Universitat Oberta de Catalunya

uoc.edu

Desplegar la herramienta "Zeek IDS" y su posterior explotación para el análisis de actividades sospechosas en la red

Alumno: Álvaro Pérez Cotillas
Plan de Estudios: Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)
Trabajo de fin de máster. Análisis de datos.

Director de TFM: Borja Guaita Pérez
Profesor/a responsable de la asignatura: Helena Rifà Pous

29/05/2020

Agradecimientos:

A mis padres, Carlos y Vicenta, por su apoyo y fe incondicional en mí, inculcándome los valores necesarios para tener éxito en esta vida.

A mi hermano Carlos, siempre ha sido un ejemplo a seguir.

A mi chica, Marina, por su apoyo y confianza en mí.

Nada de esto sería posible sin vosotros.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desplegar la herramienta "Zeek IDS" y su posterior explotación para el análisis de actividades sospechosas en la red*
Nombre del autor:	*Álvaro Pérez Cotillas*
Nombre del consultor/a:	*Borja Guaita Pérez*
Nombre del PRA:	*Helena Rifà Pous*
Fecha de entrega:	29/05/2020
Titulación:	Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)
Área del Trabajo Final:	*Análisis de datos*
Idioma del trabajo:	*Castellano*
Palabras clave	*IDS, Zeek, ELK*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La finalidad de este proyecto es implementar la herramienta Zeek IDS para el análisis de anomalías de seguridad en el tráfico de red, e integrarlo con la solución ELK (Elasticsearch, Logstash y Kibana), a través de las cuales se realizará el envío de las detecciones realizadas por el sistema de detección de intrusiones, así como el almacenado, indexado y representación de estos datos. Una vez se encuentren ambas tecnologías integradas en el entorno de trabajo preparado, se pretende incluir un sistema de alertas a través del cual, se generen notificaciones siempre y cuando se detecten en los datos generados por Zeek IDS conexiones hacia sitios web maliciosos de tipo botnet.

Además, se incluye a la funcionalidad de Zeek la integración del proyecto BZAR de Mitre ATT&CK, a través del cual, se amplía la información de las detecciones de Zeek, añadiendo la categorización del ataque de seguridad detectado.

Toda esta funcionalidad ha sido posible conseguirla a través de una metodología de trabajo de investigación de los productos y posterior aplicación de los conocimientos adquiridos a la práctica.

Finalmente, las conclusiones obtenidas son muy positivas, debido al grado de aprendizaje obtenido en este proyecto, así como la utilidad de la solución desarrollada, la cual podría ser aplicada en entornos reales de pequeñas y medianas empresas que deseen añadir una capa más de seguridad a sus entornos.

Abstract (in English, 250 words or less):

The purpose of this project is to implement the Zeek IDS tool for the analysis of security anomalies in network traffic, and to integrate it with the ELK solution (Elasticsearch, Logstash and Kibana), through which the detections made by the intrusion detection system will be sent, as well as the storage, indexing and representation of this data. Once both technologies are integrated into the prepared work environment, the intention is to include an alert system through which, notifications are generated whenever connections to malicious botnet-type websites are found in the data generated by Zeek IDS.

In addition to Zeek's functionality, the integration of Mitre ATT&CK's BZAR project have been made, through which, the information of Zeek's detections is extended, adding the categorization of the security attack which was detected.

All this functionality has been possible to achieve through a working methodology of product research and subsequent application of the knowledge acquired to practice.

Finally, the conclusions obtained are very positive, due to the degree of learning obtained in this project, as well as the usefulness of the solution developed, which could be applied in real environments, like small and medium enterprises that wish to add an extra layer of security to their environments.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

En el entorno digital en el que nos encontramos hoy en día, las brechas de seguridad tienen efectos muy graves para la reputación y situación económica de las entidades financieras que lo conforman. Los datos exactos de las consecuencias de los ataques de seguridad son difíciles de calcular de forma exacta debido a que muchas entidades tienden a ocultar la ocurrencia de ciber ataques para proteger sus intereses.

Según datos del IE Law School de 2018, la comisión europea declara que 4000 ataques de tipo ransomware ocurren todos los días, suponiendo esto un incremento del 300% con respecto al año 2015. Se estima el valor del cibercrimen en la actualidad a nivel mundial en unos 900.000 millones de euros anuales. [1]

Dadas estas estimaciones, son clara evidencia del auge de probabilidad de que una empresa sufra un ataque de seguridad informático, ya que el crecimiento de personas dedicadas a comprometer entidades ha crecido considerablemente en los últimos años. Sin embargo, la innovación tecnológica cada vez es mayor en empresas que tratan de combatir este tipo de amenazas, tomando medidas y adaptándose al nuevo entorno digital. Aunque, dada la evolución tecnológica y la aparición de numerosas vulnerabilidades desconocidas cada día, hace que sea muy difícil combatir por completo este tipo de amenazas.

