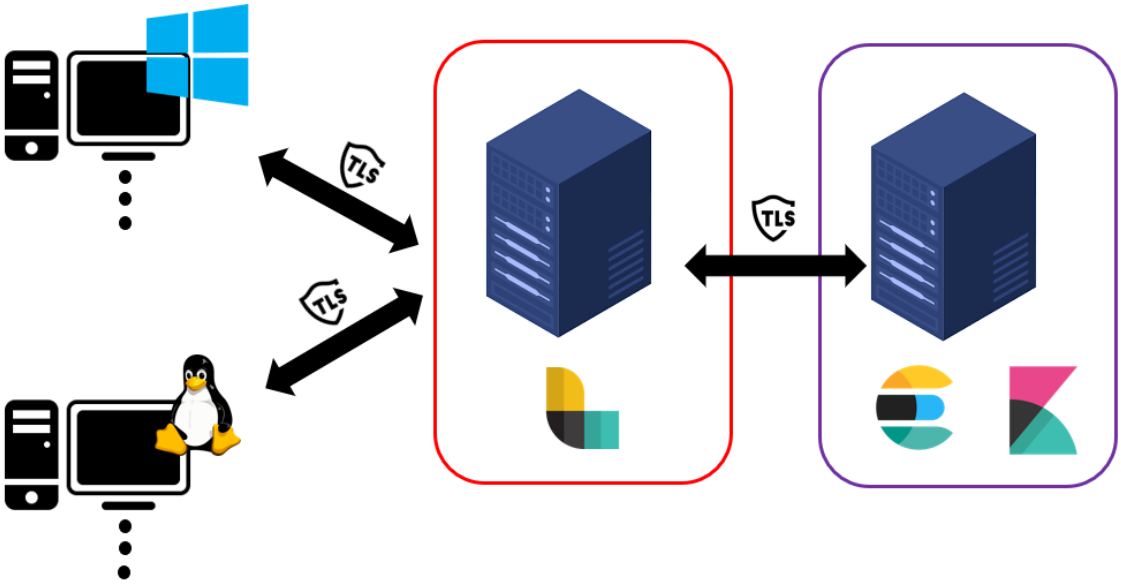

Figura 24.- Arquitectura nivel medio-básico para un entorno de pruebas.

A continuación, si la empresa es pequeña, pero desea acceder al SIEM desde el exterior, entonces una buena práctica es separar los servicios de Elasticsearch y Kibana por seguridad, Figura 25. De esta forma, solo el servidor de Kibana quedará expuesto al exterior. A parte de la ventaja mencionada, esta arquitectura tiene los mismos aspectos que la anterior.

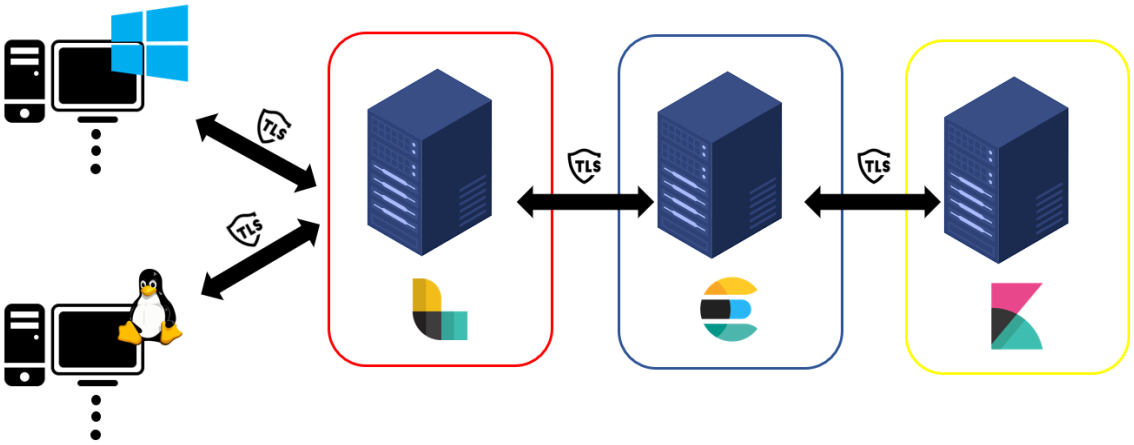

Figura 25.- Arquitectura nivel medio-avanzado para un entorno de pruebas.

Finalmente, si se desea tener una arquitectura similar a la de un entorno de producción real de una empresa mediana o de un proveedor de servicios de seguridad administrados (MSSP), se deben añadir varios nodos de Elasticsearch, Logstash y Kibana, para crear un clúster que pueda almacenar muchos datos, balancear la carga de procesamiento y enriquecimiento de los datos y aumentar el número de conexiones y búsquedas simultáneas por parte de los analistas. Esta arquitectura consume muchos recursos y es la más avanzada que se estudiará para un entorno de pruebas en este TFG, se puede observar su diseño en la Figura 26.

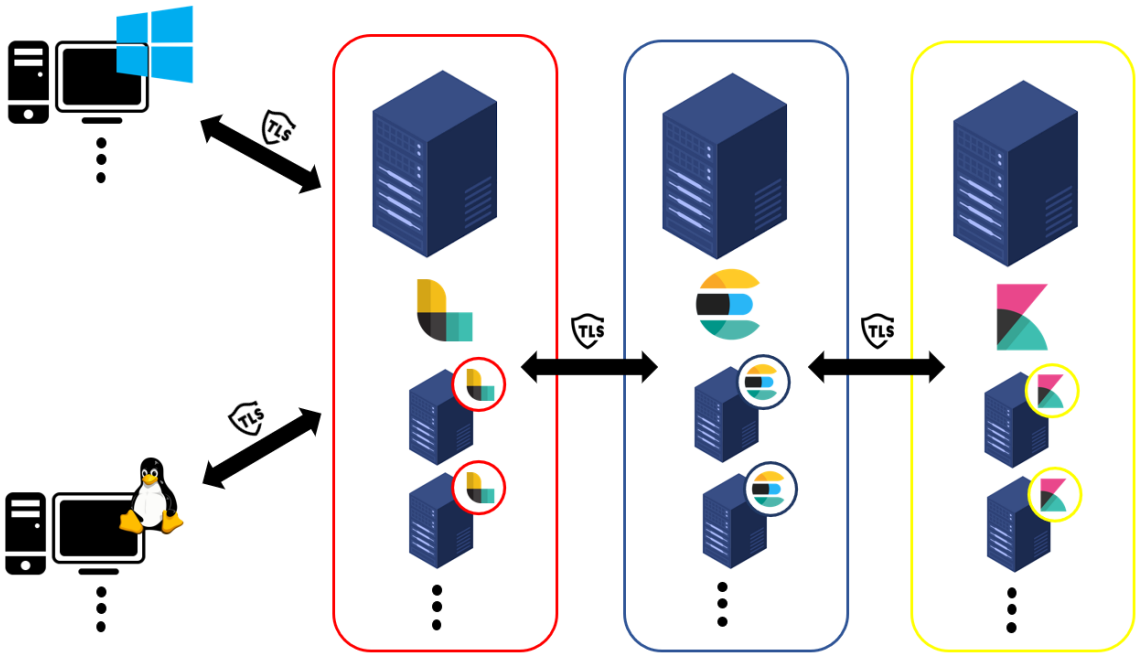

Figura 26.- Arquitectura nivel avanzado para un entorno de pruebas.

5.2.2. Arquitectura elegida

Tras evaluar las posibles arquitecturas, características y mejores prácticas que ofrece Elastic, se diseñarán las arquitecturas finales y sus requisitos de hardware para el entorno de producción y pruebas del proyecto.

En la sección 5.2.2.1. Entorno de producción, se pondrán en práctica las características estudiadas anteriormente para implementar un SOC seguro, robusto y confiable en un proceso de evolución iterativa de la infraestructura.

En la sección 5.2.2.2. Entorno de pruebas, se escogerá una de las arquitecturas evaluadas anteriormente y se especificarán los requisitos hardware, tipos de red y sistemas operativos que se implementarán.

5.2.2.1. Entorno de producción

Elastic no proporciona métricas para conocer cuánta información se puede almacenar en Elasticsearch sin perder rendimiento o cuántos registros puede procesar Logstash sin hacer uso del mecanismo de *backoff* para ralentizar el flujo de datos. En definitiva, Elastic no ofrece ninguna información sobre cuántos nodos debería desplegar una empresa, dada una cantidad de registros por unidad de tiempo. Además, muchas empresas no conocen de primera mano cuanta información va a manejar el SIEM y mucho menos en la fase de diseño, cuando los casos de uso están en proceso de llevarse a cabo, ya que no se conoce qué fuentes de datos se van a utilizar. Por ello, se ha decidido diseñar dos arquitecturas para el entorno de producción.

La primera arquitectura, Figura 27, es una arquitectura simple, válida para procesar las fuentes de datos de los primeros casos de uso. La información parte desde la red de un cliente y llega a los firewalls de la red del SIEM. A continuación, los datos filtrados pasan a los dos servidores Logstash, donde se procesarán y enriquecerán los registros, y se reenviarán al clúster Elasticsearch, compuesto por tres nodos que almacenan datos en las fases *hot* y *content* y un nodo con datos en la fase *warm*. Además, los nodos Elasticsearch de las fases *hot* y *content* correrán también instancias de Kibana a las cuales se conectarán los analistas desde la red del SOC.

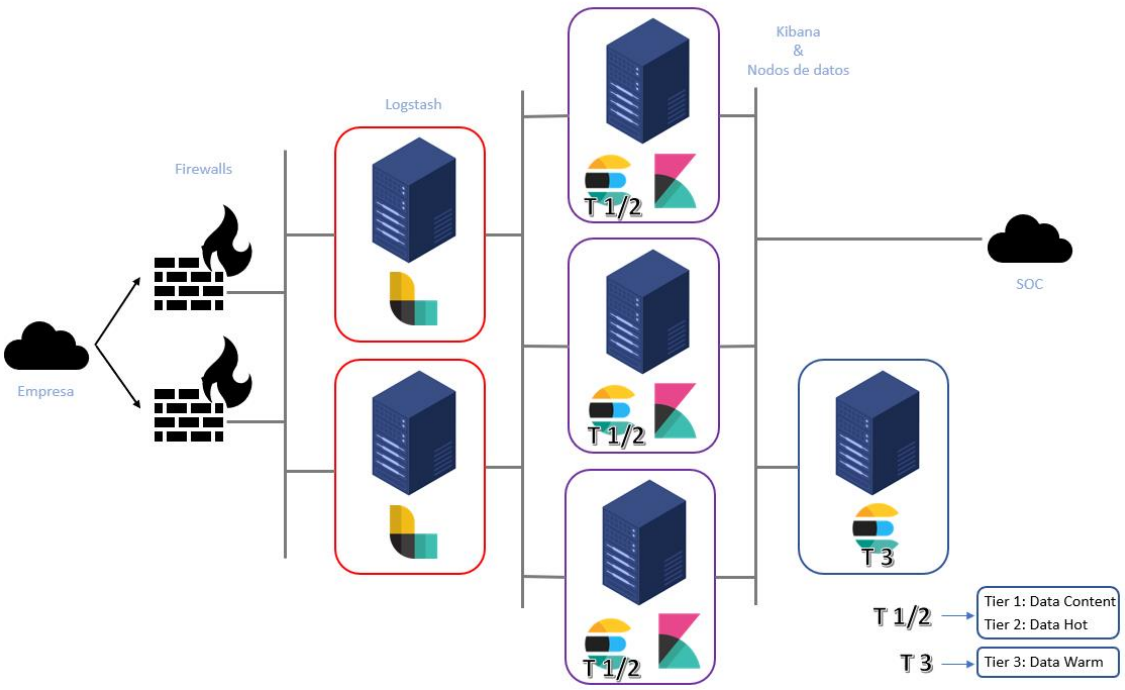

Figura 27.- Arquitectura SIEM básica elegida.

Como Elastic no proporciona información sobre los requisitos mínimos y recomendados de hardware para la instalación y despliegue de sus componentes, se ha decidido empezar con unos recursos básicos y aumentarlos o disminuirlos según el análisis del comportamiento del tráfico y los datos que se obtenga de esta arquitectura:

- Requisitos de servidores Logstash
    - CPU: 8 CPU
    - Memoria: 32 GB
    - Disco: 64 GB
- Requisitos de servidores Elasticsearch Tier 1 y 2
    - CPU: 16 CPU
    - Memoria: 64 GB
    - Disco: 512 GB
- Requisitos de servidores Elasticsearch Tier 3
    - CPU: 16 CPU
    - Memoria: 64 GB
    - Disco: 5 TB

Esta arquitectura posee muchas carencias: no posee nodos Elasticsearch que almacenen datos en las fases *cold* y *frozen*, por lo que los datos serán almacenados durante una semana antes de ser eliminados del sistema. Además, la monitorización del clúster se lleva a cabo en el propio clúster, por lo que, en caso de fallo del clúster, no se podrá investigar el estado de los nodos desde Kibana. También carece de réplicas, lo que afecta a la seguridad en caso de desastre natural y al rendimiento en caso de que la red de los clientes o del SOC esté muy alejada geográficamente de la red del SIEM. Por último, en caso de fallo de conexión dentro de la red del SIEM, los registros quedarán almacenados en la caché de Logstash. Sin embargo, si el fallo se produce entre la red del cliente y la red del SIEM, como es el caso más probable, los registros no llegarían a Logstash y se perderían, lo cual es inaceptable.

A pesar de que la arquitectura básica mostrada en la Figura 27 no es conveniente para un entorno de producción real, sí que es válida para empezar y reunir información crítica sin consumir una cantidad ingente de recursos, ya que con ella se pueden obtener datos reales del entorno, conociendo así cuántos nodos de cada tipo harán falta por cada caso de uso. Además, la monitorización y replicación de los recursos es un aspecto fundamental para un SOC maduro, pero no crítico cuando este está empezando, por lo que, si se desea un crecimiento de la red, los servicios y los recursos del SOC, esta arquitectura ofrece una buena base, ya que hay que tener en cuenta que Elastic permite el crecimiento de forma iterativa, es decir, se puede diseñar una arquitectura mucho más avanzada con esta como base, sin tener que empezar de cero.

Con el paso del tiempo, la arquitectura básica del SIEM debe evolucionar a una arquitectura mucho más compleja, robusta y segura, en este TFG se ha diseñado la arquitectura de la Figura 28. Esta arquitectura se ha diseñado siguiendo el modelo de un MSSP que ofrece sus servicios siguiendo una filosofía internacional, es decir, utiliza centros de datos repartidos geográficamente para mejorar la conexión y la latencia de las fuentes de datos de los clientes con la red del SIEM. Además, en caso de que esta conexión se pierda, los datos quedarían almacenados en servidores de colas implementados con tecnologías recomendadas por Elastic como Redis o Kafka, desplegados en la red de los clientes.

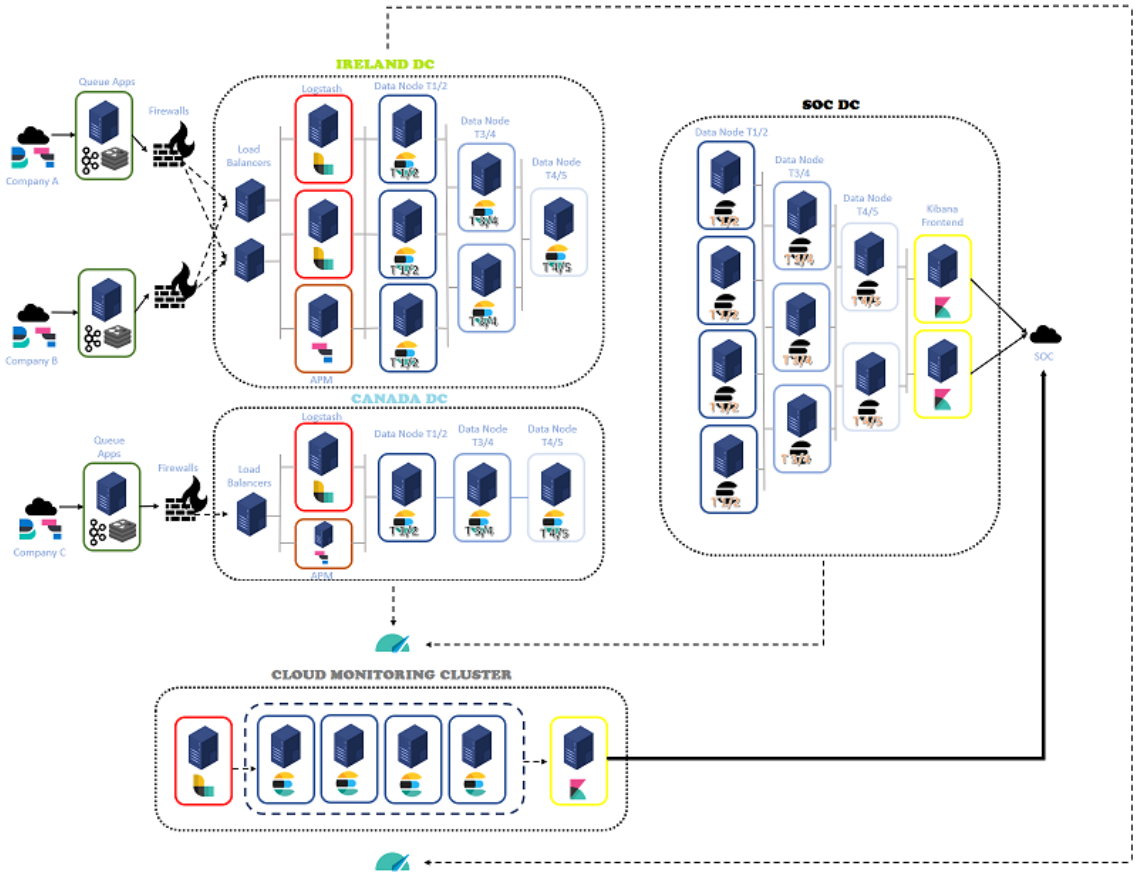

Figura 28.- Arquitectura SIEM compleja elegida.

A pesar de tener varios centros de datos distribuidos geográficamente de forma estratégica para los clientes, la latencia experimentada por los analistas del SOC no se verá afectada puesto que se implementará un clúster replica cerca de la red del SOC para mejorar la seguridad de los datos y la eficiencia de los analistas, siguiendo el modelo de la Figura 21. Además, se ha decidido establecer un clúster independiente para la monitorización desplegado en la nube, tal y como recomienda Elastic en la Figura 18. También se cuenta con servidores Elastic APM para mejorar la monitorización de las aplicaciones de los clientes y nodos Elasticsearch de todas las fases en cada uno de los clústeres, excepto en el de monitorización, para cumplir con todo tipo de contratos y leyes que obliguen una permanencia de datos superior a meses y/o años.

Los requisitos hardware quedarán claros una vez se realice el estudio del comportamiento del tráfico y los datos en la arquitectura básica. Entonces, a partir de esa base se diseñará un documento con las especificaciones recomendadas para el entorno de producción avanzado.

En cuanto a la seguridad, en ambas implementaciones, la básica y la compleja, todos los datos recibidos desde los clientes pasan a través de un firewall antes de entrar en la red del SIEM y todas las conexiones representadas en el diagrama poseen cifrado extremo a extremo con certificados. En caso de que el MSSP posea una estructura propia de certificados se debe evitar, por seguridad, almacenar el certificado raíz en los servidores de colas y fuentes de datos que hay en la red de los clientes. Además, la autenticación en el SIEM se realizará a través del sistema principal del MSSP, como Active Directory o LDAP y se creará uno o dos usuarios de

emergencia nativos en el SIEM con privilegios de administrador[6]. Finalmente, en la arquitectura compleja se han desplegado balanceadores de carga que permiten distribuir el tráfico entre los servidores Logstash para mejorar el rendimiento de estos sistemas.

5.2.2.2. Entorno de pruebas

Para el entorno de pruebas se evaluará la relación afinidad-recursos, es decir, cuanto más similar es el diseño del entorno de pruebas al de producción mayor valor tendrá el diseño, pero cuanto más recursos consuma menor será la probabilidad de implementarlo virtualmente, como se pretende en este TFG. Por ello, la arquitectura analizada más similar al entorno de producción simple de la Figura 27 es el diagrama de la Figura 24, el cual ofrece también un consumo de recursos asequible para ser implementado virtualmente en un ordenador portátil con 16 GB de RAM, 458 GB de disco y un procesador Intel Core i7-10750H a 2.60 GHz, a través de Virtual Box.

A continuación, se muestran las especificaciones utilizadas en cada máquina:

- Servidor Elastic + Kibana:
  - CPU: 1 CPU
  - Memoria: 6 GB
  - Disco: 60 GB
  - Sistema Operativo: Linux Ubuntu
- Servidor Logstash:
  - CPU: 1 CPU
  - Memoria: 2.56 GB
  - Disco: 10 GB
  - Sistema Operativo: Linux Lubuntu
- Sistema Linux:
  - CPU: 1 CPU
  - Memoria: 1.02 GB
  - Disco: 10 GB
  - Sistema Operativo: Linux Lubuntu
- Sistema Windows: Instalado en el host

El diagrama de la arquitectura final del entorno de pruebas se puede observar en la Figura 29, donde aparece también el tipo de red, los sistemas operativos utilizados y las direcciones IP de cada sistema. En lo que se refiere a las redes, se utilizará una red NAT para todas las máquinas virtuales y una red Host-Only entre el host Windows y el servidor Logstash, simulando así que las fuentes de datos Windows y Linux vienen de clientes distintos.