Uno de los ataques que más ha evolucionado estos últimos años y son cada vez más frecuentes, son los llamados intrusiones en sistemas informáticos. Se conoce una intrusión como un conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso informático.

Los sistemas utilizados en la actualidad para cubrir esta creciente problemática de intrusiones en sistemas informáticos, son los llamados sistemas de detección de intrusiones o IDS. Este tipo de tecnologías suelen estar basados en unos sensores de red que se encargan de escuchar y analizar el tráfico de red (sniffer), siendo capaz de detectar anomalías que puedan dar indicios de posibles ataques de seguridad.

Los sistemas de detección de intrusos suelen ir integrados con un firewall, con el objetivo de dar una capa más de seguridad a la primera línea de seguridad existente en los entornos de producción reales. La integración de estas herramientas es de gran importancia ya que permiten detectar a tiempo posibles intrusiones no deseadas en el sistema, permitiendo afrontar los problemas de seguridad antes de que sea demasiado tarde y se provoquen, por ejemplo, pérdidas de datos o caídas de servicio, que puedan provocar impactos reputacionales y económicos en los entornos afectados.

Sin embargo, suele existir un problema en la detección de intrusiones mediante estas herramientas, el cual consiste en el tiempo que transcurre desde que una vulnerabilidad es descubierta, hasta que la empresa de ese software comienza a preparar los parches de seguridad oportunos. Durante ese rango de tiempo, existe un foco de ataque importante, ya que la aparición de exploits suele ser más rápido que la creación del paquete que solucione dicha vulnerabilidad. Por estas razones, es necesario disponer de un sistema que disminuya este rango de tiempo y detecte automáticamente la existencia de nuevos ataques de forma fiable, de ahí la gran aceptación que tiene hoy en día un IDS como pieza fundamental de la infraestructura de seguridad de una organización. [2]

Mediante este proyecto, se pretende desarrollar una solución de detección de intrusiones a través de Zeek IDS, y su integración con la herramienta ELK, a través de la cual se pueda realizar una representación y análisis de datos, además de la creación de un sistema de alertas cuando se detecten conexiones sospechosas sobre sitios conocidos de botnets, de tipo Command and Control.

Todas las tecnologías que componen esta solución son Open Source, por lo que, mediante esta solución se pretende dar una visión, a aquellas pequeñas y medianas empresas que quizá no dispongan de todos los recursos económicos posibles para implementar sistemas de detección de fabricantes más robustos. De esta forma, pueden implementar la solución llevada a cabo en este proyecto de forma muy sencilla, y de esta forma, disponer de unos mecanismos de seguridad básicos en su infraestructura que les permitan conocer a tiempo un ataque informático de estas características.

1.2 Objetivos del Trabajo

El principal objetivo que se pretende conseguir con la realización de este trabajo es el de implementar una solución de sistema de detección de intrusiones mediante la herramienta Zeek IDS, y su integración con la herramienta ELK que realice la función de “SIEM”, mediante el almacenamiento de eventos de seguridad y su correlación con feeds de reputaciones de sitios conocidos de botnet, creando un sistema de alertas.

A partir del objetivo principal del proyecto, se identifican una serie de subobjetivos, a través de los cuales se pretende lograr el objetivo final mencionado.

- Definir un plan de trabajo que permita llevar a cabo el proyecto a tiempo y consiguiendo los objetivos definidos.
- Seguir una metodología de trabajo que se adapte al plan de trabajo definido.
- Comprender el funcionamiento de un sistema de detección de intrusiones.
- Estudiar el funcionamiento e integración de Zeek IDS en un entorno de pruebas.
- Conocer e investigar los patrones de eventos sospechosos de seguridad que se suelen dar en la red.

- Construir un entorno de trabajo adecuado, en el cual se realizará el desarrollo e implementación de la herramienta, como las pruebas de esta.
- Crear reglas de detección de eventos sospechosos de seguridad en Zeek IDS.
- Estudiar el funcionamiento e integración de ELK en un entorno de pruebas.
- Crear graficas que permitan analizar anomalías de seguridad en el tráfico de red mediante ELK.
- Investigar sobre fuentes de datos fiables para conocer los indicadores de compromiso (IOCs) sobre malware/botnet que existen en la actualidad.
- Crear un sistema de alertado en ELK mediante correlación con un feed de reputaciones de elementos maliciosos.
- Realizar una documentación sobre la solución implementada que transmita como ha ido evolucionando el proyecto y como este se ha ido adaptando a la solución final esperada.

1.3 Enfoque y método seguido

La estrategia o método seguido para llevar a cabo este proyecto será el de implementar y desarrollar un nuevo producto que cumpla la funcionalidad descrita anteriormente.

Este enfoque es el que mejor se adapta a la solución planteada, y permite la consecución de los objetivos mencionados en el apartado anterior. El hecho de partir desde cero y crear un proyecto nuevo, me va a permitir, conocer cómo funcionan los sistemas de detección de intrusos y en concreto, las tecnologías Zeek IDS y ELK.

Dado que no había trabajado previamente con ninguna de estas tecnologías, la única forma de entender el funcionamiento y aprender a configurarlas y utilizarlas, es comenzando un proyecto nuevo, pasando por una fase de investigación y estudio de las herramientas, pasando posteriormente, a la fase de implementación y desarrollo del producto.

Por esta razón, no se ha enfocado este proyecto a crear una solución a partir de algún proyecto ya existente, ya que de esta forma, no me permitiría conocer de forma adecuada el potencial y funcionalidad de las tecnologías con las que voy a trabajar, por lo que el enfoque de tomar este proyecto con la finalidad de aprender y crear una solución útil en el ámbito de la seguridad informática es el más adecuado para cumplir los objetivos propuestos.

1.4 Planificación del Trabajo

Para poder llevar a cabo el proyecto y cumplir con los objetivos definidos anteriormente, se requiere el uso de una serie de tecnologías y recursos.

Entre los principales recursos materiales identificados para llevar a cabo el proyecto:

- Portátil de trabajo que cumplirá el rol de host anfitrión donde se prepare y configure el entorno de trabajo para este proyecto, así como la estación de trabajo dónde se documente y realice la memoria del proyecto.
- Documentación y recursos bibliográficos para el estudio y aprendizaje de las diferentes tecnologías, que se mencionarán en el apartado correspondiente.

Las tecnologías principales con las que se va a trabajar en este proyecto son las siguientes:

- Entornos de virtualización como Virtual Box.
- Herramienta Zeek IDS.
- Herramienta ELK (Elasticsearch, Logstash y Kibana)

Por tanto, en base a los diferentes recursos, y las tecnologías mencionadas, surge la necesidad de elaborar un plan de trabajo, mediante el cual se organicen las diferentes tareas y tiempos estimados de elaboración de estas, que permitan abordar el proyecto de una forma adecuada.

Este plan de trabajo está formado por las siguientes fases/entregas del proyecto:

Entrega 1 (19/02/2020 - 03/03/2020)
Las tareas planificadas para esta primera entrega son:

- Investigación Zeek IDS
- Investigación sobre ELK
- Identificación de objetivos, alcance y metodología
- Planificación de trabajo

Entrega 2 (04/03/2020 - 31/03/2020)
Las tareas planificadas para la segunda fase del proyecto:

- Estudio de tecnologías Zeek IDS y ELK
- Planificación de entorno de trabajo
- Configuración de entorno de trabajo
- Documentar fundamentos teóricos y tecnologías utilizadas.
- Documentar estado del arte del proyecto.
- Documentar la planificación y configuraciones sobre el entorno de trabajo realizadas
- Instalación de herramienta Zeek IDS
- Instalación de herramienta ELK
- Documentar el proceso de instalación seguido para las tecnologías

Entrega 3 (01/04/2020 - 28/04/2020)
Las tareas planificadas para la tercera fase del proyecto:

- Planificar casos de uso de seguridad de prueba
- Documentar la planificación de casos de uso de prueba
- Configurar reglas de detección en Zeek IDS
- Configurar Logstash el envío de eventos a ELK

- Configurar almacenamiento de datos y representación en ELK
- Integrar listas de reputación para crear alertado en ELK
- Documentar todas las configuraciones de detección, almacén y representación de datos

Entrega 4 (29/04/2020 - 02/06/2020)
Las tareas planificadas para la cuarta entrega son:

- Ejecución de casos de uso de prueba. Análisis de resultados
- Revisión y mejora de reglas de detección.
- Documentar los resultados obtenidos en la fase de prueba y muestra de evidencias.
- Documentar conclusión y bibliografía del trabajo. Revisión general de memoria
- Entregar memoria final

Entrega 5 (03/06/2020 - 09/06/2020)
Las tareas planificadas para la última entrega del proyecto:

- Entregar video de presentación final de TFM

Este plan de trabajo está plasmado en el siguiente diagrama de Gantt, el cual ha sido elaborado tomando de referencia los tiempos de entrega definidos en el aula virtual del proyecto.