[6] Los privilegios de administrador más altos existentes en la herramienta Elasticsearch se otorgan a través del rol de superusuario.

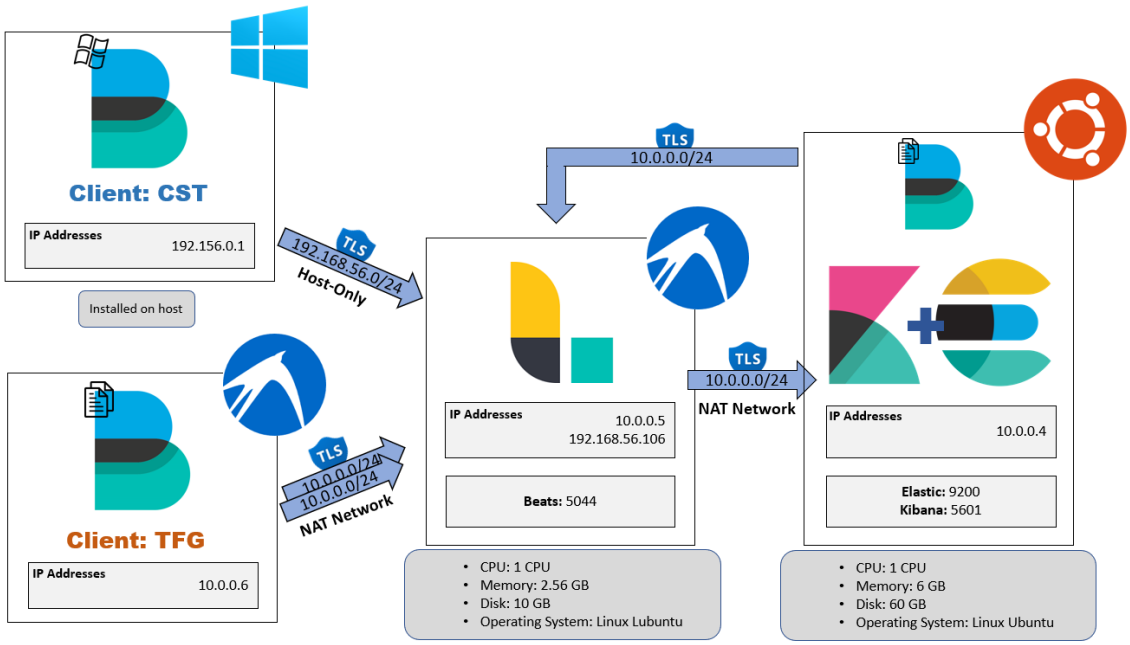

Figura 29.- Arquitectura SIEM del entorno de pruebas.

5.3. Convención de nombres

Cuando se implementa un SOC desde cero se debe acordar el nombre de cada cosa en cada momento. Por ejemplo, cuando sucede un evento de seguridad y salta un aviso en el SIEM, Elastic Security lo llama alerta, pero otros SIEM lo llaman alarma o incidente de seguridad. Algunos SOC diferencian entre aquellas alertas que son Falsos Positivos, aquellas que son Verdaderos Positivos y aquellas que se necesita más información del equipo de respuesta a incidentes para determinar su estado. Cada uno de esos tipos de alarmas debe recibir un nombre para una correcta comunicación entre el equipo del SOC. Estos conceptos dependen del flujo de trabajo y los procesos de cada SOC, por lo que en este TFG tan solo se recomienda estandarizar los conceptos relativos al SOC que se desea implementar.

Por otra parte, durante la fase de diseño los arquitectos y analistas deben de trabajar conjuntamente para conseguir toda la información necesaria en los registros y clasificarlos de forma correcta. Por ejemplo, en muchos países la mezcla de datos de dos clientes distintos es ilegal, tampoco les interesa a los analistas estudiar los registros de un cliente cuando la alerta pertenece a un caso de uso de otro cliente, por lo que la división de los registros por clientes es necesaria. Sin embargo, el nombre del cliente no aparece en los registros de las fuentes de datos por defecto. Otro campo interesante podría ser la geolocalización de las direcciones IP fuente y destino de un registro, o el segmento de red donde pertenece la fuente de datos, o el tipo de fuente de datos.

Cada uno de los campos anteriores debe tener un nombre estándar tanto en el campo como en el valor, por ejemplo, la implementación del entorno de pruebas que se realizará en este TFG seguirá los siguientes patrones:

- Campo: client_id | Valor: tfg (un código de tres letras que represente al cliente).
- Campo: geoIP | Valor: es (un código de dos letras que represente al país).

- Campo: network_type | Valor: nat network (El nombre del segmento de red donde pertenece la fuente de datos).

Estos son algunos campos y valores que se añadirán a los registros para mejorar la calidad de las investigaciones. Sin embargo, hay muchos registros que ya poseen los valores en los registros, pero estos no están bien procesados en campos, por ejemplo, registros que proceden de syslog. En esos casos habrá que estandarizar los nombres de los campos y procesar los registros a través de Logstash. Elastic ya posee una especificación de nombres llamada *Elastic Common Schema* (ECS) que ofrece una guía de mejores prácticas para nombrar los campos de los registros (38).

La convención de nombres no solo abarca conceptos y registros, durante la implementación también habrá que definir los nombres de las reglas y durante las investigaciones se hará uso de los casos, ambas funciones necesitan un estándar. En el caso del entorno de pruebas a implementar se ha decidido dividir las reglas por clientes, para conocer qué cliente se está investigando cuando salta una regla y utilizar un contador en los casos que indique cuántos casos de cada regla se han realizado. Por ejemplo, una regla para detectar ataques de fuerza bruta en una empresa llamada Trabajo Final de Grado con código TFG sería: TFG - Ataque de Fuerza Bruta Detectado, y si salta una alarma de esa regla por primera vez, el caso a investigar se llamaría: TFG - Ataque de Fuerza Bruta Detectado 1.

Finalmente, y siendo una de las convenciones de nombres más importantes en Elastic, se debe definir la estructura de nombres de un índice y en caso de que se vaya a utilizar, de un *data stream*. En este TFG se utilizarán los *data streams* para automatizar los procesos con los índices y mejorar la eficiencia del SIEM, esta práctica está muy recomendada.

Los *data streams* tienen una estructura por defecto con tres campos separados por guiones, a estos tres campos Elastic los denomina tipo, conjunto de datos y espacio de nombres. El tipo es el campo más amplio, describe el conjunto de los registros, el conjunto de datos es un poco más específico y describe el subconjunto de datos al que pertenecen los registros que se almacenan en el *data stream* y el espacio de nombres incluye detalles específicos de la infraestructura de cada cliente.

Por ello, en la implementación del entorno de pruebas y el de producción, se recomienda utilizar el nombre del cliente como tipo, la tecnología a la que pertenecen los registros como conjunto de datos y el tipo de sistema que envía dichos registros como espacio de nombres. Entonces, los registros procedentes de una fuente de datos del cliente Trabajo Fin de Grado, que envía registros de eventos de seguridad de Windows desde un controlador de dominio Active Directory, se almacenarán en el *data stream* tfg-windows.event.security-dc.

5.4. Casos de uso

Hay dos formas de orientar la creación de casos de uso, la más habitual se basa en la creación de menús con listas de casos de uso importantes para ciertos sectores o para cumplir con ciertas normas y leyes o incluso orientados a combatir ataques desde distintos frameworks como Mitre Att&ck. Cuando se aborda la integración de un cliente nuevo, este podrá decidir aquel menú de casos de uso que le interese. Es una aproximación más barata puesto que permite la reutilización de las reglas implementadas para cada caso de uso, pero es genérica, es decir, dos clientes

distintos poseen modelos de negocio, riesgos e infraestructuras distintas y con un menú preconfigurado es difícil solucionar todos los riesgos relativos a un cliente.

La forma alternativa es aquella que se basa en los riesgos del cliente. El MSSP envía a un especialista en riesgos que, mediante varias entrevistas con los directivos del cliente y los arquitectos de red, genera una lista de riesgos específicos para dicho cliente. De los riesgos detectados, se crea una lista de casos de uso y el SOC se encarga de transformar esos casos de uso en reglas. Esta aproximación genera casos de uso a medida para los riesgos de cada cliente, pero es mucho más cara, tanto económicamente como en recursos.

En el caso del entorno de pruebas se crearán reglas siguiendo la primera aproximación, ya que no hay un cliente real que analizar, sin embargo, se creará dos clientes ficticios, cuyos códigos serán TFG y CST, que trabajarán únicamente con máquinas Linux y Windows respectivamente. Dentro de los distintos menús que se pueden crear, se ha decidido monitorizar las técnicas descritas en el apartado 4.1. Problemas de seguridad, obtenidas del framework Mitre Att&ck.

El primer caso de uso a implementar pretende detectar la técnica T1046 *Network Service Discovery*. Según se ha descrito en el análisis del problema, esta técnica se basa en escáneres de puertos y vulnerabilidades. Para detectar escáneres de puertos, basta con tener un firewall como IPtables, en cambio, para detectar escáneres de vulnerabilidades es necesario un firewall de aplicación avanzado o un *endpoint* con firmas de *exploits* conocidos para poder analizar el *payload* de los paquetes que llegan al sistema. Sin embargo, esta tecnología es de pago y no se implementará en el entorno de pruebas, por lo que nos centraremos únicamente en la detección de escáneres de puertos. Una vez se compruebe que IPtables está instalado, se debe crear una regla en el firewall para que genere registros de cada paquete detectado. Esta regla se puede implementar sobre el cliente TFG del laboratorio, puesto que las máquinas Linux poseen IPtables por defecto.

El segundo caso de uso pretende detectar la técnica T1110.001 *Brute Force: Password Guessing*, es decir, un ataque de fuerza bruta. En nuestro caso, monitorizaremos las conexiones SSH para detectar este tipo de ataques contra este servicio. No necesitaremos instalar ningún software adicional puesto que SSH es un servicio por defecto en Linux. Para implementar esta regla se enviarán los registros del servicio SSH del cliente TFG al SIEM.

El tercer caso de uso monitoriza la técnica T1078.003 *Valid Accounts: Local Accounts* que monitorizará el acceso al SIEM fuera del horario laboral. Para implementar esta regla se necesitan los registros de Kibana, ubicado en un archivo dentro del servidor Kibana, por lo que habrá que instalar Filebeat para recibir los registros de dicho archivo.

El cuarto caso de uso, T1136.001 *Create Account: Local Account*, pretende monitorizar la creación de cuentas locales a un sistema. Para lograrlo, se recibirán los eventos de Windows a través de Winlogbeat. Esta regla se implementará únicamente al cliente CST.

El quinto y último caso de uso, T1498.001 *Network Denial of Service: Direct Network Flood*, pretende detectar un ataque de denegación de servicio volumétrico, es decir, con una cantidad masiva de tráfico. Para llevar a cabo esta detección se necesita un firewall como IPtables que registre todo el tráfico dirigido a un sistema, por ello, esta regla se puede implementar desde el cliente TFG.

Todas las reglas que se implementen en la máquina Linux del laboratorio serán reglas para el cliente TFG, mientras que aquellas implementadas para Windows serán para el cliente CST.

Finalmente, para cada regla se debería crear un procedimiento que indique el motivo de la regla, los riesgos, la criticalidad y los pasos a realizar para resolver una alarma de dicha regla.

5.5. Procesos

Los procesos explican paso a paso cómo debe ser ejecutado un servicio en el SOC. Para diseñarlos hay que responder a preguntas como:

- El propósito del proceso.
- La implicación del SOC a lo largo del proceso.
- El tiempo durante el cual el SOC será responsable del proceso.
- Los recursos necesarios para llevar a cabo el procedimiento.
- La obligación de reportar las acciones relacionadas con el proceso.
- El procedimiento de escalado de notificaciones, en caso de necesitar más apoyo.

Algunos de los procesos más típicos en un SOC son:

- Monitorización: Proceso en el que se detalla la vigilancia de los sistemas y redes de una organización. Periodicidad, herramientas, responsables, etc.
- Alertas: Proceso en el que se especifica el procedimiento de notificación en caso de amenaza, problemas o eventos.
- Escalado: Proceso en el que se expone el traspaso de responsabilidades para responder a eventos que necesitan un mayor apoyo.
- Investigación: Procesos que orientan al analista de seguridad a entender el propósito de una alarma, seleccionar su gravedad e investigarla.
- Legal: Procesos relativos al cumplimiento de leyes, estándares y buenas prácticas.
- Reportes: Proceso donde se especifica qué métricas se deben dar en un evento para que sea necesario reportarlo.
- Remedio: Proceso a través del cual se retorna un sistema a un estado operativo y se actualizan las vulnerabilidades para reducir un ataque futuro.

Una vez los procesos maduran, se recomienda crear plantillas para que los nuevos empleados puedan seguir el estándar establecido (1).

6. Desarrollo de la solución

Para la implementación del prototipo o entorno de pruebas siguiendo la arquitectura de la Figura 29, se ha instalado Virtualbox en un host Windows y se han desplegado tres máquinas virtuales siguiendo los requisitos hardware de la arquitectura. A continuación, se ha instalado el sistema operativo correspondiente en las tres máquinas virtuales con los siguientes nombres de host: el servidor que ejecutará Elasticsearch y Kibana se ha denominado elastic-VM, el servidor Logstash se llama logstash-VM y el sistema Linux del cliente TFG se llama linux-VM. Estos nombres servirán para identificar las máquinas cuando se ejecutan los comandos en las capturas de pantalla que aparecerán a lo largo de la implementación. Antes de instalar ninguna aplicación de Elastic, hay que instalar Java en la máquina elastic-VM.

Una vez las máquinas virtuales están listas, es recomendable realizar un *snapshot* para que, en caso de cometer un error en la instalación, sea posible volver a este punto. Los *snapshots* se deben hacer a lo largo de todo el proceso para asegurar una instalación limpia.

En la sección 6.1. Instalación de Elasticsearch y Kibana se instalarán Elasticsearch y Kibana en el servidor elastic-VM siguiendo la documentación de Elastic y se comprobará que ambos se encuentran en funcionamiento.

En la sección 6.2. Instalación y configuración de Logstash se instalará Logstash en la máquina logstash-VM y se configurarán los *pipelines* necesarios para procesar y enriquecer los registros procedentes de las fuentes de datos especificadas en el diseño de casos de uso.

En la sección 6.3. Instalación y configuración de Beats se instalarán las fuentes de datos en las máquinas correspondientes y se configurarán para recibir los registros necesarios. Además, se crearán *data streams* que almacenen los datos y administren los índices automáticamente.

En la sección 6.4. Creación de reglas se especificará la lógica de las reglas, la descripción y el título y se creará una regla para cada caso de uso.

6.1. Instalación de Elasticsearch y Kibana

Todas las instalaciones de los componentes de Elastic, se recomienda hacerlas por “apt” para conservar las mismas rutas que en este TFG y que en la documentación de Elastic. Además, para evitar tener que ejecutar ambas aplicaciones cada vez que se encienda la máquina, se recomienda ejecutarlas como servicio.

Al instalar Elasticsearch por primera vez siguiendo la documentación, aparecerá en la terminal la contraseña del superusuario elastic, Figura 30. Esta contraseña se genera automáticamente de forma aleatoria durante la instalación y es de suma importancia guardarla.

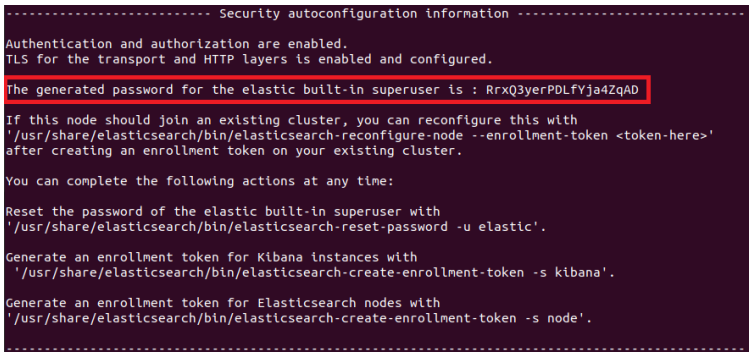

Figura 30.- Contraseña del superusuario elastic.

Para comprobar que Elasticsearch funciona se puede hacer desde la línea de comandos, como en la Figura 31, o directamente desde el navegador buscando la dirección y el puerto de Elasticsearch, en este caso ambos por defecto, localhost:9200. Hay que recordar que Elasticsearch tiene cifrado por defecto, por lo que el uso de HTTPS es obligatorio (39).

Figura 31.- Comprobación de la correcta ejecución de Elasticsearch.

En el caso de Kibana, ya no es necesario añadir los repositorios de Elastic al equipo, ya que este paso se ha hecho previamente con Elasticsearch, por lo que podemos saltarnos algunos pasos de la documentación.

Una vez instalado Kibana, se debe generar un token que será válido durante los próximos 30 minutos, Figura 32, y hay que pegarlo en el formulario que aparece al acceder a Kibana a través del navegador con la dirección y puertos por defecto, localhost:5601 a través de HTTP (40).

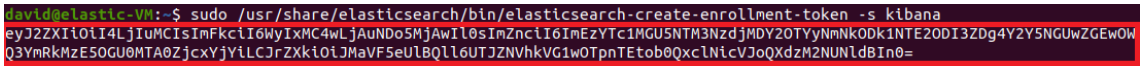

Figura 32.- Generación del token de Kibana.

A continuación, hay que generar un código de verificación siguiendo los pasos indicados en la interfaz web de Kibana e introducirlo en el navegador. En cuanto el proceso de verificación haya terminado, se podrá acceder a Elasticsearch a través de Kibana con el usuario elastic y la contraseña de la Figura 30. Finalmente, hay que cambiar la dirección IP donde escucha Elasticsearch de localhost, que es el valor por defecto, a la dirección de la interfaz de red de la red NAT que comparte con el servidor Logstash, este cambio se puede realizar en el fichero de configuración de Elasticsearch *"/etc/elasticsearch/elasticsearch.yml"* (41).

6.2. Instalación y configuración de Logstash

Para alcanzar el entorno de pruebas propuesto, en esta sección se instalará Logstash en el servidor logstash-VM. La instalación de Logstash es sencilla y solo se tiene que seguir los pasos de la documentación (42).

Logstash funciona a través de ficheros de configuración llamados *pipelines*. Posee un fichero yaml llamado *pipelines.yml*, que debe ser almacenado en la ruta *"/usr/share/logstash/config/"*, y se encarga de administrar todos los *pipelines* que deben ser activados al iniciar la aplicación. En el caso de este prototipo, serán necesarios únicamente dos *pipelines* (43). Uno para procesar los registros de Filebeat para el firewall y los ficheros de autenticación de Linux y Kibana y otro para los registros de la máquina Windows mediante Winlogbeat. Como ambas fuentes de datos son Beats, ambas utilizan el mismo puerto por defecto para enviar los registros, el 5044. Sin embargo, queremos separar ambas fuentes de datos, ya que vienen de clientes distintos y se almacenarán en distintos *data streams*. Para corregir esto hay dos soluciones: cambiar el puerto de envío por defecto de uno de los Beats para que cada *pipeline* escuche en un puerto distinto o segregar los datos dentro del fichero de configuración de *pipelines* para que cada registro sea procesado por el *pipeline* correcto. En la implementación de este TFG se ha decidido seguir la segunda opción, puesto que en un entorno real hay que abrir el menor número de puertos posibles para reducir los vectores de ataque y el riesgo de errores de configuración con los puertos (44). En la Tabla 4, se puede observar el fichero de configuración general de *pipelines*.

/usr/share/logstash/config/pipelines.yml

```
- pipeline.id: beats
  config.string: |
    input {
      beats {
        port => 5044
        ssl => true
        ssl_certificate_authorities                                    =>
["/etc/logstash/shared/http_ca_logstash.crt"]
        ssl_certificate => "/etc/logstash/shared/logstash-cert.crt"
        ssl_key => "/etc/logstash/shared/logstash-cert.key"
        ssl_verify_mode => "force_peer"
      }
    }
    output {
      if "winlogbeat" in [@metadata][beat] {
        pipeline {
          send_to => winlogbeat
        }
      }
      else {
        pipeline {
          send_to => filebeat
        }
      }
    }

- pipeline.id: filebeat
  path.config: "/etc/logstash/pipelines/filebeat.conf"

- pipeline.id: winlogbeat
  path.config: "/etc/logstash/pipelines/winlogbeat.conf"
```

Tabla 4.- Fichero de configuración de *pipeline*s.

En la tabla 4 se puede apreciar el mecanismo utilizado para segregar los datos. En este caso, como solo hay dos fuentes de datos distintas, se puede utilizar el nombre del beat para segregarlos. Sin embargo, en un entorno real se espera que cada cliente posea todos los Beats,

por lo que segregar por Beats generaría un conflicto que debería ser resuelto en los propios *pipelines*, segregando por el campo cliente. Una vez segregados los registros, estos se envían al *pipeline* correspondiente.

Para que funcione, la conexión entre Elasticsearch y Logstash debe estar cifrada, ya que como se ha mencionado anteriormente, solo se puede acceder a Elasticsearch a través de HTTPS, por defecto. Por ello, es necesario compartir el certificado CA que Elasticsearch creó automáticamente durante la instalación, llamado http_ca.crt, el cual está almacenado en la ruta *"/etc/elasticsearch/certs/"* (45). Para cifrar las conexiones de Logstash con las fuentes de datos, se debe crear un certificado CA para Logstash y con él firmar dos claves públicas con sus correspondientes claves privadas (36), una para Logstash que actúa como servidor y otra para los Beats, que actúan como clientes (46). En el archivo de la Tabla 4 aparece el certificado CA y la clave pública y privada de Logstash. En este TFG se creará una clave pública y privada por cada fuente de datos, es decir, una para Filebeat y otra para Winlogbeat a través de la herramienta elasticsearch-certutil.

Una vez los datos de los Beats lleguen a Logstash por el puerto 5044 y sean segregados a través del fichero de configuración de *pipelines*, serán enviados al *pipeline* correcto. En este *pipeline*, se añadirán los campos deseados, decididos en la convención de nombres y se enviarán al clúster Elasticsearch (47). En la Tabla 5 y la Tabla 6, se muestran ambos *pipelines*, *winlogbeat.conf* y *filebeat.conf*, respectivamente.

/etc/logstash/pipelines/winlogbeat.conf

```
input {
    pipeline{
        address => winlogbeat
    }
}
filter {
     mutate {
        add_field => { "client_id" => "cst" }
        add_field => { "geo_IP" => "uk" }
        add_field => { "network_type" => "Host-Only" }
     }
}
output {
  if[winlog][channel] == "Security"{
    elasticsearch {
        hosts => "https://10.0.0.4:9200"
        index => "%{[client_id]}-windows.event.security-default"
        user => "elastic"
        password => "RrxQ3yerPDLfYja4ZqAD"
        ssl => "true"
        cacert => "/etc/logstash/shared/http_ca_elasticsearch.crt"
        action => "create"
    }
  } else if [winlog][channel] == "Application" {
    elasticsearch {
        hosts => "https://10.0.0.4:9200"
        index => "%{[client_id]}-windows.event.application-default"
        user => "elastic"
        password => "RrxQ3yerPDLfYja4ZqAD"
        ssl => "true"
        cacert => "/etc/logstash/shared/http_ca_elasticsearch.crt"
        action => "create"
    }

  } else if [winlog][channel] == "System" {
    elasticsearch {
        hosts => "https://10.0.0.4:9200"
        index => "%{[client_id]}-windows.event.system-default"
        user => "elastic"
        password => "RrxQ3yerPDLfYja4ZqAD"
        ssl => "true"
```

```
        cacert => "/etc/logstash/shared/http_ca_elasticsearch.crt"
        action => "create"
    }
  }
}
```

Tabla 5.- Configuración del *pipeline* Winlogbeat.

/etc/logstash/pipelines/filebeat.conf

```
input {
    pipeline{
        address => filebeat
    }
}
 filter {
     mutate {
        add_field => { "client_id" => "tfg" }
        add_field => { "geo_IP" => "es" }
        add_field => { "network_type" => "NAT Network" }
     }
}
output {
  if [@metadata][pipeline] {
       #  if [log][file][path] == "/var/log/iptables.log" {
            elasticsearch {
                hosts => "https://10.0.0.4:9200"
                index => "%{[client_id]}-flatfile.fw.iptables-default"
                manage_template => false
                pipeline => "%{[@metadata][pipeline]}"
                user => "elastic"
                password => "RrxQ3yerPDLfYja4ZqAD"
                ssl => "true"
                cacert => "/etc/logstash/shared/http_ca_elasticsearch.crt"
                action => "create"
            }
       #  }
  }
  else if [log][file][path] == "/var/log/auth.log" {
    elasticsearch {
        hosts => "https://10.0.0.4:9200"
        index => "%{[client_id]}-flatfile.linux.authentication-default"
        manage_template => false
        user => "elastic"
        password => "RrxQ3yerPDLfYja4ZqAD"
        ssl => "true"
        cacert => "/etc/logstash/shared/http_ca_elasticsearch.crt"
        action => "create"
    }
  }
  else if [log][file][path] == "/var/log/kibana/kibana.log" {
    elasticsearch {
        hosts => "https://10.0.0.4:9200"
        index => "%{[client_id]}-flatfile.linux.authentication-kibana"
        manage_template => false
        user => "elastic"
        password => "RrxQ3yerPDLfYja4ZqAD"
        ssl => "true"
        cacert => "/etc/logstash/shared/http_ca_elasticsearch.crt"
        action => "create"
    }
  }
}
```

Tabla 6.- Configuración del *pipeline* Filebeat.

Elastic recomienda el uso de un usuario menos privilegiado para enviar los registros de Logstash a Elasticsearch. Para ello, habría que activar dicho usuario, si se desea utilizar un usuario creado por Elastic con permisos por defecto (48), o crear uno nuevo y ajustar sus permisos (45). Ninguna de las dos opciones se ha seguido en esta implementación puesto que es un entorno de pruebas, pero se recomienda realizarla en entornos de producción.

En las Tablas 5 y 6, se puede observar que los datos son segregados nuevamente en los *pipelines* según el tipo de datos para almacenarlos en el *data stream* correcto en Elasticsearch

(49), es decir, en el *pipeline filebeat.conf* se clasifican los registros y se dividen según sean registros de firewall o de autenticación, y dentro de los registros de autenticación se divide entre los registros de Linux y los de Kibana. Por otra parte, en el *pipeline winlogbeat.conf* se clasifican según el tipo de registro de los eventos.

6.3. Instalación y configuración de Beats

Tras instalar y configurar el agente (Logstash), la base de datos y el motor central (Elasticsearch), queda el último componente del SIEM, las fuentes de datos. Dividiremos esta sección en dos partes, una para instalar Filebeat en las máquinas linux-VM y elastic-VM y otra para instalar Winlogbeat en la máquina anfitriona.

Para empezar, hay que instalar Filebeat en la máquina linux-VM, configurarlo como servicio y configurar el archivo filebeat.yml para que en la primera conexión se conecte a Elasticsearch y Kibana (50) (51) (52). Además, hay que configurar el fichero de configuración de Kibana en elastic-VM para que escuche en la dirección IP de la red NAT y reiniciar el servicio (53). Entonces, al ejecutar Filebeat con el parámetro "*-e setup*", se cargarán todos los *dashboards* y *pipelines* de Filebeat en Elasticsearch (54). En concreto, este paso servirá para poder procesar los registros del firewall de linux IPtables de forma correcta a través del módulo de integración de Elastic (55).

A continuación, se configurará filebeat.yml para que se conecte a partir de ahora a través de Logstash por el puerto 5044 y con los certificados creados anteriormente para la fuente de datos Filebeat (56). Además, hay que concretar qué archivos de la máquina linux-VM debe leer el Beat, necesitaremos registros del firewall para detectar ataques de denegación de servicio y escáneres de puertos, y también necesitaremos registros procedentes del archivo auth.log, el cual posee registros relativos a la autenticación, tanto inicios de sesión exitosos como fallidos, por ello es la fuente de datos adecuada para monitorear ataques de contraseñas (57) (55) (4).

Los registros del firewall IPtables se almacenan por defecto en el fichero kern.log en distribuciones basadas en Debian, pero como el mismo fichero almacena registros relacionados con problemas del núcleo, de hardware y de conexiones de red, se redireccionarán los registros del firewall a un nuevo fichero llamado iptables.log creando un archivo de configuración en la carpeta del demonio de syslog (4) (58). Por ello, en el fichero de configuración de Filebeat, se especificarán los ficheros iptables.log y auth.log como fuentes de datos de este Beat (57).

Para terminar la configuración de esta fuente de datos, se activará el módulo de IPtables y se configurará para que procese los datos del archivo que contiene los registros de IPtables (55). Una vez configurado Filebeat, se debe iniciar como un servicio y crear una regla en el firewall que registre cualquier paquete de red, esta regla se puede observar en la Figura 33.

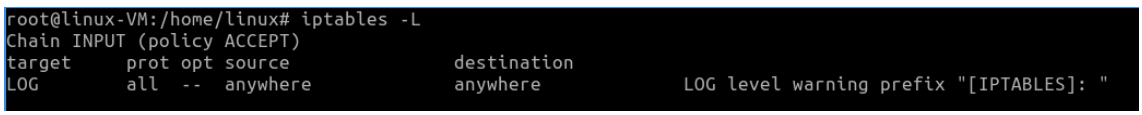

Figura 33.- Regla de IPtables para registrar cualquier conexión de red.

La detección de accesos a Kibana fuera del horario laboral también necesita un agente Filebeat instalado que envíe registros del fichero kibana.log ubicado en la máquina elastic-VM. Por ello,

habrá que instalar Filebeat en dicha máquina y especificar la ruta al fichero kibana.log en el fichero de configuración filebeat.yml para que envíe los registros a Logstash (50).

Para instalar Winlogbeat, se deben seguir los pasos de la documentación y ejecutarlo como un servicio (59). A continuación, se debe configurar el fichero winlogbeat.yml para que se conecte a la máquina Logstash a través de la red Host-Only, usando los certificados creados anteriormente para este beat, firmados por el CA de Logstash (56). Finalmente, hay que especificar el tipo de eventos que queremos recibir de Windows. Siguiendo el *pipeline* creado anteriormente en Logstash, se esperan registros de aplicación, seguridad y sistema (59).

Una vez se han instalado y configurado todos los componentes del SIEM, hay que probar que se reciben los registros y se almacenan en el índice correcto antes de crear los *data streams*. Para ello, desde *Stack Management* → *Index Management* → *Indices* en la interfaz de Kibana, se pueden ver los índices existentes en la actualidad, Figura 34.

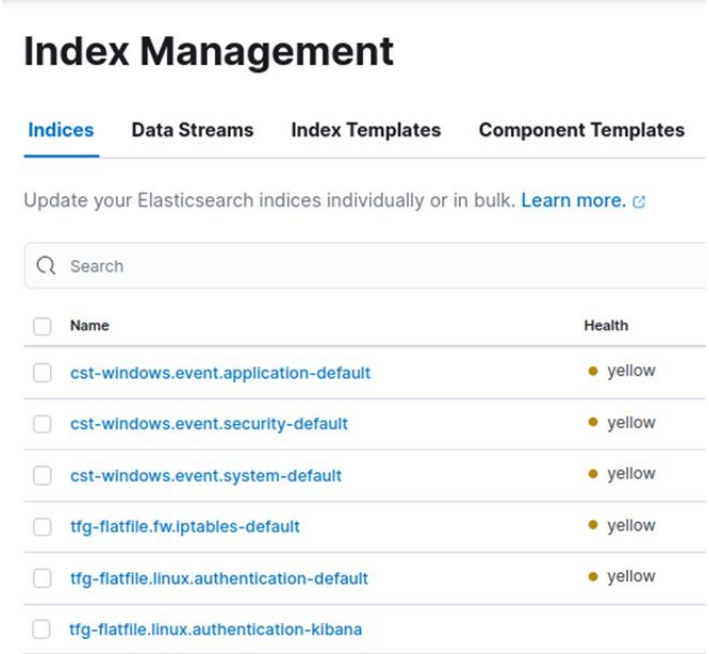

Figura 34.- Índices creados en el SIEM.

A continuación, se crearán los *data streams* para automatizar la administración de los índices. En primer lugar, se obtiene el *mapping* [7] de aquellos índices cuyos registros han sido correctamente procesados, es decir, los índices de Winlogbeat y Filebeat, que son procesados por defecto, y los del firewall IPtables, que son procesados mediante los *pipelines* cargados en Elasticsearch y el módulo de integración. A continuación, se crea una política ILM definiendo en qué etapas (*hot*, *warm*, *cold*, *frozen*, *delete*) estarán los índices del *data stream* y cuánto durarán en cada una de ellas (60). Luego se creará un *component template* donde se definirá el campo que se utilizará de referencia temporal y, finalmente, se creará un *data stream* que comprenda todos los objetos creados anteriormente y se aplique a los índices correspondientes según un patrón. Durante el proceso de creación del *data stream*, se aplicarán los *mappings* copiados anteriormente y se cambiará el tipo de datos de aquellos campos cuyo valor sea una dirección IP de *Text* y *Keyword*, que son los valores por defecto, a IP.

[7] https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-mapping.html

Tras crear los *data streams*, para comprobar que todo ha ido como debería, hay que eliminar los índices creados en el paso anterior y esperar a que reaparezcan como *data streams* en *Stack Management* → *Index Management* → *Data Streams*, Figura 35.

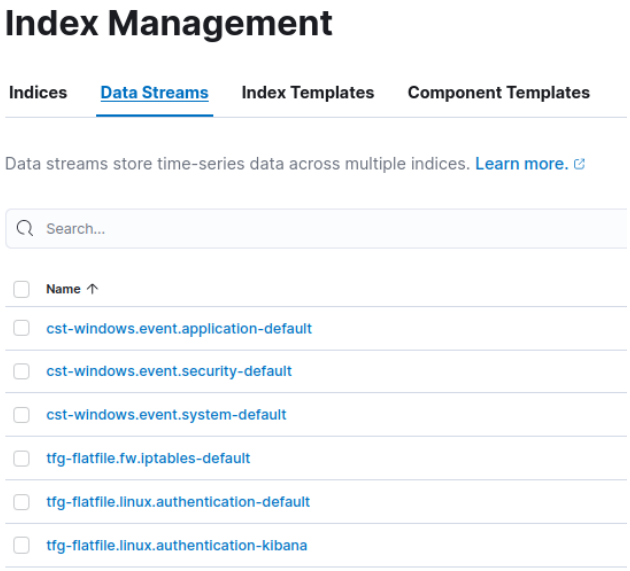

Figura 35.- *data streams* creados en el SIEM.

6.4. Creación de reglas

En esta sección crearemos las reglas especificadas en el diseño de casos de uso. Para activar la creación de reglas en Kibana, se debe crear una clave de cifrado a través de la terminal para los objetos que almacena Kibana y añadirla al fichero de configuración de Kibana. Esto es así puesto que las reglas se almacenan como objetos y deben ser protegidas (61). Antes de comenzar a crear las reglas, se ha decidido nombrarlas en inglés para mantener una terminología internacional, al menos en el SIEM.

Una vez disponible la opción de crear reglas, desde la interfaz *Security* → *Rules*, se creará la primera regla, en este caso TFG - Port Scanning Detected. Esta regla generará una alarma cada vez que una dirección IP se intente conectar a 2000 puertos distintos de otra IP durante un periodo de 5 minutos. Para ello hay que configurar la regla como se especifica en la Figura 36, definir los campos deseados como el título, la descripción, el autor o tácticas y técnicas del Framework de Mitre Att&ck, como en la Figura 37, y definir la periodicidad a la que la regla se activará. Todas las reglas del entorno de pruebas se activarán cada 5 minutos y realizarán búsquedas de 6 minutos en el pasado.

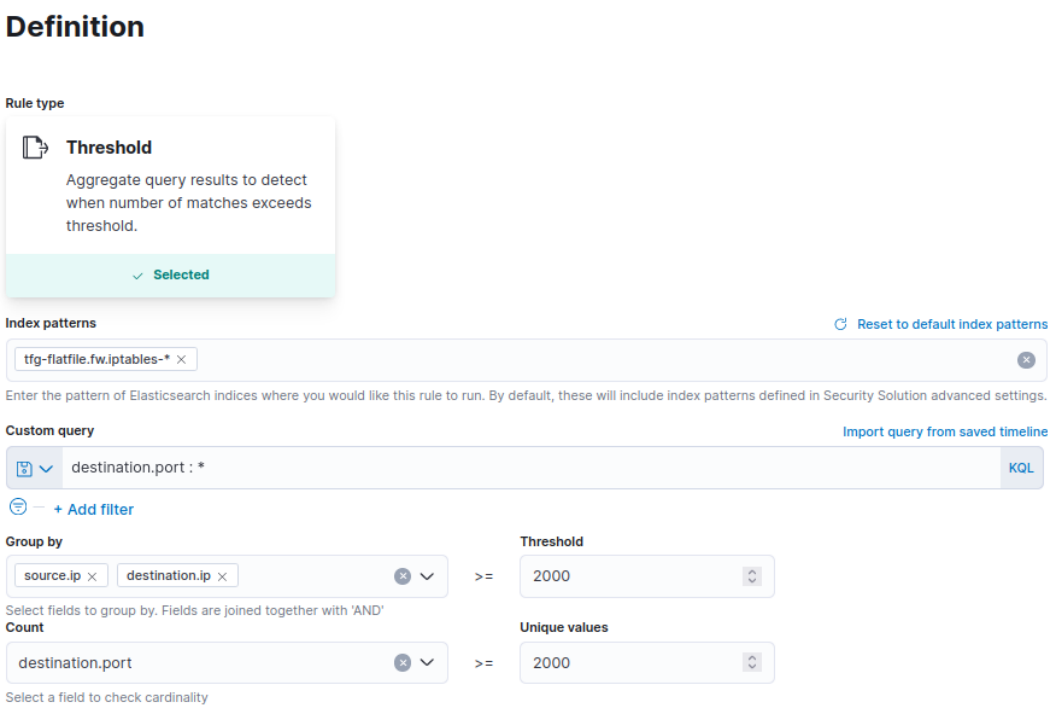

Figura 36.- Configuración de la regla de Escáner de puertos.

Figura 37.- Definición de la regla de Escáner de puertos.

En la Figura 36 se especifica la configuración de la regla. Esta regla es de tipo *Threshold* puesto que queremos agrupar los registros del firewall por direcciones IP y generar una alarma si el campo de puerto destino alcanza un cierto número de valores únicos. Este número o *threshold* depende del entorno que se desee monitorizar. En este caso, el entorno de pruebas no tiene mucho tráfico, pero para evitar falsos positivos se definirá 2000 conexiones a puertos distintos como *threshold* mínimo. En la configuración de la regla también se puede especificar el patrón de índices para reducir los resultados y mejorar la eficiencia, gracias a la convención de nombres propuesta podemos especificar un cliente o una fuente de datos. En el caso propuesto, se ha seleccionado la fuente de datos IPtables del cliente TFG.

La segunda regla se llamará TFG - SSH Brute Force Attack Detected y pretende detectar múltiples intentos de autenticación fallidos a un sistema a través de SSH. Para configurarla se ha utilizado el tipo de regla *Threshold*, como en el escáner de puertos, y los registros almacenados en el índice de autenticaciones de Linux del cliente TFG. Además, se ha

implementado una consulta que obtiene los registros de fallos de autenticación SSH en una máquina específica y, en caso de haber más de 10 fallos en el mismo sistema en menos de cinco minutos, la alarma saltará. La configuración mencionada se puede apreciar en la Figura 38 y el número de fallos puede ser modificado según las necesidades del entorno de producción.

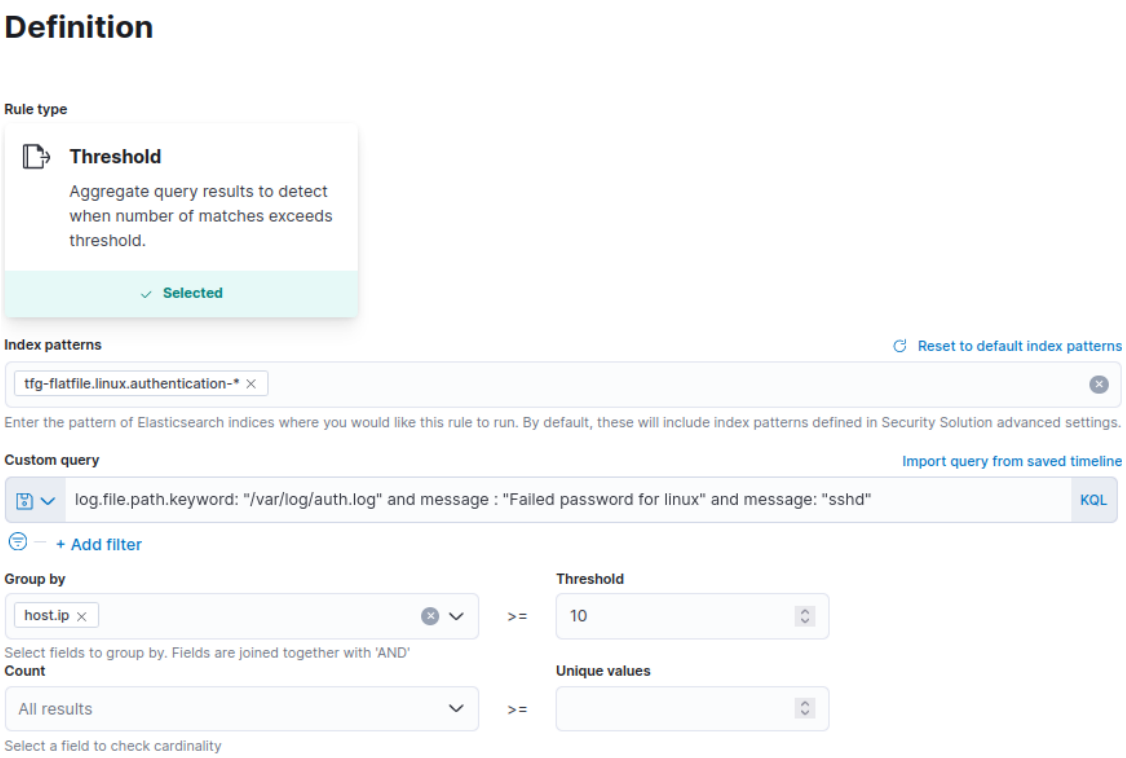

Figura 38.- Configuración de la regla de Ataque de fuerza bruta.