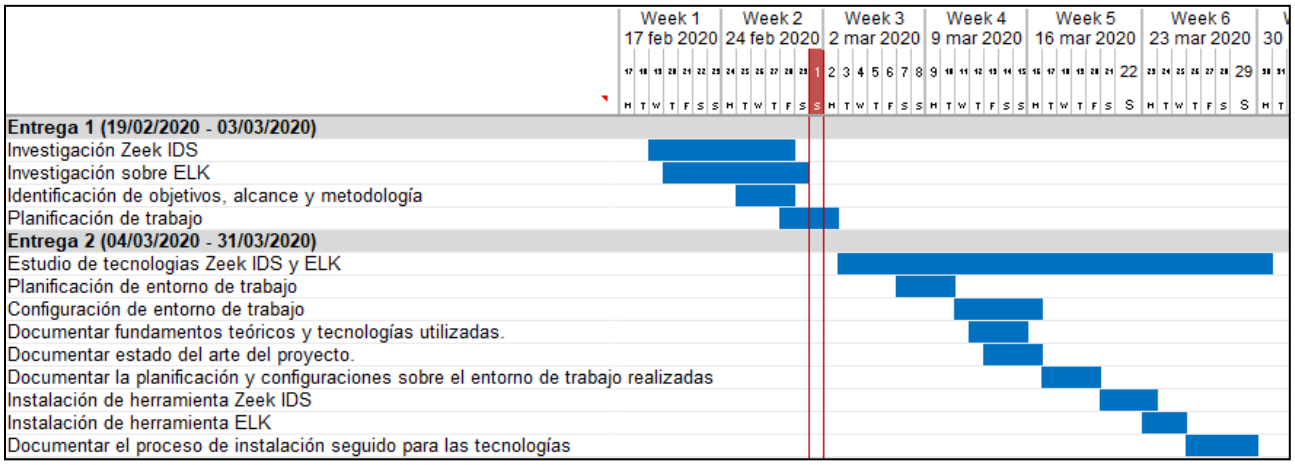

Ilustración 1. Diagrama Gantt. Parte 1.

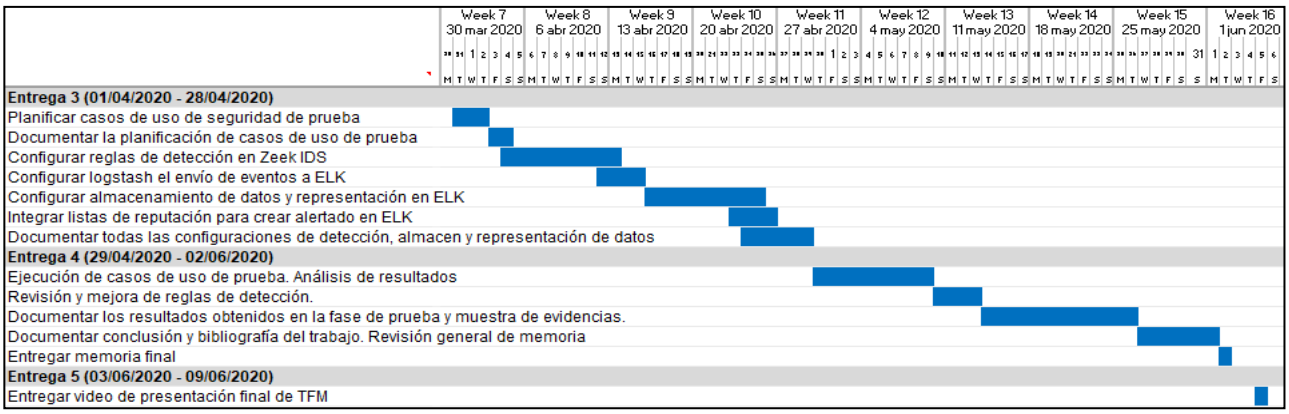

Ilustración 2. Diagrama Gantt. Parte 2.