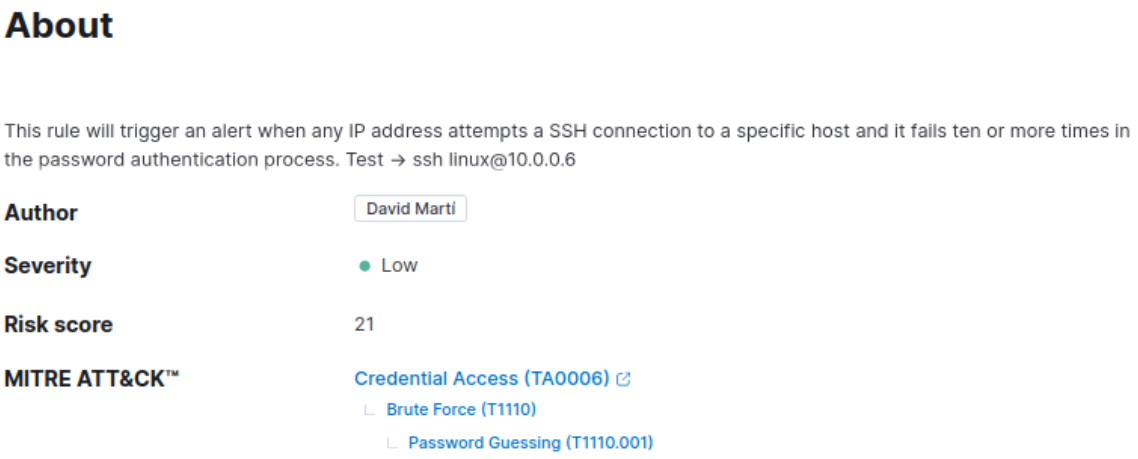

Figura 39.- Definición de la regla de Ataque de fuerza bruta.

En la Figura 39 aparece la definición de la regla, el autor y las tácticas y técnicas del framework de Mitre Att&ck asociadas con la regla.

A continuación, la regla de acceso a Kibana fuera del horario laboral se llamará TFG - Kibana Access during Non-Working Hours. Esta regla hará saltar una alarma cada vez que se detecte un acceso a la aplicación cualquier día desde las 17:00 de la tarde hasta las 6:00 de la mañana o los fines de semana. Para ello, en la Figura 40 se puede observar que se ha escogido una regla de

tipo *Custom Query*, por lo que se realizará una búsqueda simple de forma periódica en los índices especificados por el campo *index pattern*. En este caso, solo se ha de buscar en el índice que almacena los registros de autenticación de Kibana, dentro del cliente TFG (62).

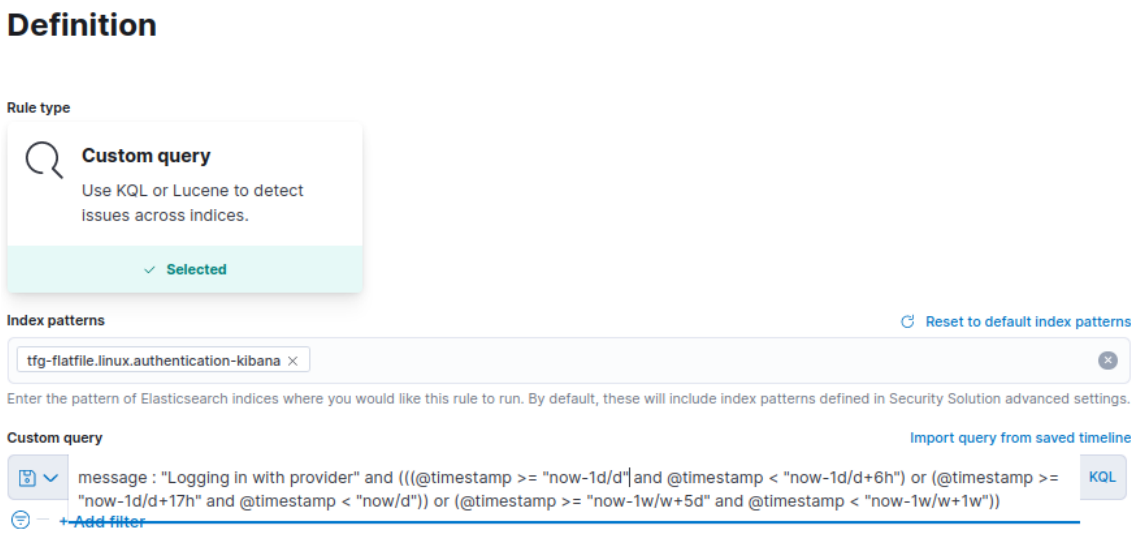

Figura 40.- Configuración de la regla de monitorización de acceso a Kibana.

Para lograr especificar el tiempo fuera del horario laboral se deben hacer algunos cálculos con el campo @timestamp. La Tabla 7 indica la expresión utilizada en Elastic para hacer cálculos con diferentes unidades temporales y la Tabla 8 muestra algunos ejemplos escogidos para entender los cálculos realizados en la regla de la Figura 40. Además, la Figura 41 nos muestra la descripción, el autor y las tácticas y técnicas de Mitre Att&ck relacionadas con esta regla.

y	Años
M	Meses
w	Semanas
d	Días
h	Horas
m	Minutos
s	Segundos
now	Ahora

Tabla 7.- Expresiones temporales en Elastic.

Si ahora = 2022-08-25T10:56:22	
now-1d/d	2022-08-24T00:00:00
now-1d/d + 17h	2022-08-24T17:00:00
now-1w/w+5d[8]	2022-08-27T00:00:00

[8] El 25 de agosto de 2022 es miércoles. Al redondear por abajo en la semana y sumarle 5 días se queda en el sábado de esa misma semana.

Tabla 8.- Cálculos temporales con Elastic.

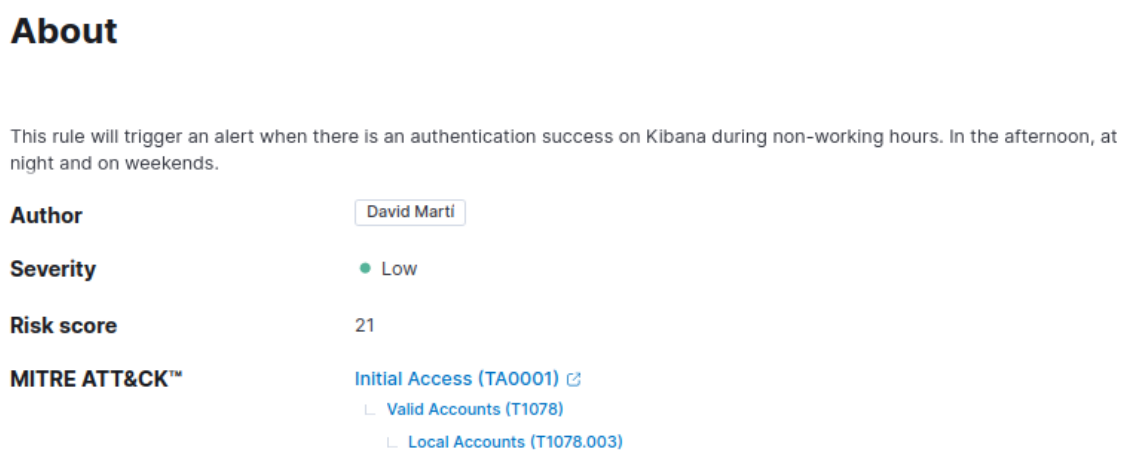

Figura 41.- Definición de la regla de monitorización de acceso a Kibana.

Para monitorizar la cuarta regla sobre la creación de cuentas de usuario en Windows para obtener persistencia, se ha decidido utilizar una regla de tipo *Custom Query* cuya consulta busque el evento 4720 (Se creó una cuenta de usuario) llevado a cabo de forma satisfactoria. Estos eventos son eventos de seguridad por lo que centraremos la búsqueda en el índice dedicado a los eventos de seguridad de Windows del cliente CST, tal y como se puede apreciar en la Figura 42, la nueva regla se llamará CST - Windows Account Created. En la Figura 43 aparece la descripción, el autor y las tácticas y técnicas relacionadas con la regla de creación de cuentas.

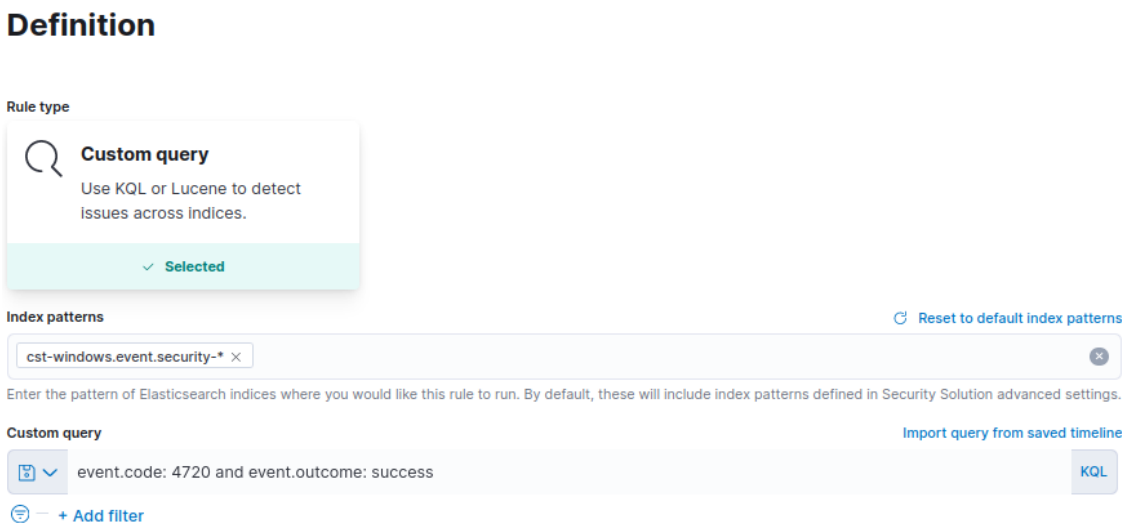

Figura 42.- Configuración de la regla de creación de cuentas en Windows.

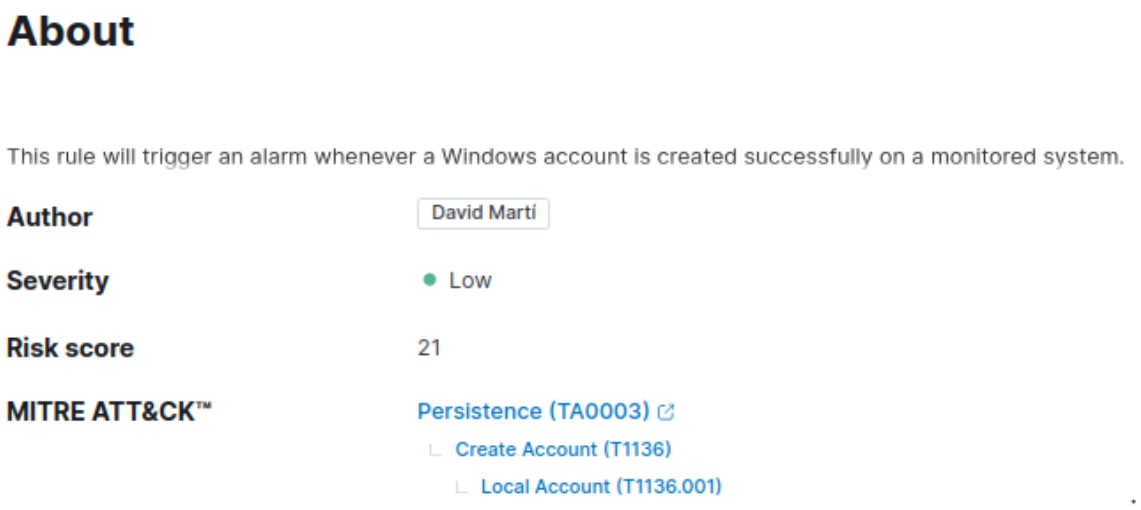

Figura 43.- Definición de la regla de creación de cuentas en Windows.

La quinta y última regla pretende detectar ataques DoS, por ello se llamará TFG - Volumetric DoS Attack Detected. Para monitorizar este tipo de ataques se ha usado una regla de tipo *Threshold* que, tal y como se puede apreciar en la Figura 44, saltará cuando el firewall de IPtables detecte 50000 conexiones desde una misma IP fuente en menos de 5 minutos (62).

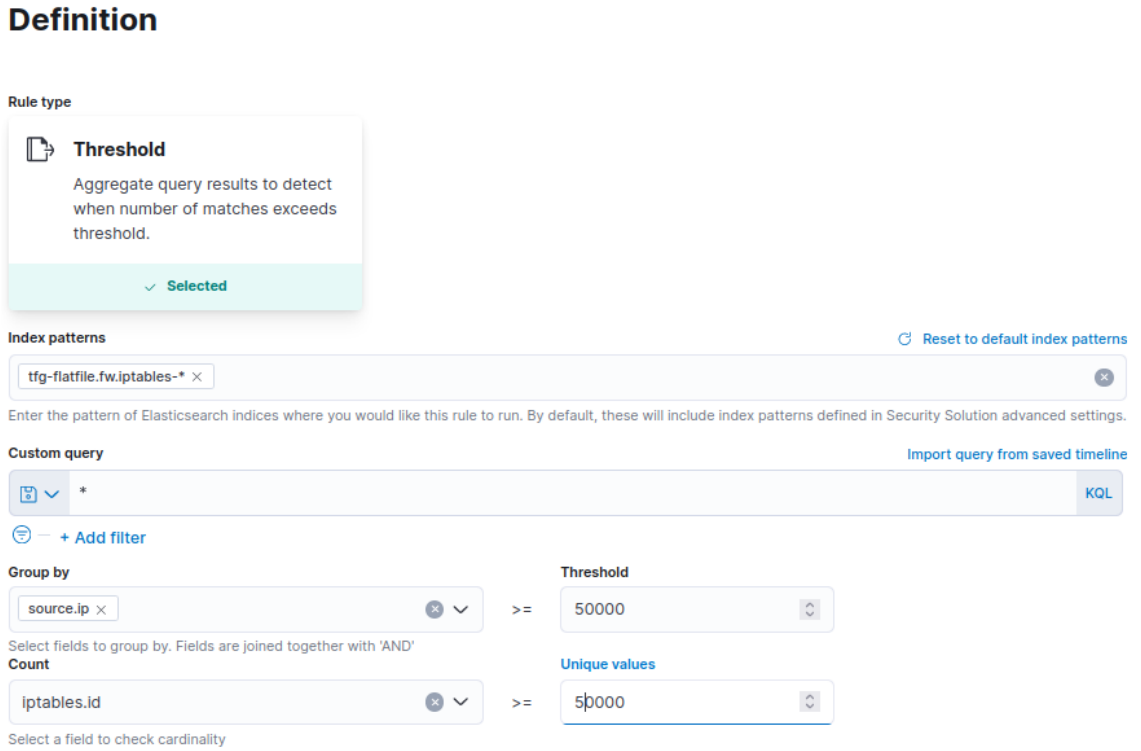

Figura 44.- Configuración de la regla de detección de ataques DoS.

En la Figura 45 aparece la definición, el autor y las tácticas y técnicas de Mitre Att&ck relacionadas con esta regla.

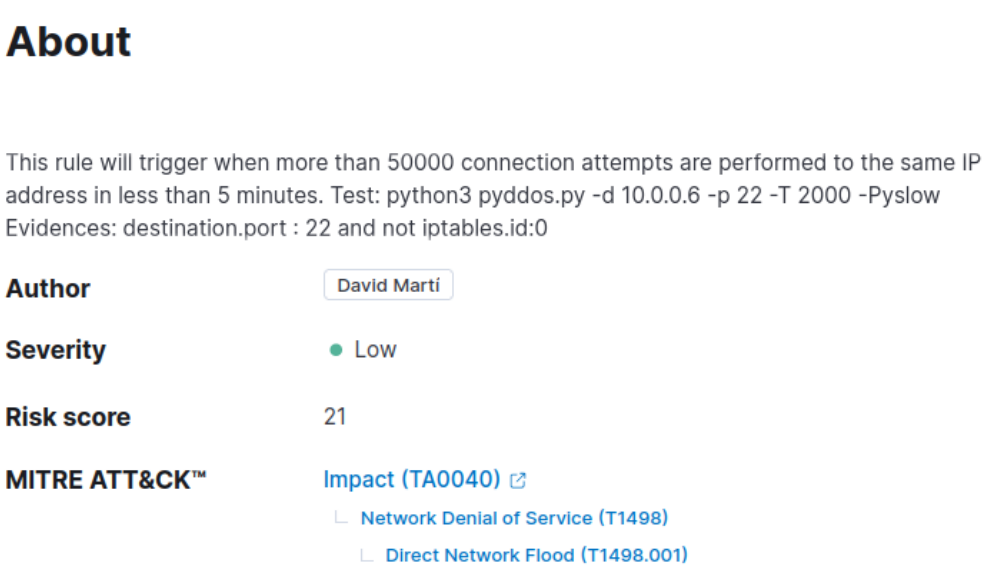

Figura 45.- Definición de la regla de detección de ataques DoS.

Finalmente, se pueden revisar todas las reglas creadas en *Security* → *Rules*, Figura 46:

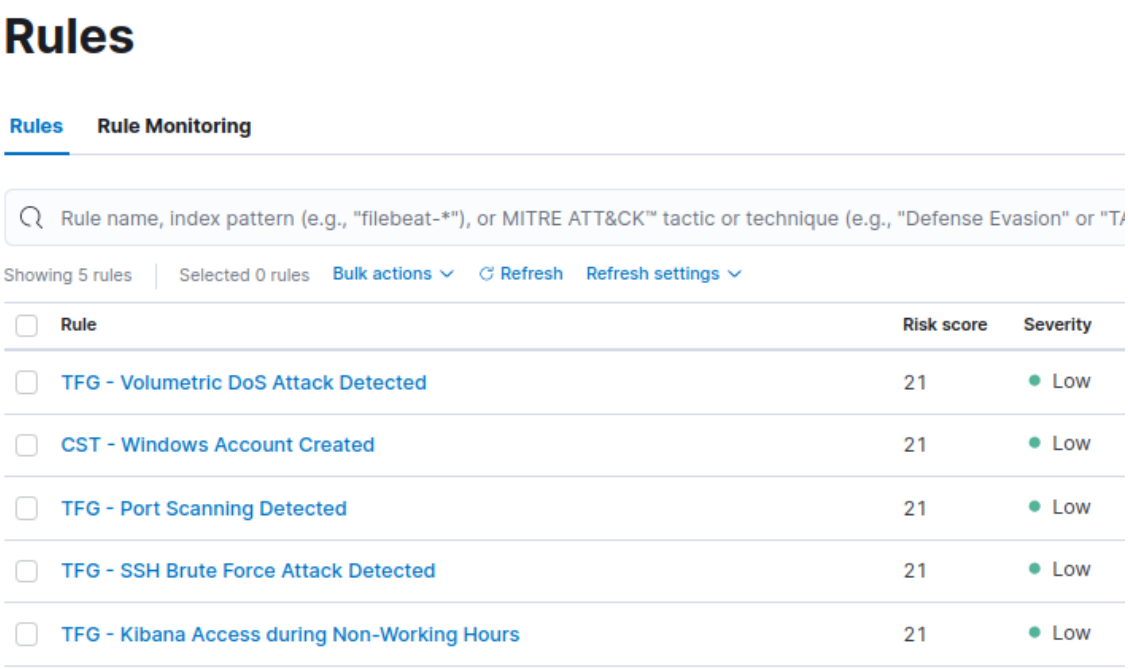

Figura 46.- Conjunto de alarmas creadas en Elastic Security.

7. Pruebas

Ya se ha visto a lo largo del capítulo 6. Desarrollo de la solución algunas pruebas de que Elastic recibe los registros de las fuentes de datos elegidas y los almacena en los *data streams* correctos. Por lo que, en este capítulo, se obviará demostrar el funcionamiento general de las aplicaciones y nos centraremos en la confidencialidad de las comunicaciones y en las reglas.

En la sección 7.1. Pruebas de cifrado se demostrará que tanto las comunicaciones entre Logstash y las fuentes de datos, como las comunicaciones entre Logstash y Elasticsearch están siendo cifradas correctamente.

En la sección 7.2. Pruebas de reglas se demostrará el correcto funcionamiento de las reglas creadas en el capítulo anterior.

7.1. Pruebas de cifrado

Para comprobar la confidencialidad de los datos, se instalará un *sniffer* de red como Wireshark[9] en la máquina logstash-VM y se iniciarán las fuentes de datos una a una.

En primer lugar, se probará la conexión del cliente CST desde la máquina Windows a través de Winlogbeat con Logstash. El resultado se muestra en la Figura 47, donde se puede observar que los datos se envían a través de TLSv1.3.

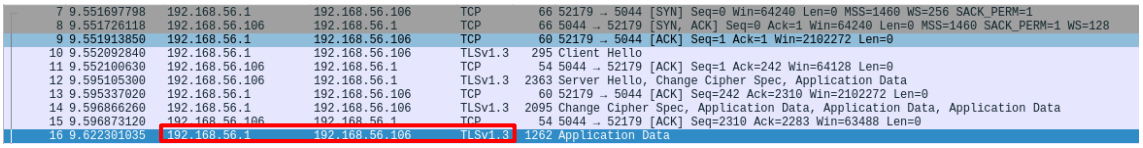

Figura 47.- Cifrado Winlogbeat-Logstash.

En segundo lugar, se cambiará la interfaz de red y ejecutaremos Filebeat desde la máquina linux-VM del cliente TFG. El resultado se muestra en la Figura 48, donde al igual que en la prueba anterior, se puede observar que los datos se envían a través de TLSv1.3.

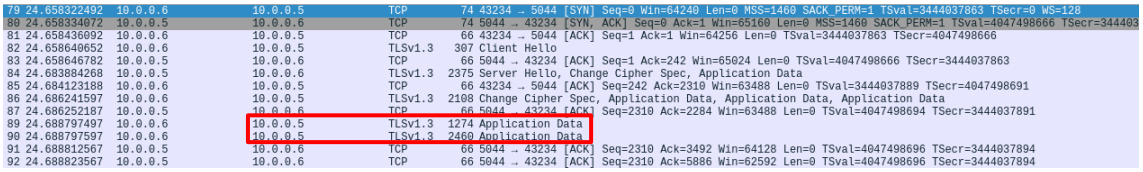

Figura 48.- Cifrado Filebeat-Logstash en logstash-VM.

En tercer lugar, se comprobará la conexión entre el Filebeat de la máquina elastic-VM y Logstash. El resultado se muestra en la Figura 49, donde se puede observar de nuevo que los datos se envían a través de TLSv1.3.

[9] https://www.wireshark.org/

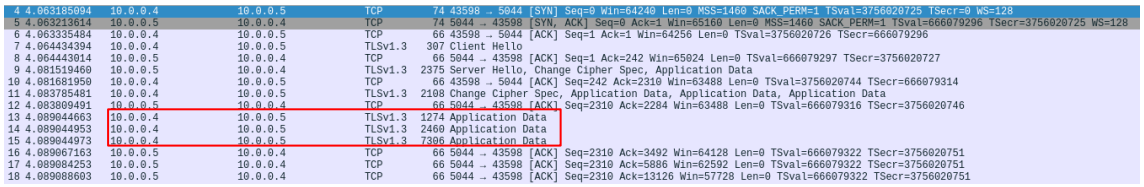

Figura 49.- Cifrado Filebeat-Logstash en elastic-VM

Finalmente, para comprobar el cifrado entre Logstash y Elasticsearch, se han parado todas las fuentes de datos y se ha ejecutado únicamente Winlogbeat. Sin embargo, en lugar de escuchar por la interfaz de la red Host-Only como en la primera prueba, esta vez se han obtenido los paquetes de la interfaz de red NAT, donde la máquina logstash-VM solo se comunicaba con la máquina elastic-VM para reenviar los registros de Windows. El resultado se muestra en la Figura 50, donde se puede observar que los datos se envían a través de TLSv1.2.

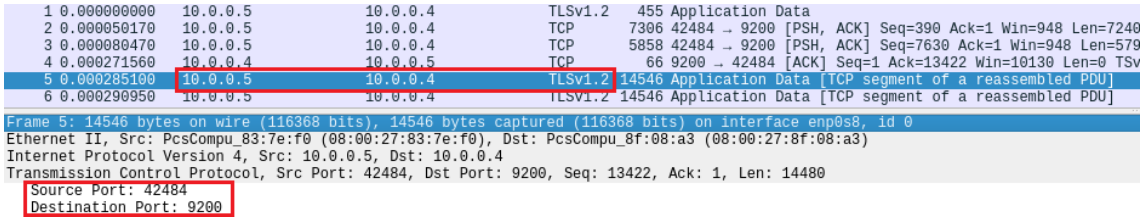

Figura 50.- Cifrado Logstash-Elasticsearch.

7.2. Pruebas de reglas

Siguiendo el orden en el que se han presentado los casos de uso, empezamos con la primera regla TFG - Port Scanning Detected. Esta regla debería saltar cuando cualquier sistema monitorizado está siendo objeto de un escáner de puertos, es decir, que recibe conexiones desde una misma fuente a muchos puertos distintos en poco tiempo. Para probar el correcto funcionamiento de la regla, se ha creado un script en bash que ejecuta netcat sobre una dirección IP objetivo a través de todos los puertos desde el 1 hasta el 65535. El script se puede analizar en la Tabla 9 y, como resultado, tras unos minutos aparecerá en Elastic Security la alarma generada por el escáner de puertos, Figura 54.

port.sh

```
#!/bin/bash
if ["$1" == "-h" ]
then
  echo "Usage: ./port.sh [IP]"
  echo "Example ./port.sh 192.168.1.10"
else
  echo "Scanning ports from $1"
  nc -nvz $1 1-65535
  echo "Done"
fi
```

Tabla 9.- Prueba de la regla TFG - Port Scanning Detected.

La siguiente regla para verificar es TFG - SSH Brute Force Attack Detected. Esta regla salta cuando una máquina monitorizada recibe diez o más conexiones SSH erróneas en menos de 5 minutos. Para comprobar su correcto funcionamiento, se ha ejecutado una conexión SSH cuatro veces con tres intentos de contraseña fallidos. La prueba se puede ver en la Figura 51 y los resultados en la Figura 54.

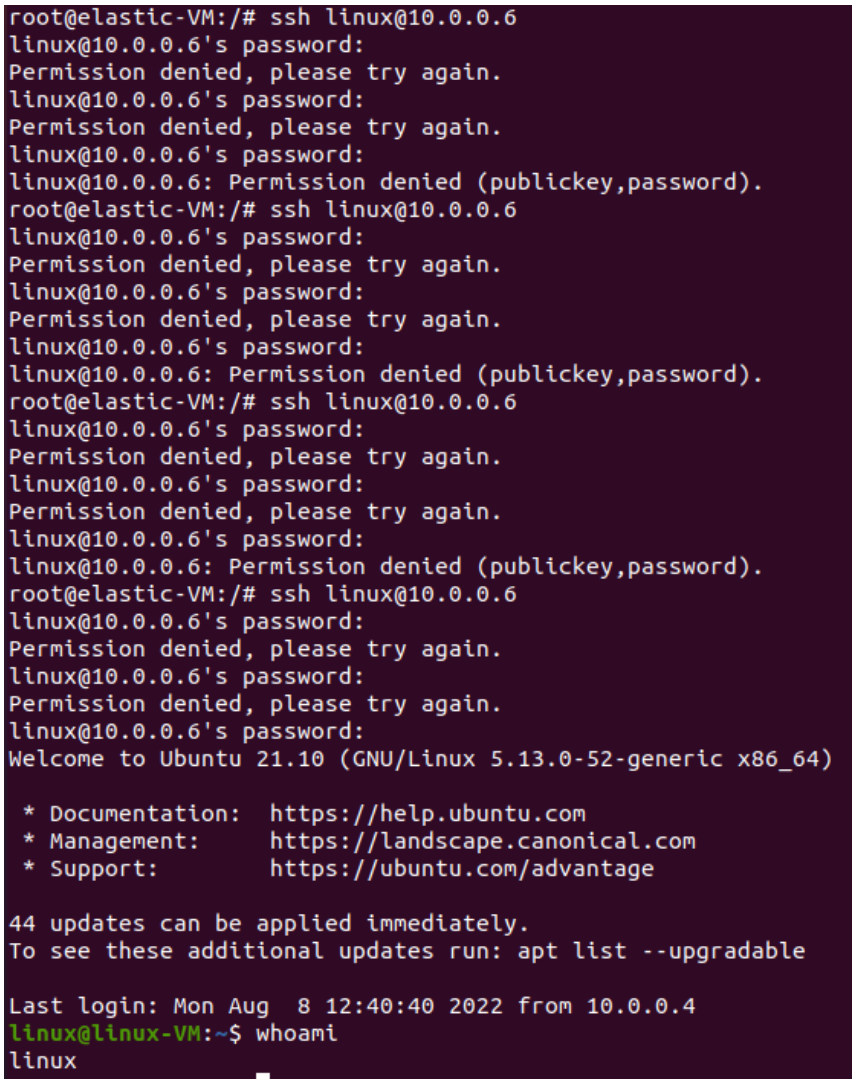

Figura 51.- Prueba de la regla TFG - SSH Brute Force Attack Detected.

La regla TFG - Kibana Access during Non-Working Hours es una regla que tan solo se puede verificar en ciertos momentos, es decir por la tarde o la noche o los fines de semana. Para ello, tan solo hay que cerrar sesión y volver a iniciar sesión en Kibana en el rango de tiempo especificado. Los resultados de la prueba se pueden observar el la Figura 54.

La regla CST - Windows Account Created saltará cuando se cree un nuevo usuario en cualquiera de los sistemas Windows monitorizados. Para comprobar su correcto funcionamiento, se puede abrir Powershell como administrador y ejecutar los comandos necesarios para crear un usuario nuevo en el sistema, tal y como aparece en la Figura 52, donde se crea un usuario llamado new.

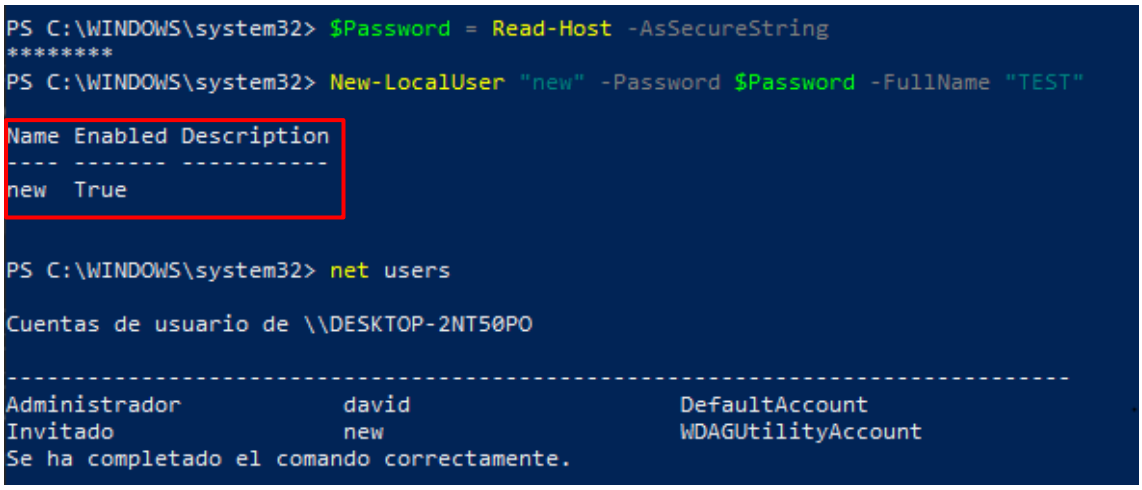

Figura 52.- Prueba de la regla CST - Windows Account Created.

La última regla TFG - Volumetric DoS Attack Detected pretende detectar grandes volúmenes de tráfico procedentes de una misma dirección IP. Para comprobar su correcto funcionamiento simularemos un ataque DoS al puerto 22 de la máquina linux-VM desde elastic-VM. Para simular un ataque DoS se ha utilizado una herramienta open source escrita en Python y publicada en GitHub llamada Pyddos[10], que permite realizar ataques DoS mediante envío masivo de paquetes SYN o peticiones, o a través de ataques *Slow DoS*. En nuestro caso especificaremos la dirección IP objetivo mediante el parámetro -d, especificaremos el puerto SSH con el parámetro -p y doblaremos el número de hilos por defecto con el parámetro -T. Finalmente, se especificará el tipo de ataque *Slow DoS* y dejaremos corriendo el programa durante unos minutos. El programa envía alrededor de mil peticiones por segundo. La prueba descrita se puede observar en la Figura 53.

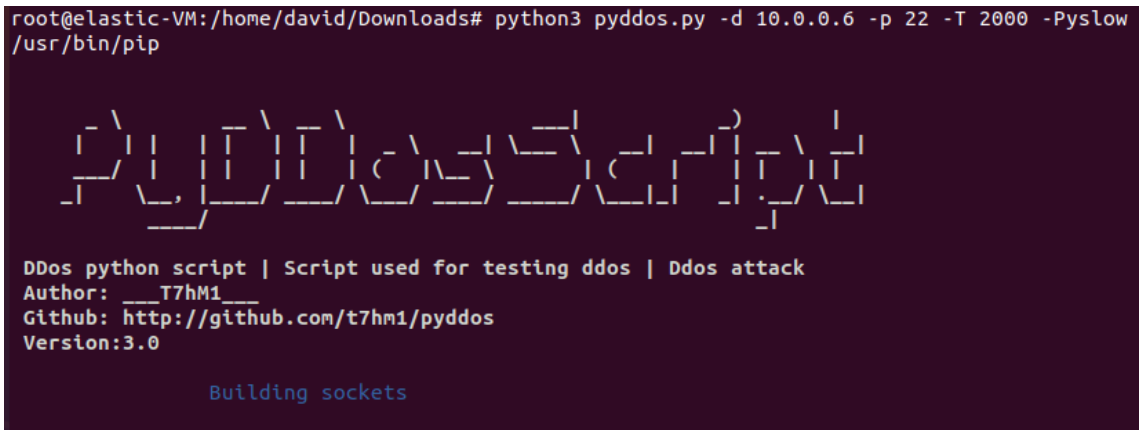

Figura 53.- Prueba de la regla TFG - Volumetric DoS Attack Detected.

En la Figura 54 se puede observar el resultado de las cinco pruebas realizadas y se comprueba que existe una nueva alarma por cada regla creada.

[10] https://github.com/mach1el/pyddos

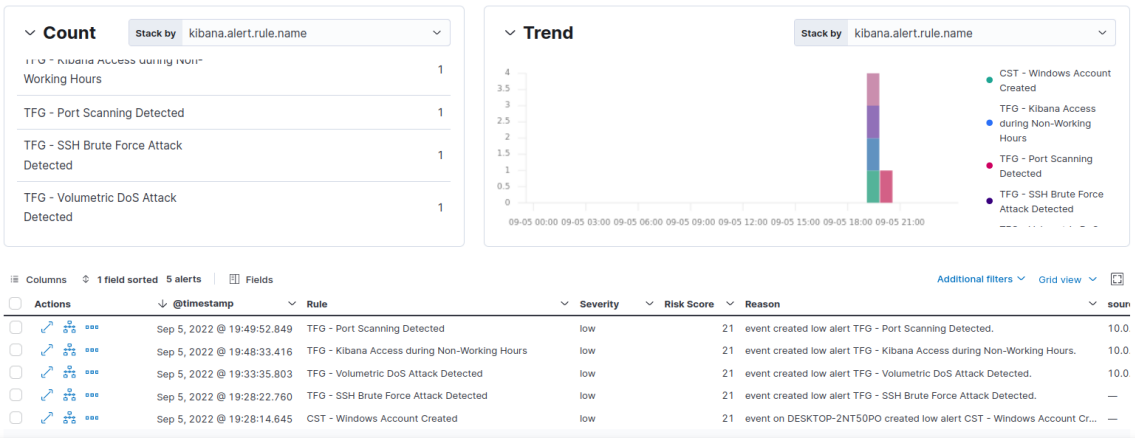

Figura 54.- Resultados de las pruebas realizadas.

8. Conclusiones

En este capítulo se extraerán las conclusiones del TFG y se expondrán algunas vías de investigación para proseguir con el trabajo realizado.

En el capítulo 8.1. Conclusión se comentarán las actividades realizadas a lo largo del proyecto, poniendo en valor el trabajo realizado por el estudiante y analizando los objetivos cumplidos.

En el capítulo 8.2. Futuras vías de trabajo se sugerirán varios temas abiertos para complementar la documentación y el laboratorio práctico realizado.

8.1. Conclusión

A lo largo de este TFG se han analizado los problemas de seguridad que enfrentan actualmente las empresas y una solución para detectarlos de forma rápida y eficiente, el SOC. Además, se han estudiado los retos que se pueden experimentar al implantar uno partiendo de cero y se ha propuesto un proyecto para implementarlo.

Para contextualizar al lector, se ha realizado un estudio de mercado de algunas de las herramientas más conocidas y valoradas del sector, y se ha profundizado en Elasticsearch y las herramientas que lo complementan: Kibana, Logstash y Beats, ya que se ha considerado que esta herramienta era idónea para llevar a cabo el propósito del TFG con los recursos disponibles.

A continuación, se ha obtenido el presupuesto base necesario para invertir durante el periodo inicial de despliegue. También, se ha diseñado un diagrama para implementar un SIEM de forma eficiente, robusta y segura en un entorno de producción, y otro para un entorno de pruebas con el mínimo consumo de recursos económicos, materiales, humanos y temporales posible.

Una vez diseñado el prototipo de pruebas, se ha implementado el SIEM Elastic Security en un entorno virtual con detalles técnicos, para que con ayuda de la documentación de Elastic y este documento, cualquier estudiante de cuarto curso del grado de Ingeniería Informática pueda reproducir el laboratorio. Además, no solo se ha instalado el SIEM, sino que también se han creado 5 casos de uso genéricos reales y se han implementado reglas para detectarlos. Para terminar la parte práctica, se han realizado pruebas para demostrar que todo lo implementado en el TFG funciona correctamente y se han superado con éxito.

Con todo lo anteriormente expuesto, se analizará el cumplimiento de los objetivos propuestos al inicio del proyecto:

OBJETIVO 1 - Divulgar información de defensa informática: el Centro de Operaciones de Seguridad es una herramienta de defensa informática avanzada muy utilizada en empresas de gran tamaño y en este proyecto se detalla en que consiste y como implementarlo.

OBJETIVO 2 - Documentar la creación de un prototipo de SOC de forma reproducible y asequible: con la solución propuesta e implementada, cualquier lector con un ordenador de gama media podría monitorizar su red domestica para proteger sus propiedades TI. Así pues, cualquier empresa podría encontrar en este TFG una guía detallada para proteger sus activos.

OBJETIVO 3 - Aprender a crear reglas en un SIEM: se han extraído cinco técnicas de ataque del framework Mitre Att&ck, se ha construido un caso de uso para cada una de ellas y se ha pasado a la práctica, creando reglas totalmente funcionales para detectar dichos ataques.

OBJETIVO 4 - Mostrar las dificultades encontradas durante la implementación de un SOC: se han analizado los desafíos que se pueden experimentar a lo largo del proceso de creación de un SOC y despliegue del SIEM y, además, se ha mostrado el trabajo titánico y la monumental cantidad de recursos que son necesarios para realizar de forma exitosa un proyecto de este tamaño.

Finalmente, se puede afirmar que se han logrado todos los objetivos propuestos de manera satisfactoria y el lector tiene las herramientas y documentación necesarias para crear un entorno TI más seguro.

8.2. Futuras vías de trabajo

La implementación de un SOC es un proyecto de gran magnitud cuya dificultad se acentúa en caso de que el SIEM elegido sea Elastic Security, ya que al ser un herramienta *open source* es extremadamente maleable y necesita mucha más configuración que un producto de pago. Por ello, existen muchas vías de investigación para mejorar la eficiencia del SIEM y evolucionar a un SOC maduro:

En primer lugar, la infraestructura de un SOC puede ser el tema mismo de otro Trabajo de Fin de Grado. La elección de la tecnología como los firewalls, IDS/IPS, *honeypots*, así como el diseño de una arquitectura de red segura basada en la confianza cero y la defensa en profundidad, capaz de proteger un entorno dedicado a la seguridad de otros entornos supone un reto digno de investigación y documentación.

En segundo lugar, la arquitectura elegida para el entorno de pruebas es bastante simple debido a los recursos y probar a desplegar nuevas arquitecturas con clústeres de varios nodos, replicación y monitorización, servicios de colas, conexiones Logstash-to-Logstash, etc. es una buena forma de avanzar hacia un entorno complejo, seguro y robusto. En cuanto a la monitorización mencionada, el uso de los módulos de Metricbeat puede proporcionar una visión del estado general del SIEM a través del *Stack Monitoring* de Kibana y su documentar su implementación en un entorno real resulta de interés para todo aquel que utilice Elastic para monitorizar sistemas.

En tercer lugar, el procesamiento de registros desde los *pipelines* de Logstash es uno de los grandes desafíos que enfrenta un SOC en sus inicios y que en este proyecto no ha sido necesario implementar. Un trabajo que detalle el uso de más módulos de Beats, integraciones de Fleet y el uso de patrones Grok y expresiones regulares para el correcto procesamiento de los registros complementaría a la perfección el presente documento.

En cuarto lugar, la monitorización de más casos de uso y más escenarios de ataque de los casos de uso descritos en este trabajo, y la creación de guías de análisis y respuesta a incidentes de todos ellos, acompañados de demostraciones de ataques ficticios, tal y como se ha hecho en este TFG, mejoraría la funcionalidad del SOC desplegado en este proyecto. Además, ejemplos de uso de *Cases* y *Timelines* de Elastic Security y la aplicación de *Threat Hunting* con Kibana en

las pruebas de análisis de los casos de uso resultarían de mucho interés para cualquier persona, empresa u organización que desee probar a implementar la solución propuesta en este TFG.

Para terminar, este documento se centra en las aplicaciones del *Elastic Stack* (Elasticsearch, Logstash, Kibana y Beats). Sin embargo, Elastic posee un surtido de productos que complementan a la perfección los ya descritos a lo largo del proyecto. Algunos de estas herramientas que sería interesante estudiar para aplicarlas en el entorno del SOC son Elastic Agent, Fleet, APM y Endgame.

9. Glosario

0-Day: Vulnerabilidad con exploit asociado que a fecha de publicación no ha sido resuelta por el fabricante (63).

1-Day: Vulnerabilidad que ha sido resuelta por el fabricante pero usada por los atacantes en sistemas no actualizados (63).

Active Directory: Active Directory almacena información acerca de los objetos de una red y facilita su búsqueda y uso por parte de los usuarios y administradores. Active Directory usa un almacén de datos estructurado como base para una organización jerárquica lógica de la información del directorio (64).

Advanced Persistent Threat (APT): Amenaza persistente. y avanzada. Típicamente es una nación o grupo patrocinado por un estado (63).

Antivirus: Tecnología básica utilizada para detección de software malicioso (63).