1.5 Análisis de riesgos

En este apartado se identifican los principales riesgos o causas técnicas que puedan provocar desviaciones que afecten temporalmente a la realización del proyecto de acuerdo con la planificación establecida. Para cada uno de los riesgos identificados, se proponen una serie de acciones de mitigación que puedan ser llevadas a cabo:

- **Riesgo 1: Falta de conocimiento sobre las tecnologías a utilizar**. Las tecnologías por implementar en este proyecto son Zeek IDS y ELK. Dado que no cuento con experiencia previa con ninguna de ellas, puede ser que ocurran puntos bloqueantes debido a falta de entendimiento o fallos inesperados en el proceso de configuración. Además, se va a desarrollar una funcionalidad más dentro de Zeek con el proyecto BZAR de Mitre.

  Para mitigar este posible riesgo, es necesario realizar los estudios previos de ambas herramientas en base a las documentaciones y formaciones públicas, que permitan abordar el proceso de instalación y configuración con el mayor conocimiento posible. Además, es necesario documentarse sobre el proyecto BZAR y hacer las pruebas oportunas para verificar que Zeek puede realmente integrarse con este proyecto.

- **Riesgo 2: Incompatibilidades en el envío de datos entre Zeek y ELK**. En una de las fases de este proyecto, será necesario enviar los logs generados por Zeek al entorno ELK. Los logs generados por Zeek, cuentan con un formato un tanto peculiar, lo que puede dificultar el envío y almacenado de datos en ELK.

  Para mitigar este riesgo, es necesario hacer pequeños desarrollos de scripting que se encarguen de transforman los logs generados por Zeek a un formato más legible, como por ejemplo en formato JSON, que facilite su integración en el envío de datos en ELK.

- **Riesgo 3. Falta de información sobre fuentes de reputaciones**. En este proyecto, se trabajará con fuentes de reputación sobre botnets existentes actualmente, para conseguir integrar estas fuentes en Zeek y de esta forma, enriquecer los logs generados mediante el módulo de inteligencia de Zeek. Dado que desconozco de fuentes públicas de datos a utilizar y que sean confiables, puede surgir algún bloqueo en esta fase de desarrollo dentro de Zeek.

  Como recomendaciones de mitigación, es necesario estudiar e investigar sobre el módulo de inteligencia de Zeek en su documentación oficial, así como el estudio de fuentes de información fiables sobre botnets con las que poder trabajar.

- **Riesgo 4: Posibles casos prácticos complejos**. Dentro de la fase de pruebas del proyecto, se pretenden analizar una serie de capturas

de tráfico sobre incidentes reales de seguridad. Cabe la posibilidad de enfrentarnos a algunos casos demasiado complejos de los cuales no podamos obtener demasiada información o la desconozcamos.

Las acciones mitigadoras para este riesgo serían, limitar el ámbito de análisis del caso práctico al justamente necesario para proponer y mostrar el potencial de Zeek IDS y como estos datos son representados en ELK. En el peor de los casos, se deberá de sustituir algún caso práctico por otro más adaptable a la solución.

1.6 Breve sumario de productos obtenidos

La idea general de este proyecto consiste en disponer de un entorno de pruebas, en el cual se despliegue la herramienta Zeek IDS que se encarga de analizar el tráfico de red de dicho entorno. En Zeek se configurarán una serie de reglas de detección con el fin de generar eventos cuando se detecte en el tráfico de red alguna de las anomalías de seguridad contempladas.

Todos aquellos patrones anómalos de seguridad que sean detectados por Zeek serán enviados a la herramienta ELK disponible en otra máquina virtual. Este envío de datos se realizará mediante Logstash, y una vez los datos se almacenen en Elasticsearch, se pretende lograr la generación de una serie de gráficas de análisis que permitan una mejor representación y análisis de datos.

Por último, una vez dispongamos de todos los eventos de seguridad almacenados y representados, se pretende realizar una integración con un feed de reputaciones de botnets, con el objetivo de detectar conexiones hacia equipos sospechosos y generar alertas en caso de que se cumplan dichas conexiones.

Por tanto, el producto final obtenido, sería el equivalente a la integración de una herramienta de tipo IDS mediante Zeek, y su integración con ELK, el cual mediante la configuración de alertas y su correlación para encontrar conexiones sospechosas, cumplirá el rol de una herramienta similar a un SIEM.