Address space layout randomization (ASLR): Técnica de seguridad relacionada con la prevención de vulnerabilidades asociadas a la corrupción de memoria. Cambia las posiciones de memoria de un proceso de forma aleatoria en memoria (65).

Ataque de Denegacion de Servicio (DoS): Un ataque de denegación de servicio, tiene como objetivo inhabilitar el uso de un sistema, una aplicación o una máquina, con el fin de bloquear el servicio para el que está destinado. Este ataque puede afectar, tanto a la fuente que ofrece la información como puede ser una aplicación o el canal de transmisión, como a la red informática (66).

Ataque de fuerza bruta: Un ataque de fuerza bruta es un intento de descifrar una contraseña o nombre de usuario, de buscar una página web oculta o de descubrir la clave utilizada para cifrar un mensaje, que consiste en aplicar el método de prueba y error con la esperanza de dar con la combinación correcta finalmente (67).

Autoridad de certificación (CA): Es una empresa u organización que actúa para validar las identidades de las entidades (como sitios web, direcciones de correo electrónico, empresas o personas individuales) y vincularlas a claves criptográficas mediante la emisión de documentos electrónicos conocidos como Certificados digitales (68).

Autoridad de registro (RA): Es una entidad que identifica de forma inequívoca al solicitante de un certificado. La Autoridad de Registro suministra a la Autoridad de Certificación los datos verificados del solicitante a fin de que la Autoridad de Certificación emita el correspondiente certificado (69).

Beats: Referido a Elastic, es una plataforma gratuita y abierta para agentes de datos con un solo propósito. Envían datos de cientos o miles de máquinas y sistemas a Logstash o Elasticsearch. Estos datos varían entre archivos de logs (Filebeat), métricas (Metricbeat), datos de red (Packetbeat), logs de eventos de Windows (Winlogbeat), información de auditoría (Auditbeat), monitoreo de tiempo de actividad (Heartbeat) y agentes sin servidor (Functionbeat) (70).

Centro de Operaciones de Redes (NOC): Es una central donde el equipo de redes puede monitorizar constantemente la actividad y salud de una red. El NOC actúa como una primera línea de defensa contra disrupciones y fallos en la red (71).

Centro de Operaciones de Seguridad (SOC): El SOC es una plataforma que permite la supervisión y administración de la seguridad del sistema de información a través de herramientas de recogida, correlación de eventos e intervención remota (72).

Caso de uso: En referencia a la ciberseguridad, es un determinado escenario, comportamiento o técnica que queremos representar, con el objetivo (en el marco de este TFG) de identificar los puntos de entrada y fuentes de datos que serán necesarios implantar para permitir una correcta monitorización del ataque por parte del equipo de seguridad.

Chief Information Security Officer (CISO): Es el director de seguridad de la información. Básicamente es un rol desempeñado a nivel ejecutivo y su función principal es la de alinear la seguridad de la información con los objetivos de negocio. De esta forma se garantiza en todo momento que la información de la empresa está protegida adecuadamente (73).

Clúster: En referencia a la Elastic, un clúster de Elasticsearch es un grupo de nodos que poseen el mismo atributo *cluster.name*. Cuando un nodo se une o sale del clúster, el clúster se reorganiza y distribuye eventualmente la información entre el resto de los nodos disponibles. Si se corre una única instancia de Elasticsearch, entonces se obtiene un clúster de un solo nodo (74).

Data Streams: En referencia a la Elastic, un *data stream* almacena datos organizados temporalmente en varios índices bajo un mismo alias, a través del cual realizar todas las peticiones. Los *data streams* son eficientes en el almacenamiento de registros, eventos, métricas y otros datos generados continuamente sin interrupción (75).

Dashboard: En referencia a la Elastic, los *dashboards* en Kibana permiten crear rápidamente vistas que reúnen gráficos, mapas y filtros para mostrar el panorama completo de los datos de Elasticsearch. Desde pantallas de monitoreo de amenazas en tiempo real hasta resúmenes ejecutivos que muestran los indicadores clave de rendimiento (76).

Documento: En referencia a la Elastic, Elasticsearch almacena datos como documentos JSON. Cada documento correlaciona un conjunto de claves (nombres de campos o propiedades) con sus valores correspondientes (textos, números, Booleanos, fechas, variedades de valores, geolocalizaciones u otros tipos de datos) (77).

Elasticsearch: Es un motor de búsqueda y analítica distribuido, gratuito y abierto para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y no estructurados (77).

Elastic AMP: Es un sistema de monitorización de rendimiento de aplicaciones, permite monitorizar en tiempo real aplicaciones y servicios, recolectando información detallada sobre el rendimiento del tiempo de respuesta de peticiones, búsquedas en bases de datos, llamadas a cache, peticiones HTTP externas, etc (78).

Elastic Common Schema (ECS): Es una especificación de código abierto, desarrollada con el apoyo de la comunidad de usuarios de Elastic. ECS define un conjunto común de campos que se utilizan cuando se almacenan datos de eventos en Elasticsearch, como registros y métricas (38).

Elastic Endgame: Es la solución de Elastic para prevención, detección y respuesta (EPP + EDR) en *endpoints* y complementa la solución SIEM Elastic Security (79).

Elastic Security: Es la solución SIEM de Elastic y ofrece un motor de detección de ataques y fallos de configuración, un espacio para realizar las funciones de analista, visualizaciones interactivas de cómo se relacionan los procesos, administración de casos de alarmas y detección de ataques con *machine learning*, detección de anomalías y reglas de correlación de eventos (80).

ELK Stack: Es un grupo de productos de código abierto de Elastic diseñados para ayudar a los usuarios a tomar datos de cualquier tipo de fuente y en cualquier formato, y buscar, analizar y visualizar esos datos en tiempo real (81).

Enterprise Search: Es una colección de herramientas para buscar contenido, con la tecnología de Elasticsearch, permite crear experiencias de búsqueda para clientes y equipos internos (82).

Endpoint: Se encarga de monitorizar terminales (ordenadores y dispositivos móviles conectados a Internet) en búsqueda de actividades inseguras. En estos sistemas de seguridad intervienen tecnologías avanzadas como la Inteligencia Artificial (IA) y el *machine learning*, que pueden detectar actividades anómalas y contrarrestarla de forma inmediata. Se especializan en amenazas avanzadas como el *ransomware* o el *phishing*, además de ser efectivo contra malware y virus en general con una efectividad mucho más alta que los antivirus (83).

Exploit: Programa o método para subvertir la seguridad de un sistema a partir de una o más vulnerabilidades (63).

Falso positivo: Es un resultado falso para una actividad que nunca ocurrió, es decir, se producen cuando una alerta da la alarma, pero no se ha producido ningún ataque (4).

Firewall: Es un dispositivo de seguridad de la red que monitoriza el tráfico entrante y saliente y decide si debe permitir o bloquear un tráfico específico en función de un conjunto de restricciones de seguridad ya definidas (84).

Fleet: El Servidor de Fleet es el componente de infraestructura que gestiona la comunicación con los Agentes de Elastic. Proporciona el plano de control que actualiza los agentes y les indica realizar acciones como ejecutar OSQuery en los hosts o aislar hosts en la capa de red para contener las amenazas de seguridad (85).

Gestor de información y eventos de seguridad (SIEM): Es una solución de seguridad que ayuda a las organizaciones a reconocer posibles amenazas y vulnerabilidades de seguridad antes de que tengan la oportunidad de interrumpir las operaciones comerciales. Muestra anomalías en el comportamiento del usuario y utiliza inteligencia artificial para automatizar muchos de los procesos manuales asociados con la detección de amenazas y la respuesta de incidentes, y se ha convertido en un elemento básico en los centros de operaciones de seguridad (SOC) (12).

Hacktivismo: Realización de actos, normalmente maliciosos, en Internet para promover unas ideas políticas, religiosas o sociales (86).

Honeypot: Es un sistema informático que se "sacrifica" para atraer ciberataques, como un señuelo. Simula ser un objetivo para los hackers y utiliza sus intentos de intrusión para obtener

información sobre los cibercriminales y la forma en que operan o para distraerlos de otros objetivos (87).

Índices: En referencia a la Elastic, un índice de Elasticsearch es una colección de documentos relacionados entre sí (77).

Internet de las cosas (IoT): La Internet de las cosas (IoT) describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas (88).

ISO27001: Especifica los requisitos para establecer implementar, mantener y mejorar un sistema para administrar la seguridad de la información en una empresa (2).

Kafka: Apache Kafka emplea un sistema de colas para ayudar a sus clientes a mejorar el rendimiento de los *pipelines* en tiempo real (89).

Kerberos: Es un protocolo de autenticación de redes de ordenador creado por el MIT que permite a dos ordenadores en una red insegura demostrar su identidad mutuamente de manera segura (90).

Kibana: Kibana es una aplicación de *frontend* gratuita y abierta que se encuentra sobre el *Elastic Stack* y proporciona capacidades de visualización de datos y de búsqueda para los datos indexados en Elasticsearch (91).

LDAP (Lightweight Directory Access Protocol): Es un protocolo de la capa de aplicación TCP/IP que permite el acceso a un servicio de directorio ordenado y distribuido, para buscar cualquier información en un entorno de red. Generalmente un servidor LDAP se encarga de almacenar información de autenticación (92).

Listas blancas (whitelists): Una lista blanca es un mecanismo que permite explícitamente a algunas entidades identificadas acceder a un determinado privilegio o servicio, es decir, es una lista de cosas permitidas cuando todo está denegado por defecto (93).

Logstash: Es un *pipeline* de procesamiento de datos gratuito y abierto del lado del servidor que ingesta datos de una multitud de fuentes, los transforma y los reenvía (94).

Mecanismo de backoff: En referencia a Logstash, mecanismo por el cual Logstash regula el flujo de datos enviado a Elasticsearch según este último pueda procesarlos e indexarlos.

Mitre Att&ck: Es una base de conocimientos de acceso global sobre las tácticas y técnicas de los cibercriminales basada en observaciones del mundo real. La base de conocimientos ATT&CK se utiliza como base para el desarrollo de modelos y metodologías de amenazas específicas en el sector privado, en el gobierno y en la comunidad de productos y servicios de ciberseguridad (95).

Netcat: Netcat es una herramienta de línea de comandos que sirve para escribir y leer datos en la red. Para la transmisión de datos, Netcat usa los protocolos de red TCP/IP y UDP (96).

No eXecute (NX): Técnica de seguridad relacionada con la ejecución de código en memoria que previene que una aplicación o servicio se ejecute desde una región de memoria no ejecutable (65).

Payload: Es la parte del código del malware que realiza la acción maliciosa en el sistema (97).

PCI-DSS: Es un estándar de seguridad propietario para administrar la información de tarjetas de crédito, débito, ATM, etc (2).

Pipeline: En referencia a Logstash, los *pipelines* permiten realizar transformaciones comunes en registros antes de reenviarlos a Elasticsearch. Por ejemplo, se pueden utilizar los *pipelines* para eliminar campos, extraer valores del texto y enriquecer registros (98).

Política de control del ciclo de vida de los índices (ILM): Permite gestionar automáticamente los índices según los requisitos de rendimiento, resistencia y retención de los que se disponga (99).

Public Key Infrastructure (PKI): Es un sistema de recursos, políticas y servicios que da soporte al uso del cifrado de claves públicas para autenticar a las partes que participan en una transacción (100).

Qradar: Es un SIEM de IBM que ayuda a los equipos de seguridad a detectar y priorizar con precisión las amenazas en toda la empresa. Proporciona información inteligente que permite a los equipos responder rápidamente para reducir el impacto de los incidentes. Está disponible en las instalaciones y en un entorno de nube (4).

Redis: Es un servicio de colas de mensajería utilizado en Elastic para almacenar los registros temporalmente en momentos donde Logstash o Elasticsearch no pueden absorber todo el tráfico de registros.

Red Host-Only: Las máquinas virtuales pueden hablar entre sí y con el host como si estuvieran conectadas a través de un conmutador Ethernet físico, no es necesario que haya una interfaz de red física, y las máquinas virtuales no pueden conectar con el exterior, ya que no están conectadas a una interfaz de red física (101).

Red NAT: Una máquina virtual con NAT activado actúa como un ordenador real que se conecta a Internet a través de un router (102).

Reglamento General de Protección de Datos (GDPR): Es la ley de privacidad y seguridad más dura del mundo. Aunque fue redactado y aprobado por la Unión Europea (UE), impone obligaciones a las organizaciones de cualquier lugar, siempre que se dirijan o recojan datos relacionados con personas de la UE. El reglamento entró en vigor el 25 de mayo de 2018 (103).

SAML: Es una forma estandarizada de indicar a las aplicaciones y servicios externos que un usuario es quien dice ser. SAML hace posible la tecnología de inicio de sesión único (SSO) al ofrecer una manera de autenticar a un usuario una vez y luego comunicar esa autenticación a múltiples aplicaciones (104).

Security Orchestration, Automation and Response (SOAR): Ayuda a coordinar, ejecutar y automatizar tareas entre personas y herramientas, todo dentro de una única plataforma. Esto permite a las organizaciones no solo responder rápidamente a los ataques de ciberseguridad, sino también observar, comprender y prevenir futuros incidentes, mejorando así su postura general de seguridad (105).

Servicios de seguridad gestionados (MSSP): Ofrece servicios de seguridad de red a una organización. Como tercero, un MSSP puede aliviar la tensión de los equipos de TI (106).

Shard: En referencia a la Elastic, un shard es la unidad en la que Elasticsearch distribuye los datos en el clúster (107).

Sistema de detección de intrusiones (IDS): Es una aplicación usada para detectar accesos no autorizados a un ordenador o a una red, es decir, son sistemas que monitorizan el tráfico entrante y lo cotejan con una base de datos actualizada de firmas de ataque conocidas. Ante cualquier actividad sospechosa, emiten una alerta (108).

Sistema de monitorización de actividades de bases de datos (DAM): Es un conjunto de herramientas que permiten identificar y reportar un comportamiento fraudulento, ilegal o no deseado con el mínimo impacto en las operaciones de usuario y la productividad (109).

Sistema de prevención de intrusiones (IPS): Es un software que se utiliza para proteger a los sistemas de ataques e intrusiones. Su actuación es preventiva. Estos sistemas llevan a cabo un análisis en tiempo real de las conexiones y los protocolos para determinar si se está produciendo o se va a producir un incidente, identificando ataques según patrones, anomalías o comportamientos sospechosos (108).

Sistema de tickets: Es una herramienta que permite el acceso a una cantidad ilimitada de solicitudes e incidentes de los clientes, provenientes desde diferentes canales, para ser gestionados a través de una única interfaz (110).

Slow DoS: Es una categoría específica de ataques de denegación de servicio que hace uso de una tasa de ancho de banda baja para lograr su objetivo (111).

Snapshot: En referencia a la Elastic, es una copia de seguridad de un clúster de Elasticsearch en funcionamiento. Puedes utilizar los *snapshots* para realizar copias de seguridad periódicas de un clúster sin tiempo de inactividad, recuperar datos tras un borrado o un fallo de hardware, transferir datos entre clústeres o reducir tus costes de almacenamiento utilizando *snapshots* con capacidad de búsqueda en los niveles de datos *cool* y *frozen* (112).

Splunk: Es un SIEM que proporciona información para detectar y responder rápidamente a los ataques internos y externos y simplificar la gestión de las amenazas minimizando el riesgo. Ayuda a los equipos a obtener visibilidad e inteligencia de seguridad en toda la organización para la supervisión continua, la respuesta a incidentes, las operaciones del SOC y para ofrecer a los ejecutivos una ventana al riesgo empresarial (4).

Stack Smashing Protector (SSP): Es una técnica de seguridad que ayuda a detectar desbordamientos de búfer en la pila, abortando si un valor secreto en la pila (canario) se cambia (65).

Tecnología industrial (OT): La tecnología operativa (TO) consiste en utilizar el software y el hardware para controlar los equipos industriales, e incluye los sistemas especializados que se utilizan en los sectores de fabricación, energía, medicina y gestión de los edificios, entre otros (113).

User Entity Behavior Analytics (UEBA): Es una solución de ciberseguridad que utiliza algoritmos y *machine learning* para detectar anomalías en el comportamiento no sólo de los usuarios de una red corporativa, sino también de los routers, servidores y puntos finales de esa red (114).

Verdadero positivo: Es un resultado verdadero o correcto para un evento que se produce en la red. Mediante los verdaderos positivos, se identifican los eventos maliciosos reales. El SIEM tiene que ser diseñado de tal manera que sólo produzca verdaderos positivos (4).

10. Bibliografía

1. **MUNIZ, Joseph, y otros.** *The Modern Security Operations Center.* s.l. : Addison-Wesley Professional, 2021. 9780135619773.

2. **EC-COUNCIL.** *Ethical Hacking and Countermeasures Version 11.* Albuquerque : EC-Council, 2020.

3. *On the effectiveness of NX, SSP, RenewSSP and ASLR against buffer overflows.* **MARCO GISBERT, Hector y RIPOLL RIPOLL, Ismael.** Valencia : s.n., 2014.

4. **EC-COUNCIL.** *Certified SOC Analyst.* Albuquerque : Professional Series, 2019.

5. **BUSSA, Toby , KAVANAGH, Kelly y COLLINS, John.** *Critical Capabilities for Security Information and Event Management.* s.l. : Gartner, 2021.

6. **LOGRHYTHM.** *Budgeting for a modern SIEM.* s.l. : LogRhythm.

7. **BUSSA, Toby, KAVANAGH, Kelly y COLLINS, John.** gartner.com. *Magic Quadrant for Security Information and Event Management.* [En línea] Gartner, 29 de Junio de 2021. [Citado el: 20 de Abril de 2022.] https://www.gartner.com/doc/reprints?id=1-26Q47L81&ct=210706&st=sb. G00467384.

8. **ELASTIC.** elastic.co. *Suscripciones del Elastic Stack.* [En línea] Elastic. [Citado el: 20 de Abril de 2022.] https://www.elastic.co/es/subscriptions.

9. *Implementación de un centro de operaciones de seguridad (SOC) de código abierto con elementos de red para sistemas industriales.* **MARTÍNEZ GÓMEZ, Mònica.** Valencia : UPV, 30 de Septiembre de 2021.

10. *Generación de ciberinteligencia con Splunk.* **GONZÁLEZ DE JUANA, Osmany.** Valencia : UPV, 23 de Julio de 2021.

11. *Implantación de Qradar en un entorno genérico multicliente para SOC.* **SÁNCHEZ SANZ, Alexis.** Valencia : UPV, 25 de Septiembre de 2019.

12. **IBM.** ibm.com. *¿Qué es SIEM?* [En línea] IBM. [Citado el: 2 de Mayo de 2022.] https://www.ibm.com/es-es/topics/siem.

13. **ELASTIC.** elastic.co. *Elastic Stack.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/elastic-stack/.

14. —. elastic.co. *Search. Observe. Protect.* [En línea] Elastic. [Citado el: 5 de Abril de 2022.] https://www.elastic.co/es/products/.

15. —. elastic.co. *What is Elasticsearch?* [En línea] Elastic. [Citado el: 5 de Abril de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html#elasticsearch-intro.

16. —. elastic.co. *Kibana - Your window into Elastic.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/guide/en/kibana/current/introduction.html#introduction.

17. —. elastic.co. *Logstash Introduction.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/guide/en/logstash/current/introduction.html.

18. —. elastic.co. *Structure of a pipeline.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/guide/en/logstash/current/configuration-file-structure.html.

19. —. elastic.co. *What are Beats?* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html.

20. —. elastic.co. *Auditbeat.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/beats/auditbeat.

21. —. elastic.co. *Filebeat.* [En línea] Elastic. [Citado el: 5 de Abril de 2022.] https://www.elastic.co/es/beats/filebeat.

22. —. elastic.co. *Heartbeat.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/beats/heartbeat.

23. —. elastic.co. *Metricbeat.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/beats/metricbeat.

24. —. elastic.co. *Packetbeat.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/beats/packetbeat.

25. —. elastic.co. *Winlogbeat.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/beats/winlogbeat.

26. —. elastic.co. *Functionbeat.* [En línea] Elastic. [Citado el: 5 de Mayo de 2022.] https://www.elastic.co/es/beats/functionbeat.

27. **IBM Security.** *Cost of a Data Breach Report.* s.l. : IBM, 2021.

28. **GUTNIKOV, Alexander, KURPREEV, Oleg y SHMELEV, Yaroslav.** securelist.com. *DDoS attacks in Q1 2022.* [En línea] Kaspersky, 25 de Abril de 2022. [Citado el: 23 de Mayo de 2022.] https://securelist.com/ddos-attacks-in-q1-2022/106358/.

29. **EXABEAM.** *Cybersecurity Professionals Salary, Skills and Stress Survey.* s.l. : Exabeam, 2019.

30. **SECURONIX.** securonix.com. *Training Courses.* [En línea] Securonix. [Citado el: 27 de Mayo de 2022.] https://www.securonix.com/services/training/.

31. **ELASTIC.** elastic.co. *Open doors with Elastic Certification.* [En línea] Elastic. [Citado el: 27 de Mayo de 2022.] https://www.elastic.co/es/training/certification.

32. **LOGRHYTHM.** logrhythm.com. *LogRhythm learning paths.* [En línea] LogRhythm. [Citado el: 27 de Mayo de 2022.] https://logrhythm.com/services/training/logrhythm-training/.

33. **WESTBERG, Eric.** elastic.co. *Elasticsearch Architecture Best Practices.* [En línea] Elastic. [Citado el: Junio de 6 de 2022.] https://www.elastic.co/pdf/architecture-best-practices.pdf.