1.7 Breve descripción de los otros capítulos de la memoria

A continuación, se presenta la organización de la memoria de este proyecto para facilitar la lectura y compresión de las diferentes partes de esta. La organización ha sido planteada en base a los siguientes capítulos, excluyendo el capítulo 1 de introducción al contexto y objetivos del proyecto:

- **Capítulo 2: Fundamentos teóricos y tecnologías utilizadas**. En este capítulo se describirá que son los sistemas de detección de intrusos o IDS y las distintas funcionalidades de estos, así como los principios básicos en el ámbito de la seguridad informática.

- **Capítulo 3: Tecnologías utilizadas**: Se describirán las tecnologías Zeek IDS y ELK, así como las funcionalidades principales que ofrecen.

- **Capítulo 4: Estado del arte**. Se describirá el estado actual del mercado sobre tecnologías IDS y SIEM, así como los productos más importantes de cada tecnología y sus principales funcionalidades y diferencias.

- **Capítulo 5: Configuración del entorno de trabajo**. En este apartado se detallará cómo será el entorno de trabajo dónde se implementará y desarrollará la solución definida para el proyecto, así como el entorno donde se realizarán las pruebas de funcionamiento.

- **Capítulo 6: Proceso de instalación de tecnologías.** En este capítulo se mencionarán los detalles y pasos seguidos para el proceso de instalación de cada una de las tecnologías en las maquinas del entorno preparado.

- **Capítulo 7: Escenarios de pruebas de seguridad.** En este capítulo se mostrarán algunas de las principales anomalías de seguridad que se puedan dar en el tráfico de red, así como la definición de los casos de pruebas que se van a intentar detectar por la herramienta.

- **Capítulo 8: Proceso de configuración de tecnologías**. En base a los casos de seguridad que se van a intentar detectar en las pruebas, se documentarán todas las configuraciones oportunas de las tecnologías para las pruebas definidas.

- **Capítulo 9: Evaluación de prestaciones de la herramienta.** En este capítulo se mostrará cómo se ejecutan las pruebas de seguridad y se mostrará evidencias de que la herramienta es capaz de detectar los patrones de seguridad definidos, asegurando su funcionalidad.

- **Capítulo 10: Conclusiones**: En el capítulo final se expondrán las principales ideas de la realización de este proyecto, comparando los resultados finales con los previstos antes de la realización de este en cuanto a objetivos, así como el planteamiento de posibles futuros trabajos que amplíen la funcionalidad y horizontes de la solución desarrollada.

2. Fundamentos teóricos.

En este capítulo se abordarán los principales fundamentos teóricos sobre los cuales está basada la solución implementada para este proyecto. Entre los principales contenidos teóricos que se comentaran a continuación, destacan el concepto de seguridad en el ámbito de la informática, así como la explicación de la importancia de los sistemas de detección de intrusiones y sus principales características.

2.1 Seguridad informática.

La seguridad informática no para de innovarse y crecer, debido al rápido crecimiento y la masiva utilización de las tecnologías de la información por parte de los seres humanos. Dado que estas tecnologías son el medio más empleado y al ser imprescindibles actualmente para el ser humano, lo convierten en un objetivo por parte de ciberdelincuentes, que intentarán aprovechar brechas de seguridad en estas tecnologías para el robo de información, dinero, datos, etc.

Existen muchas definiciones para el concepto de Seguridad Informática. La más extendida o utilizada entiende la seguridad informática como una rama de la tecnología de la información que consiste en la implantación de un conjunto de medidas técnicas destinadas a preservar la confidencialidad, integridad y disponibilidad de la información, pudiendo, además, garantizar otras propiedades sobre la información como la autenticidad, la responsabilidad, fiabilidad y el no repudio. [3]

Los **principales atributos** de la Seguridad informática son:

- **Confidencialidad**: Consiste en garantizar que solo podrá acceder a la información aquellos agentes o usuarios que están autorizados para ello.

- **Integridad**: La información no debe de sufrir alteraciones cuando esta es almacenada, recuperada o transmitida.

- **Disponibilidad**: La información debe de poder ser utilizada siempre que sea necesitada por los usuarios autorizados.

Ilustración 3. Atributos principales de la seguridad informática.

Estos son los pilares básicos de la seguridad informática, pero, además, existen otros **atributos secundarios** a mencionar: [3]

- **Autenticidad**: Consiste en el proceso de identificar el generador de la información, y verificar que este usuario es realmente quien dice ser, así como el usuario al que iría destinada la información generada.