34. **ELASTIC.** elastic.co. *Monitoring Overview.* [En línea] Elastic. [Citado el: 6 de Junio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/monitoring-overview.html.

35. **ELASTIC SUPPORT.** elastic.co. *Configuración de SSL, TLS y HTTPS para asegurar Elasticsearch, Kibana, Beats y Logstash.* [En línea] Elastic, 11 de Junio de 2019. [Citado el: 2 de Julio de 2022.] https://www.elastic.co/es/blog/configuring-ssl-tls-and-https-to-secure-elasticsearch-kibana-beats-and-logstash.

36. **ELASTIC.** elastic.co. *elasticsearch-certutil.* [En línea] Elastic. [Citado el: 2 de Julio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/certutil.html.

37. —. elastic.co. *Realms.* [En línea] Elastic. [Citado el: 2 de Julio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/realms.html.

38. —. elastic.co. *What is ECS?* [En línea] Elastic. [Citado el: Julio de 5 de 2022.] https://www.elastic.co/guide/en/ecs/current/ecs-reference.html.

39. —. elastic.co. *Install Elasticsearch with Debian Package.* [En línea] Elastic. [Citado el: 7 de Julio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html.

40. —. elastic.co. *Install Kibana with Debian package.* [En línea] Elastic. [Citado el: 7 de Julio de 2022.] https://www.elastic.co/guide/en/kibana/current/deb.html.

41. —. elastic.co. *Networking.* [En línea] Elastic. [Citado el: 7 de Julio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-network.html.

42. —. elastic.co. *Installing Logstash.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/logstash/current/installing-logstash.html.

43. —. elastic.co. *Multiple Pipelines.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/logstash/current/multiple-pipelines.html.

44. —. elastic.co. *Pipeline-to-pipeline communication.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/logstash/current/pipeline-to-pipeline.html.

45. —. elastic.co. *Secure your connection to Elasticsearch.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/logstash/current/ls-security.html.

46. —. elastic.co. *Secure communication with Logstash.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/configuring-ssl-logstash.html.

47. —. elastic.co. *Mutate filter plugin: add_field.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/logstash/current/plugins-filters-mutate.html#plugins-filters-mutate-add_field.

48. —. elastic.co. *Built-in users.* [En línea] Elastic. [Citado el: 10 de Julio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/built-in-users.html.

49. —. elastic.co. *Accessing event data and fields.* [En línea] Elastic. [Citado el: 7 de Agosto de 2022.] https://www.elastic.co/guide/en/logstash/current/event-dependent-configuration.html#conditionals.

50. —. elastic.co. *Repositories for APT and YUM.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/setup-repositories.html.

51. —. elastic.co. *Filebeat and systemd.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/running-with-systemd.html.

52. —. elastic.co. *Configure the Elasticsearch output.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/elasticsearch-output.html.

53. —. elastic.co. *Configure Kibana.* [En línea] ELastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/kibana/current/settings.html.

54. —. elastic.co. *Load Kibana dashboards.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/load-kibana-dashboards.html.

55. —. elastic.co. *Iptables module.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-module-iptables.html.

56. —. elastic.co. *Configure the Logstash output.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/logstash-output.html.

57. —. elastic.co. *Configure inputs.* [En línea] Elastic. [Citado el: 15 de Julio de 2022.] https://www.elastic.co/guide/en/beats/filebeat/current/configuration-filebeat-options.html.

58. **BEARCAT.** Askubuntu.com. *Redirect iptables logging to another logfile.* [En línea] Askubuntu, 21 de Junio de 2016. [Citado el: 15 de Julio de 2022.] https://askubuntu.com/questions/789516/redirect-iptables-logging-to-another-logfile.

59. **ELASTIC.** elastic.co. *Winlogbeat quick start: Installation and configuration.* [En línea] Elastic. [Citado el: 22 de Julio de 2022.] https://www.elastic.co/guide/en/beats/winlogbeat/current/winlogbeat-installation-configuration.html.

60. —. elastic.co. *Configure a lifecycle policy.* [En línea] Elastic. [Citado el: 29 de Julio de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/set-up-lifecycle-policy.html.

61. —. elastic.co. *Alerting and action settings in Kibana: General settings.* [En línea] Elastic. [Citado el: 29 de Julio de 2022.] https://www.elastic.co/guide/en/kibana/master/alert-action-settings-kb.html#general-alert-action-settings.

62. —. elastic.co. *Create a detection rule.* [En línea] Elastic. [Citado el: 3 de Agosto de 2022.] https://www.elastic.co/guide/en/security/current/rules-ui-create.html.

63. **RIPOLL RIPOLL, Ismael y MARCO GISBERT, Hector.** *Hacking Ético Tema1 - Introducción.* [PoliformaT] Valencia : UPV, 1 de Febrero de 2022.

64. **MICROSOFT.** microsoft.com. *Introducción a Active Directory Domain Services.* [En línea] Microsoft, 18 de Agosto de 2022. [Citado el: 3 de Septiembre de 2022.] https://docs.microsoft.com/es-es/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview.

65. **RIPOLL RIPOLL, Ismael y MARCO GISBERT, Hector.** *Hacking Ético Tema5 - Mitigación.* [PoliformaT] Valencia : UPV, 21 de Marzo de 2021.

66. **OFICINA DE SEGURIDAD DEL INTERNAUTA.** osi.es. *¿Qué son los ataques DoS y DDoS?* [En línea] INCIBE, 21 de Agosto de 2018. [Citado el: 21 de Septiembre de 2022.] https://www.osi.es/es/actualidad/blog/2018/08/21/que-son-los-ataques-dos-y-ddos.

67. **KASPERSKY.** kaspersky.es. *¿Qué es un ataque de fuerza bruta?* [En línea] Kaspersky. [Citado el: 3 de Septiembre de 2022.] https://www.kaspersky.es/resource-center/definitions/brute-force-attack.

68. **SSL.COM.** ssl.com. *¿Qué es una autoridad de certificación (CA)?* [En línea] 6 de Diciembre de 2021. [Citado el: 3 de Septiembre de 2022.] https://www.ssl.com/es/preguntas-frecuentes/%C2%BFQu%C3%A9-es-una-autoridad-de-certificaci%C3%B3n%3F/.

69. **FABRICA NACIONAL DE MONEDA Y TIMBRE.** fnmt.gob.es. *1028 - ¿Qué es una Autoridad de Registro?* [En línea] Real Casa de la Moneda. [Citado el: 3 de Septiembre de 2022.] https://www.sede.fnmt.gob.es/preguntas-frecuentes/otras-preguntas/-/asset_publisher/1RphW9IeUoAH/content/1028-que-es-una-autoridad-de-registro-?inheritRedirect=false.

70. **ELASTIC.** elastic.co. *Beats.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/beats/.

71. **SPLUNK.** splunk.com. *What Is a Network Operations Center (NOC)?* [En línea] Splunk. [Citado el: 3 de Septiembre de 2022.] https://www.splunk.com/en_us/data-insider/network-operations-center.html.

72. **ORACLE.** oracle.com. *¿Qué es un SOC?* [En línea] Oracle, 2022. [Citado el: 3 de Septiembre de 2022.] https://www.oracle.com/es/database/security/que-es-un-soc.html.

73. **INCIBE.** incibe.es. *CEO, CISO, CIO… ¿Roles en ciberseguridad?* [En línea] INCIBE, 30 de Noviembre de 2016. [Citado el: 3 de Septiembre de 2022.] https://www.incibe.es/protege-tu-empresa/blog/ceo-ciso-cio-roles-ciberseguridad.

74. **ELASTIC.** elastic.co. *Add and remove nodes in your cluster.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/add-elasticsearch-nodes.html.

75. —. elastic.co. *Data Streams.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/data-streams.html.

76. —. elastic.co. *Kibana Dashboard.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/kibana/kibana-dashboard.

77. —. elastic.co. *¿Qué es Elasticsearch?* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/what-is/elasticsearch.

78. —. elastic.co. *Application performance monitoring (APM).* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/observability/current/apm.html.

79. —. elastic.co. *Welcome Endgame: Bringing Endpoint Security to the Elastic Stack.* [En línea] ELastic, 5 de Junio de 2019. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/blog/endgame-joins-forces-with-elastic.

80. —. elastic.co. *Elastic Security overview.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/security/current/es-overview.html.

81. **YASAR, Kinza.** techtarget.com. *Elastic Stack (ELK Stack).* [En línea] Junio de 2022. [Citado el: 3 de Septiembre de 2022.] https://www.techtarget.com/searchitoperations/definition/Elastic-Stack.

82. **ELASTIC.** elastic.co. *Elastic Enterprise Search.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/enterprise-search/current/introduction.html.

83. **INGENS NETWORKS.** info.ingens-networks.com. *Diferencias entre Antivirus y Endpoints ¿Cuál elegir?* [En línea] Ingens Networks, 15 de Octubre de 2019. [Citado el: 3 de Septiembre de 2022.] https://info.ingens-networks.com/blog/diferencias-entre-antivirus-y-endpoints-c%C3%BAal-elegir.

84. **CISCO.** cisco.com. *¿Qué es un firewall?* [En línea] Cisco. [Citado el: 3 de Septiembre de 2022.] https://www.cisco.com/c/es_es/products/security/firewalls/what-is-a-firewall.html.

85. **SKOWRONSKI, Jason.** elastic.co. *El Agente de Elastic y Fleet facilitan la integración de tus sistemas en Elastic.* [En línea] Elastic, 3 de Agosto de 2021. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/blog/elastic-agent-and-fleet-make-it-easier-to-integrate-your-systems-with-elastic.

86. **LISA INSTITUTE.** lisainstitute.com. *Hacktivismo: definición, tipos, modus operandi y motivaciones.* [En línea] LISA Institute, 14 de Julio de 2021. [Citado el: 3 de Septiembre de 2022.] https://www.lisainstitute.com/blogs/blog/hacktivismo-definicion-tipos-modus-operandi-motivaciones.

87. **KASPERSKY.** kaspersky.com. *¿Qué es un honeypot?* [En línea] Kaspersky. [Citado el: 3 de Septiembre de 2022.] https://latam.kaspersky.com/resource-center/threats/what-is-a-honeypot.

88. **ORACLE.** oracle.com. *¿Qué es el IoT?* [En línea] Oracle. [Citado el: 3 de Septiembre de 2022.] https://www.oracle.com/es/internet-of-things/what-is-iot/.

89. **KHANDAVILLI, Preetipadma.** hevodata.com. *Apache Kafka Queue 101: Messaging Made Easy.* [En línea] HEVO, 31 de Enero de 2022. [Citado el: 3 de Septiembre de 2022.] https://hevodata.com/learn/kafka-queue/.

90. **WIKIPEDIA.** es.wikipedia.org. *Kerberos.* [En línea] Fundación Wikimedia, Inc, 30 de Abril de 2022. [Citado el: 3 de Septiembre de 2022.] https://es.wikipedia.org/wiki/Kerberos.

91. **ELASTIC.** elastic.co. *¿Qué es Kibana?* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/what-is/kibana.

92. **DE LUZ, Sergio.** redeszone.net. *Para qué sirve el protocolo LDAP y cómo funciona.* [En línea] Redes Zone, 15 de Mayo de 2022. [Citado el: 3 de Septiembre de 2022.] https://www.redeszone.net/tutoriales/servidores/que-es-ldap-funcionamiento/.

93. **WIKIPEDIA.** wikipedia.org. *Whitelist.* [En línea] Wikipedia, 3 de Agosto de 2022. [Citado el: 3 de Septiembre de 2022.] https://en.wikipedia.org/wiki/Whitelist.

94. **ELASTIC.** elastic.co. *Logstash.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/logstash/.

95. **MITRE.** mitre.org. *Mitre Attack.* [En línea] Mitre. [Citado el: 3 de Septiembre de 2022.] https://attack.mitre.org/.

96. **IONOS.** ionos.es. *¿Qué es Netcat y cómo funciona?* [En línea] IONOS, 2 de Octubre de 2020. [Citado el: 3 de Septiembre de 2022.] https://www.ionos.es/digitalguide/servidores/herramientas/netcat/.

97. **ACENS.** acens.com. *¿Qué es Payload?* [En línea] acens. [Citado el: 3 de Septiembre de 2022.] https://ayuda.acens.com/hc/es/articles/360018220377--Qu%C3%A9-es-Payload-.

98. **ELASTIC.** elastic.co. *Ingest pipelines.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/ingest.html.

99. —. elastic.co. *ILM: Manage the index lifecycle.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/index-lifecycle-management.html.

100. **IBM.** ibm.com. *Infraestructura de claves públicas (PKI).* [En línea] IBM, 20 de Abril de 2021. [Citado el: 3 de Septiembre de 2022.] https://www.ibm.com/docs/es/ibm-mq/7.5?topic=ssfksj-7-5-0-com-ibm-mq-sec-doc-q009900--htm.

101. **VIRTUAL BOX.** virtualbox.org. *Virtual Networking: Host-Only Networking.* [En línea] Oracle. [Citado el: 3 de Septiembre de 2022.] https://www.virtualbox.org/manual/ch06.html#network_hostonly.

102. —. oracle.com. *Network Address Translation (NAT).* [En línea] Oracle. [Citado el: 3 de Septiembre de 2022.] https://docs.oracle.com/en/virtualization/virtualbox/6.0/user/network_nat.html.

103. **WOLFORD, Ben.** gdpr.eu. *What is GDPR, the EU's new data protection law?* [En línea] Proton Technologies AG. [Citado el: 3 de Septiembre de 2022.] https://gdpr.eu/what-is-gdpr/.

104. **CLOUDFLARE.** cloudflare.com. *¿Qué es SAML? | Cómo funciona la autenticación SAML.* [En línea] Cloudflare. [Citado el: 3 de Septiembre de 2022.] https://www.cloudflare.com/es-es/learning/access-management/what-is-saml/.

105. **PALO ALTO.** paloaltonetworks.com. *What Is SOAR?* [En línea] Palo Alto. [Citado el: 3 de Septiembre de 2022.] https://www.paloaltonetworks.com/cyberpedia/what-is-soar.

106. **FORTINET.** fortinet.com. *Managed Security Service Provider (MSSP).* [En línea] Fortinet. [Citado el: 3 de Septiembre de 2022.] https://www.fortinet.com/resources/cyberglossary/what-is-mssp.

107. **DAHLQVIST, Christian.** elastic.co. *¿Cuántos shards debo tener en mi cluster de Elasticsearch?* [En línea] Elastic, 6 de Julio de 2022. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/es/blog/how-many-shards-should-i-have-in-my-elasticsearch-cluster.

108. **INCIBE.** incibe.es. *¿Qué son y para qué sirven los SIEM, IDS e IPS?* [En línea] INCIBE, 3 de Septiembre de 2020. [Citado el: 3 de Septiembre de 2022.] https://www.incibe.es/protege-tu-empresa/blog/son-y-sirven-los-siem-ids-e-ips.

109. **ZALTOR.** zaltor.com. *Protección y Auditoría de actividad en Bases de Datos (DAM).* [En línea] Zaltor. [Citado el: 3 de Septiembre de 2022.] https://zaltor.com/datasunrise/.

110. **DA SILVA, Douglas.** zendesk.com.mx. *(GUÍA) Sistema de tickets para optimizar el flujo de trabajo.* [En línea] Zendesk, 19 de Junio de 2020. [Citado el: 3 de Septiembre de 2022.] https://www.zendesk.com.mx/blog/sistema-de-gestion-de-tickets/.

111. **WIKIPEDIA.** wikipedia.org. *Slow DoS Attack.* [En línea] Wikipedia, 2 de Abril de 2022. [Citado el: 3 de Septiembre de 2022.] https://en.wikipedia.org/wiki/Slow_DoS_Attack.

112. **ELASTIC.** elastic.co. *Snapshot and restore.* [En línea] Elastic. [Citado el: 3 de Septiembre de 2022.] https://www.elastic.co/guide/en/elasticsearch/reference/current/snapshot-restore.html.

113. **RED HAT.** redhat.com. *¿Qué es la tecnología operativa (TO)?* [En línea] Red Hat, 19 de Agosto de 2021. [Citado el: 3 de Septiembre de 2022.] https://www.redhat.com/es/topics/edge/what-is-ot.

114. **FORTINET.** fortinet.com. *What is UEBA?* [En línea] Fortinet. [Citado el: 3 de Septiembre de 2022.] https://www.fortinet.com/resources/cyberglossary/what-is-ueba.

Anexo A: Objetivos de desarrollo sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	**Alto**	**Medio**	**Bajo**	**No Procede**
ODS 1. **Fin de la pobreza.**				**X**
ODS 2. **Hambre cero.**				**X**
ODS 3. **Salud y bienestar.**				**X**
ODS 4. **Educación de calidad.**				**X**
ODS 5. **Igualdad de género.**			**X**	
ODS 6. **Agua limpia y saneamiento.**				**X**
ODS 7. **Energía asequible y no contaminante.**				**X**
ODS 8. **Trabajo decente y crecimiento económico.**	**X**			
ODS 9. **Industria, innovación e infraestructuras.**				**X**
ODS 10. **Reducción de las desigualdades**.		**X**		
ODS 11. **Ciudades y comunidades sostenibles**.				**X**
ODS 12. **Producción y consumo responsables.**				**X**
ODS 13. **Acción por el clima.**				**X**
ODS 14. **Vida submarina.**				**X**
ODS 15. **Vida de ecosistemas terrestres.**				**X**
ODS 16. **Paz, justicia e instituciones sólidas.**	**X**			
ODS 17. **Alianzas para lograr objetivos.**				**X**

Tabla I.- Objetivos de desarrollo sostenible

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

El 25 de septiembre de 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.

De los objetivos expuestos en la Tabla I, el proyecto de este TFG se relaciona con los siguientes:

- **Paz, justicia e instituciones sólidas,** ya que la solución propuesta en este TFG ofrece una tecnología capaz de brindar seguridad digital avanzada tanto a individuos como a

organizaciones. El ciberespacio está actualmente en guerra como se ha expuesto en el capítulo 4.1. Problemas de seguridad, y la implementación de un SOC ayuda a naciones y organizaciones a mejorar su capacidad defensiva e instaurar la paz. Además, como se explica en este TFG, uno de los principales servicios de un Centro de Operaciones de Seguridad es el análisis forense, que apoya y ayuda a los organismos de derecho y seguridad nacionales e internacionales a impartir justicia en el mundo digital.

- **Trabajo decente y crecimiento económico,** puesto que los servicios de SOC apoyan la modernización tecnológica de los países creando empleos seguros y de calidad debido a la gran demanda de trabajadores en este sector actualmente, es más, el sector de la ciberseguridad y de la informática en general reduce la proporción de jóvenes desempleados, ya que estos entienden y se adaptan mejor a las nuevas tecnologías. También, con este proyecto se fomenta la formalización y el crecimiento de las microempresas y las pequeñas y medianas empresas brindándoles la oportunidad de cubrir los requisitos de seguridad mínimos a un precio razonable.

- **Reducción de las desigualdades,** pues la solución implementada se puede llevar a la práctica de forma totalmente gratuita, como se ha visto en el entorno de pruebas, con un ordenador de gama baja, ya que como se ha mostrado a lo largo del proyecto, toda la arquitectura del *Elastic Stack* se puede desplegar en una sola máquina. Así pues, se ofrece la oportunidad a empresas, organizaciones e individuos de monitorizar su propio entorno con unos recursos mínimos.

- **Igualdad de género,** a causa de la gran desigualdad de género experimentada actualmente en el sector informático, cualquier proyecto de divulgación de información dentro del sector, que pueda resultar interesante tanto para mujeres como a hombres, resulta en una pequeña contribución a reducir la brecha de género.