- **Responsabilidad**: Son el conjunto de medidas preventivas o acciones a llevar a cabo por los usuarios de las tecnologías para evitar malas prácticas que puedan causar posibles brechas de seguridad. La seguridad comienza en la responsabilidad del usuario.

- **No repudio**: Se trata del proceso de identificar de forma biunívoca la pertenencia de una información a un individuo.

- **Fiabilidad**: Se entiende como la probabilidad de que un sistema se comporte tal y como se espera de este, de esta forma se podrá decir que está siendo ejecutado de forma segura.

Por último, a modo general, es importante comentar que existen **tres tipos de seguridad informática**, según los diferentes ámbitos o tecnologías específicas que aborden: [3]

- **Seguridad Hardware**: Consiste en la protección de los equipos o sistemas físicos ante cualquier evento, ataque o suceso que pueda dañar su integridad o fiabilidad. Algún ejemplo de sistema que garantice este tipo de seguridad pueden ser sistemas antirrobo, cortafuegos, servidores proxy, etc.

- **Seguridad Software**: Se trata de aquellas técnicas que eviten o dificulten el hecho de que un posible atacante, mediante técnicas de hacking o similares, pueda introducirse en los sistemas con fines maliciosos. Normalmente, este tipo de seguridad se ve afectada por lo que se conoce como vulnerabilidades, las cuales pueden ser de los sistemas operativos del sistema del usuario, de las aplicaciones utilizadas por este, etc.

- **Seguridad de la Red**: Se refiere a todas aquellas actividades destinadas a proteger la red utilizada por los sistemas tecnológicos. Estas actividades tienen como fin proteger la fiabilidad, integridad y seguridad de los datos que se transmiten por la red, para que estos no puedan ser interceptados o manipulados por un atacante. Normalmente los ataques a este tipo de seguridad suelen venir por los conocidos virus, gusanos, spyware, etc.

2.2 Sistemas de detección de intrusos.

Los **sistemas de detección de intrusiones (IDS)** son aquellos encargados de detectar las intrusiones o el intento de estas dentro del entorno que se pretende proteger, entendiendo una intrusión, como el conjunto de acciones llevadas a

cabo por un sujeto con el fin de comprometer la integridad, confidencialidad o disponibilidad de un recurso informático. [2]

Un IDS, normalmente es utilizado para el ámbito de seguridad de red mencionado en el apartado anterior. Por esta razón, el **funcionamiento básico** de estas herramientas es realizar un análisis exhaustivo y desgranado del tráfico de red, donde dicha información es comparada con comportamientos sospechosos, firmas, ataques conocidos, permitiendo detectar y categorizar ataques. Un IDS utiliza alguna de estas técnicas para detectar un ataque:[4]

- **Patrones**: Un IDS basado en patrones se encarga de analizar el tráfico de red y los compara con una serie de firmas de ataques conocidos. Esta técnica requiere un margen de tiempo desde que el ataque ocurre y este es descubierto mediante un patrón, siendo necesario su configuración en el IDS para que sea capaz de detectar el ataque en ocasiones posteriores.

- **Heurística**: Esta técnica permite categorizar actividad normal dentro de la red basado en el ancho de banda, servicios levantados, protocolos utilizados, etc. Todo lo que varíe de esa actividad normal, será categorizado como actividad anómala.

Normalmente, este tipo de tecnologías suelen **implementarse** en conjunto con un firewall, ya que, de esta forma, se une el potencial de detección y la inteligencia aportada por un IDS, con el potencial de bloqueo de un firewall, permitiendo que los paquetes y trafico identificado como sospechoso pueda ser bloqueado a tiempo. Se diferencian dos formas normalmente sobre la implementación de un IDS en redes: [4]

- Si la red se encuentra segmentada con hub, se puede analizar el tráfico de red realizando la conexión al IDS de forma directa a cualquier puerto.

- Si se utiliza un switch, se requiere conectar el IDS a un puerto SPAN para que pueda analizar todo el tráfico de esta red.

Los **principales requisitos de un IDS** para que este pueda funcionar correctamente en el entorno deseado: [2]

- Un sistema IDS debe de ser autónomo, es decir, debe ser capaz de ejecutarse continuamente de forma ininterrumpida y sin que nadie este obligado a supervisar su funcionamiento.

- La aceptabilidad dentro del entorno en el que se va a implementar, asegurando que no sobrecargue el sistema o que no sobrecargue el tráfico de red.