Anexo B: Configuración de elasticsearch.yml

/etc/elasticsearch/elasticsearch.yml

```
# ======================== Elasticsearch Configuration =========================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
#cluster.name: my-application
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
#node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ----------------------------------- Paths ------------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ----------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 10.0.0.4
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
#http.port: 9200
#
# For more information, consult the network module documentation.
#
# --------------------------------- Discovery ----------------------------------
#
```

```
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
#
#  For  more  information,  consult  the  discovery  and  cluster  formation  module
documentation.
#
# ---------------------------------- Readiness -----------------------------------
#
# Enable an unauthenticated TCP readiness endpoint on localhost
#
#readiness.port: 9399
#
# ---------------------------------- Various -----------------------------------
#
# Allow wildcard deletion of indices:
#
#action.destructive_requires_name: false

#----------------------- BEGIN SECURITY AUTO CONFIGURATION -----------------------
#
# The following settings, TLS certificates, and keys have been automatically
# generated to configure Elasticsearch security features on 15-05-2022 14:04:17
#
# --------------------------------------------------------------------------------

# Enable security features
xpack.security.enabled: true

xpack.security.enrollment.enabled: true

#  Enable  encryption  for  HTTP  API  client  connections,  such  as  Kibana,  Logstash,  and
Agents
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12

# Enable encryption and mutual authentication between cluster nodes
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
cluster.initial_master_nodes: ["elastic-VM"]

# Allow HTTP API connections from anywhere
# Connections are encrypted and require user authentication
http.host: 0.0.0.0

# Allow other nodes to join the cluster from anywhere
# Connections are encrypted and mutually authenticated
#transport.host: 0.0.0.0

#----------------------- END SECURITY AUTO CONFIGURATION -------------------------
```

Anexo C: Configuración de kibana.yml

/etc/kibana/kibana.yml

```
# For more configuration options see the configuration guide for Kibana in
# https://www.elastic.co/guide/index.html

# ================== System: Kibana Server ==================
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "10.0.0.4"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy.
# Use the `server.rewriteBasePath` setting to tell Kibana if it should remove the basePath
# from requests it receives, and to prevent a deprecation warning at startup.
# This setting cannot end in a slash.
#server.basePath: ""

# Specifies whether Kibana should rewrite requests that are prefixed with
# `server.basePath` or require that they are rewritten by your reverse proxy.
# Defaults to `false`.
#server.rewriteBasePath: false

# Specifies the public URL at which Kibana is available for end users. If
# `server.basePath` is configured this URL should end with the same basePath.
#server.publicBaseUrl: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayload: 1048576

# The Kibana server's name. This is used for display purposes.
#server.name: "your-hostname"

# ================== System: Kibana Server (Optional) ==================
# Enables SSL and paths to the PEM-format SSL certificate and SSL key files, respectively.
# These settings enable SSL for outgoing requests from the Kibana server to the browser.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# ================== System: Elasticsearch ==================
# The URLs of the Elasticsearch instances to use for all your queries.
#elasticsearch.hosts: ["http://localhost:9200"]

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
#elasticsearch.username: "kibana_system"
#elasticsearch.password: "pass"

# Kibana can also authenticate to Elasticsearch via "service account tokens".
# Service account tokens are Bearer style tokens that replace the traditional username/password based configuration.
# Use this token instead of a username/password.
# elasticsearch.serviceAccountToken: "my_token"

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the value of
```

```
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500

# Time in milliseconds to wait for responses from the back end or Elasticsearch. This
value
# must be a positive integer.
#elasticsearch.requestTimeout: 30000

# The maximum number of sockets that can be used for communications with elasticsearch.
# Defaults to `Infinity`.
#elasticsearch.maxSockets: 1024

# Specifies whether Kibana should use compression for communications with elasticsearch
# Defaults to `false`.
#elasticsearch.compression: false

# List of Kibana client-side headers to send to Elasticsearch. To send *no* client-side
# headers, set this value to [] (an empty list).
#elasticsearch.requestHeadersWhitelist: [ authorization ]

# Header names and values that are sent to Elasticsearch. Any custom headers cannot be
overwritten
#  by client-side headers, regardless of the elasticsearch.requestHeadersWhitelist
configuration.
#elasticsearch.customHeaders: {}

# Time in milliseconds for Elasticsearch to wait for responses from shards. Set to 0 to
disable.
#elasticsearch.shardTimeout: 30000

# ================== System: Elasticsearch (Optional) ==================
#  These files are used to verify the identity of Kibana to Elasticsearch and are
required when
# xpack.security.http.ssl.client_authentication in Elasticsearch is set to required.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key

# Enables you to specify a path to the PEM file for the certificate
# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# To disregard the validity of SSL certificates, change this setting's value to 'none'.
#elasticsearch.ssl.verificationMode: full

# ================== System: Logging ==================
# Set the value of this setting to off to suppress all logging output, or to debug to
log everything. Defaults to 'info'
#logging.root.level: debug

# Enables you to specify a file where Kibana stores log output.
logging:
  appenders:
    file:
      type: file
      fileName: /var/log/kibana/kibana.log
      layout:
        type: json
  root:
    appenders:
      - default
      - file
#  layout:
#    type: json

# Logs queries sent to Elasticsearch.
#logging.loggers:
#  - name: elasticsearch.query
#    level: debug

# Logs http responses.
#logging.loggers:
#  - name: http.server.response
#    level: debug

# Logs system usage information.
#logging.loggers:
#  - name: metrics.ops
```

```
#    level: debug

# ================== System: Other ==================
# The path where Kibana stores persistent data not saved in Elasticsearch. Defaults to
data
#path.data: data

# Specifies the path where Kibana creates the process ID file.
pid.file: /run/kibana/kibana.pid

# Set the interval in milliseconds to sample system and process performance
# metrics. Minimum is 100ms. Defaults to 5000ms.
#ops.interval: 5000

# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English (default) "en", Chinese "zh-CN",
Japanese "ja-JP", French "fr-FR".
#i18n.locale: "en"

# ================== Frequently used (Optional)==================

# ================== Saved Objects: Migrations ==================
# Saved object migrations run at startup. If you run into migration-related issues, you
might need to adjust these settings.

# The number of documents migrated at a time.
#   If   Kibana   can't   start   up   or   upgrade   due   to   an   Elasticsearch
`circuit_breaking_exception`,
# use a smaller batchSize value to reduce the memory pressure. Defaults to 1000 objects
per batch.
#migrations.batchSize: 1000

# The maximum payload size for indexing batches of upgraded saved objects.
# To avoid migrations failing due to a 413 Request Entity Too Large response from
Elasticsearch.
#  This  value  should  be  lower  than  or  equal  to  your  Elasticsearch  cluster's
`http.max_content_length`
# configuration option. Default: 100mb
#migrations.maxBatchSizeBytes: 100mb

# The number of times to retry temporary migration failures. Increase the setting
# if migrations fail frequently with a message such as `Unable to complete the [...]
step after
# 15 attempts, terminating`. Defaults to 15
#migrations.retryAttempts: 15

# ================== Search Autocomplete ==================
# Time in milliseconds to wait for autocomplete suggestions from Elasticsearch.
# This value must be a whole number greater than zero. Defaults to 1000ms
#data.autocomplete.valueSuggestions.timeout: 1000

# Maximum number of documents loaded by each shard to generate autocomplete suggestions.
# This value must be a whole number greater than zero. Defaults to 100_000
#data.autocomplete.valueSuggestions.terminateAfter: 100000


# This section was automatically generated during setup.
elasticsearch.hosts: ['https://10.0.0.4:9200']
elasticsearch.serviceAccountToken:
AAEAAWVsYXN0aWMva2liYW5hL2Vucm9sbC1wcm9jZXNzLXRva2VuLTE2NTI2MjY2NDAzMzQ6ME56SzI0MWFSZkdM
QmVVbGxUaEZ6UQ

elasticsearch.ssl.certificateAuthorities: [/var/lib/kibana/ca_1652626641367.crt]
xpack.fleet.outputs:  [{id:  fleet-default-output,  name:  default,  is_default:  true,
is_default_monitoring:  true,  type:  elasticsearch,  hosts:  ['https://10.0.0.4:9200'],
ca_trusted_fingerprint:
a3a750e953777c0669626cd895516827d88cf94e0da09d7bdd3198e4104f71b6}]

xpack.encryptedSavedObjects.encryptionKey: d7ae816f6d46c09248b035d783551b1b

monitoring.kibana.collection.enabled: false
```

Anexo D: Configuración de filebeat.yml

Filebeat en Elastic-VM:

/etc/filebeat/filebeat.yml

```
###################### Filebeat Configuration Example #########################

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

# For more available modules and options, please see the filebeat.reference.yml sample
# configuration file.

# ============================== Filebeat inputs ===============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: my-filestream-id

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/kibana/kibana.log
    #- c:\programdata\elasticsearch\logs\*

  # Exclude lines. A list of regular expressions to match. It drops the lines that are
  # matching any regular expression from the list.
  #exclude_lines: ['^DBG']

  # Include lines. A list of regular expressions to match. It exports the lines that
are
  # matching any regular expression from the list.
  #include_lines: ['^ERR', '^WARN']

  # Exclude files. A list of regular expressions to match. Filebeat drops the files
that
  # are matching any regular expression from the list. By default, no files are
dropped.
  #prospector.scanner.exclude_files: ['.gz$']

  # Optional additional fields. These fields can be freely picked
  # to add additional information to the crawled log files for filtering
  #fields:
  #  level: debug
  #  review: 1

# ============================== Filebeat modules ==============================

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  reload.enabled: false
```

```
  # Period on which files under path should be checked for changes
  #reload.period: 10s

# ======================== Elasticsearch template setting ========================

setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false


# ================================== General ===================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging

# ================================== Dashboards =================================
# These settings control loading the sample dashboards to the Kibana index. Loading
# the dashboards is disabled by default and can be enabled either by setting the
# options here or by using the `setup` command.
#setup.dashboards.enabled: false

# The URL from where to download the dashboards archive. By default this URL
# has a value which is computed based on the Beat name and version. For released
# versions, this URL points to the dashboard archive on the artifacts.elastic.co
# website.
#setup.dashboards.url:

# =================================== Kibana ===================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  #   In   case   you   specify   and   additional   path,   the   scheme   is   required:
http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "localhost:5601"

  # Kibana Space ID
  # ID of the Kibana Space into which the dashboards should be loaded. By default,
  # the Default Space will be used.
  #space.id:

# ============================== Elastic Cloud ================================

#    These    settings    simplify    using    Filebeat    with    the    Elastic    Cloud
(https://cloud.elastic.co/).

# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:

# The cloud.auth setting overwrites the `output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth:

# ================================== Outputs ===================================

# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ----------------------------
#output.elasticsearch:
```

```
  # Array of hosts to connect to.
#  hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  #username: "elastic"
  #password: "changeme"

# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["10.0.0.5:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  ssl.certificate_authorities: ["/usr/share/elasticsearch/http_ca_logstash.crt"]

  # Certificate for SSL client authentication
  ssl.certificate: "/usr/share/elasticsearch/filebeat-cert.crt"

  # Client Certificate Key
  ssl.key: "/usr/share/elasticsearch/filebeat-cert.key"

# ================================= Processors =================================
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~

# ================================== Logging ===================================

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
#logging.level: debug

# At debug level, you can selectively enable logging only for some components.
# To enable all selectors use ["*"]. Examples of other selectors are "beat",
# "publisher", "service".
#logging.selectors: ["*"]

# ============================= X-Pack Monitoring ==============================
# Filebeat can export internal metrics to a central Elasticsearch monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.  The
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
#monitoring.enabled: false

# Sets the UUID of the Elasticsearch cluster under which monitoring data for this
# Filebeat instance will appear in the Stack Monitoring UI. If output.elasticsearch
#  is enabled, the UUID is derived from the Elasticsearch cluster referenced by
output.elasticsearch.
#monitoring.cluster_uuid:

# Uncomment to send the metrics to Elasticsearch. Most settings from the
# Elasticsearch output are accepted here as well.
# Note that the settings should point to your Elasticsearch *monitoring* cluster.
# Any setting that is not set is automatically inherited from the Elasticsearch
# output configuration, so if you have the Elasticsearch output configured such
# that it is pointing to your Elasticsearch monitoring cluster, you can simply
# uncomment the following line.
#monitoring.elasticsearch:

# ============================== Instrumentation ===============================

# Instrumentation support for the filebeat.
#instrumentation:
    # Set to true to enable instrumentation of filebeat.
    #enabled: false

    # Environment in which filebeat is running on (eg: staging, production, etc.)
    #environment: ""
```

```
    # APM Server hosts to report instrumentation results to.
    #hosts:
    #   - http://localhost:8200

    # API Key for the APM Server(s).
    # If api_key is set then secret_token will be ignored.
    #api_key:

    # Secret token for the APM Server(s).
    #secret_token:


# ================================== Migration ==================================

# This allows to enable 6.7 migration aliases
#migration.6_to_7.enabled: true
```

Filebeat en linux-VM:

/etc/filebeat/filebeat.yml

```
###################### Filebeat Configuration Example #########################

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

# For more available modules and options, please see the filebeat.reference.yml sample
# configuration file.

# ============================== Filebeat inputs ===============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filestream is an input for collecting log messages from files.
- type: filestream

  # Unique ID among all inputs, an ID is required.
  id: first_filebeat

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/iptables.log
    - /var/log/auth.log

  # Exclude lines. A list of regular expressions to match. It drops the lines that are
  # matching any regular expression from the list.
  #exclude_lines: ['^DBG']

  # Include lines. A list of regular expressions to match. It exports the lines that
are
  # matching any regular expression from the list.
  #include_lines: ['^ERR', '^WARN']

  # Exclude files. A list of regular expressions to match. Filebeat drops the files
that
  # are matching any regular expression from the list. By default, no files are
dropped.
  #prospector.scanner.exclude_files: ['.gz$']

  # Optional additional fields. These fields can be freely picked
  # to add additional information to the crawled log files for filtering
  #fields:
```

```
  #  level: debug
  #  review: 1

# ============================== Filebeat modules ==============================

filebeat.config.modules:
  # Glob pattern for configuration loading
  path: ${path.config}/modules.d/*.yml

  # Set to true to enable config reloading
  reload.enabled: false

  # Period on which files under path should be checked for changes
  #reload.period: 10s

# ======================= Elasticsearch template setting =======================

setup.template.settings:
  index.number_of_shards: 2
  #index.codec: best_compression
  #_source.enabled: false


# ================================== General ===================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging

# ================================= Dashboards =================================
# These settings control loading the sample dashboards to the Kibana index. Loading
# the dashboards is disabled by default and can be enabled either by setting the
# options here or by using the `setup` command.
#setup.dashboards.enabled: false

# The URL from where to download the dashboards archive. By default this URL
# has a value which is computed based on the Beat name and version. For released
# versions, this URL points to the dashboard archive on the artifacts.elastic.co
# website.
#setup.dashboards.url:

# =================================== Kibana ===================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.

setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  #   In   case   you   specify   and   additional   path,   the   scheme   is   required:
http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  host: "10.0.0.4:5601"

  # Kibana Space ID
  # ID of the Kibana Space into which the dashboards should be loaded. By default,
  # the Default Space will be used.
  #space.id:

# =============================== Outputs ======================================

# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ----------------------------

#output.elasticsearch:
  # Array of hosts to connect to.
```

```
#  hosts: ["https://10.0.0.4:9200"]

  # Protocol - either `http` (default) or `https`.
#  protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
#  username: "elastic"
#  password: "RrxQ3yerPDLfYja4ZqAD"
#  ssl:
#    enable: true
#                                                                ca_trusted_fingerprint:
a3a750e953777c0669626cd895516827d88cf94e0da09d7bdd3198e4104f71b6

# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["10.0.0.5:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  ssl.certificate_authorities: ["/etc/filebeat/shared/http_ca_logstash.crt"]

  # Certificate for SSL client authentication
  ssl.certificate: "/etc/filebeat/shared/filebeat-cert.crt"

  # Client Certificate Key
  ssl.key: "/etc/filebeat/shared/filebeat-cert.key"

# ================================= Processors =================================
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~

# ================================== Logging ===================================

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
#logging.level: debug

# At debug level, you can selectively enable logging only for some components.
# To enable all selectors use ["*"]. Examples of other selectors are "beat",
# "publisher", "service".
#logging.selectors: ["*"]

# ============================== Elastic Cloud ==============================

#    These    settings    simplify    using    Filebeat    with    the    Elastic    Cloud
(https://cloud.elastic.co/).

# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:

# The cloud.auth setting overwrites the `output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth:

# ============================== X-Pack Monitoring ==============================
# Filebeat can export internal metrics to a central Elasticsearch monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.  The
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
#monitoring.enabled: false

# Sets the UUID of the Elasticsearch cluster under which monitoring data for this
# Filebeat instance will appear in the Stack Monitoring UI. If output.elasticsearch
#  is  enabled,  the  UUID  is  derived  from  the  Elasticsearch  cluster  referenced  by
output.elasticsearch.
#monitoring.cluster_uuid:

# Uncomment to send the metrics to Elasticsearch. Most settings from the
```


```
# Elasticsearch output are accepted here as well.
# Note that the settings should point to your Elasticsearch *monitoring* cluster.
# Any setting that is not set is automatically inherited from the Elasticsearch
# output configuration, so if you have the Elasticsearch output configured such
# that it is pointing to your Elasticsearch monitoring cluster, you can simply
# uncomment the following line.
#monitoring.elasticsearch:

# ============================== Instrumentation ===============================

# Instrumentation support for the filebeat.
#instrumentation:
    # Set to true to enable instrumentation of filebeat.
    #enabled: false

    # Environment in which filebeat is running on (eg: staging, production, etc.)
    #environment: ""

    # APM Server hosts to report instrumentation results to.
    #hosts:
    #  - http://localhost:8200

    # API Key for the APM Server(s).
    # If api_key is set then secret_token will be ignored.
    #api_key:

    # Secret token for the APM Server(s).
    #secret_token:


# ================================= Migration ==================================

# This allows to enable 6.7 migration aliases
#migration.6_to_7.enabled: true
```

Anexo E: Configuración de winlogbeat.yml

C:\Program Files\Winlogbeat\winlogbeat.yml

```
###################### Winlogbeat Configuration Example #########################

# This file is an example configuration file highlighting only the most common
# options. The winlogbeat.reference.yml file from the same directory contains
# all the supported options with more comments. You can use it as a reference.
#
# You can find the full configuration reference here:
# https://www.elastic.co/guide/en/beats/winlogbeat/index.html

# ======================== Winlogbeat specific options ========================

# event_logs specifies a list of event logs to monitor as well as any
# accompanying options. The YAML data type of event_logs is a list of
# dictionaries.
#
# The supported keys are name, id, xml_query, tags, fields, fields_under_root,
# forwarded, ignore_older, level, event_id, provider, and include_xml.
# The xml_query key requires an id and must not be used with the name,
# ignore_older, level, event_id, or provider keys. Please visit the
# documentation for the complete details of each option.
# https://go.es.io/WinlogbeatConfig

winlogbeat.event_logs:
  - name: Application
    ignore_older: 72h

  - name: System

  - name: Security

#  - name: Setup

#  - name: Microsoft-Windows-Sysmon/Operational

#  - name: Windows PowerShell
#    event_id: 400, 403, 600, 800

#  - name: Microsoft-Windows-PowerShell/Operational
#    event_id: 4103, 4104, 4105, 4106

#  - name: ForwardedEvents
#    tags: [forwarded]

# ====================== Elasticsearch template settings ======================

#setup.template.settings:
#  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false


# ================================== General ===================================

# The name of the shipper that publishes the network data. It can be used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their own field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to the
# output.
#fields:
#  env: staging
```

```
# ================================== Dashboards =====================================
# These settings control loading the sample dashboards to the Kibana index. Loading
# the dashboards is disabled by default and can be enabled either by setting the
# options here or by using the `setup` command.
#setup.dashboards.enabled: false

# The URL from where to download the dashboards archive. By default this URL
# has a value which is computed based on the Beat name and version. For released
# versions, this URL points to the dashboard archive on the artifacts.elastic.co
# website.
#setup.dashboards.url:

# =================================== Kibana ===================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  #   In   case   you   specify   and   additional   path,   the   scheme   is   required:
http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "localhost:5601"

  # Kibana Space ID
  # ID of the Kibana Space into which the dashboards should be loaded. By default,
  # the Default Space will be used.
  #space.id:

# ============================== Elastic Cloud ==============================

#    These    settings    simplify    using    Winlogbeat    with    the    Elastic    Cloud
(https://cloud.elastic.co/).

# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:

# The cloud.auth setting overwrites the `output.elasticsearch.username` and
# `output.elasticsearch.password` settings. The format is `<user>:<pass>`.
#cloud.auth:

# ================================== Outputs ===================================

# Configure what output to use when sending the data collected by the beat.

# ---------------------------- Elasticsearch Output ----------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  #username: "elastic"
  #password: "changeme"

  # Pipeline to route events to security, sysmon, or powershell pipelines.
  #pipeline: "winlogbeat-%{[agent.version]}-routing"

# ------------------------------ Logstash Output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["192.168.56.106:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  ssl.certificate_authorities:                                          ['C:\Program
Files\Winlogbeat\certs\http_ca_logstash.crt']
  ssl.certificate: 'C:\Program Files\Winlogbeat\certs\winlogbeat-cert.crt'
  ssl.key: 'C:\Program Files\Winlogbeat\certs\winlogbeat-cert.key'
```

```
# ================================== Processors ===================================
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~

# ================================== Logging ===================================

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
logging.level: debug
logging.to_files: true
logging.files:
  path: C:\Program Files\Winlogbeat\logs

# At debug level, you can selectively enable logging only for some components.
# To enable all selectors use ["*"]. Examples of other selectors are "beat",
# "publisher", "service".
#logging.selectors: ["*"]

# ============================= X-Pack Monitoring ==============================
# Winlogbeat can export internal metrics to a central Elasticsearch monitoring
# cluster.  This requires xpack monitoring to be enabled in Elasticsearch.  The
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
#monitoring.enabled: false

# Sets the UUID of the Elasticsearch cluster under which monitoring data for this
# Winlogbeat instance will appear in the Stack Monitoring UI. If output.elasticsearch
#  is  enabled,  the  UUID  is  derived  from  the  Elasticsearch  cluster  referenced  by
output.elasticsearch.
#monitoring.cluster_uuid:

# Uncomment to send the metrics to Elasticsearch. Most settings from the
# Elasticsearch output are accepted here as well.
# Note that the settings should point to your Elasticsearch *monitoring* cluster.
# Any setting that is not set is automatically inherited from the Elasticsearch
# output configuration, so if you have the Elasticsearch output configured such
# that it is pointing to your Elasticsearch monitoring cluster, you can simply
# uncomment the following line.
#monitoring.elasticsearch:

# ============================== Instrumentation ===============================

# Instrumentation support for the winlogbeat.
#instrumentation:
    # Set to true to enable instrumentation of winlogbeat.
    #enabled: false

    # Environment in which winlogbeat is running on (eg: staging, production, etc.)
    #environment: ""

    # APM Server hosts to report instrumentation results to.
    #hosts:
    #  - http://localhost:8200

    # API Key for the APM Server(s).
    # If api_key is set then secret_token will be ignored.
    #api_key:

    # Secret token for the APM Server(s).
    #secret_token:


# ================================= Migration ==================================

# This allows to enable 6.7 migration aliases
#migration.6_to_7.enabled: true
```