- La adaptabilidad del IDS a los cambios dentro del entorno de trabajo dado el dinamismo continuo del mundo de la informática.

- Tolerancia a fallos y capacidad de respuesta ante situaciones inesperadas, que puedan provenir de un cambio en el entorno, reinicio inesperado de máquinas, etc.

Por último, una vez conocido que son los sistemas de detección de intrusiones, cuál es su funcionamiento básico, su forma de implementación y los requisitos para su correcto funcionamiento, comentamos a continuación **la clasificación o tipos de IDS existentes actualmente**: [4]

- **HIDS o Host IDS**: Este tipo de sistemas intenta detectar las modificaciones o rastros dejados por un atacante al lograr una intrusión en el sistema.

- **NIDS o Network IDS**: Se trata de un IDS basado en red, detectado todo tipo de ataques que afecten y ocurran dentro del segmento de red a monitorizar. Para su funcionamiento, se requiere la configuración de la interfaz de red en modo promiscuo, capturando así todo el tráfico de red.

- **IDS basados en firmas**: Estos sistemas se encargan de monitorizar los paquetes de red y los compara con una base de datos de firmas que disponen en su analizador.

- **IDS basados en anomalías:** Estos sistemas monitorizar el tráfico de red y lo comparan con una línea base definida de tráfico normal o legítimo de red, categorizando como anómalo todo aquello que se salga de esta línea base definida.

- **IDS Pasivo**: Trata de un IDS cuya función principal se reduce a la detección y el alarmado de cualquier tipo de amenaza, bloqueando dicha actividad de forma preventiva.

- **IDS Reactivo**: Estos IDS se encargan de detectar actividad maliciosa y alertar sobre estas amenazas, además de tomar acciones contra ellas, como puede ser, ordenar el bloqueo de tráfico en el cortafuegos. Normalmente, este tipo de IDS, se denominan IPS o sistema de prevención de intrusos.

En este proyecto, trabajamos con Zeek IDS, el cual tarta de un IDS de red o NIDS, el cual comentaremos más en detalle en el siguiente capítulo.

2.3 SIEM.

Los sistemas **SIEM** (Security Information and Event Monitoring) tienen como objetivo detectar preventivamente amenazas potenciales a la empresa y resolverlas lo más rápido y de la forma más eficaz posible.

Las herramientas de tipo SIEM provienen de dos soluciones diferentes: [5]

- **SEM:** Seria el administrador de eventos de seguridad, encargado de procesar y monitorizar eventos de seguridad en tiempo real, aplicando

inteligencia sobre estos eventos mediante un motor de correlación capaz de crear alertas.

- **SIM:** Corresponde a la parte de administrador de información de seguridad, encargado de almacenar todos los eventos generados a nivel histórico, utilizando estos eventos para poder generar informes, análisis de eventos en el tiempo y análisis forense.

Normalmente en una arquitectura de seguridad donde se empleen soluciones SIEM, esta suele ir alimentada de los logs de la organización que se pretende securizar, así como de logs que provengan de otros agentes de seguridad implementados en la organización, como pueden ser firewall, IDS, IPS, etc.

Entre las **principales capacidades** que ofrece un SIEM destacan las siguientes: [5]

- Recolección y gestión de logs.
- Correlación de eventos de seguridad en tiempo real.
- Monitorización y alertas de seguridad.
- Informes, reportes y dashboard de análisis de información.
- Capacidad forense.
- Aplicar seguridad en los equipos clientes o workstations.
- Ofrece capacidad de respuesta tomando medidas reactivas.

Por estas capacidades, resulta muy aconsejable contar con una solución SIEM dentro de toda organización que pretenda monitorizar y usar una solución de seguridad.

En este proyecto, se pretende realizar una integración parecida a esta, entre un ELK simulando el funcionamiento de SIEM, que recoja los logs provenientes de Zeek IDS.

2.4 Botnets y su detección.

Botnet es el nombre genérico por el que se denomina a un grupo de sistemas infectados y controlados por un atacante u organización de forma remota.

Normalmente, una **botnet** es creada usando un malware que infecta a una gran cantidad de máquinas, siendo estas máquinas denominadas “bots” o “zombies”, formando parte de la botnet. Las botnets más conocidas en los últimos años por su gran volumen de afectación que provocaron son: [6]

- Storm
- Conficker
- Zeus
- Trickbot
- Flashback
- Windingo

Las botnets hoy en día suponen un gran problema para la gran mayoría de organizaciones y empresas, ya que constantemente estas actualizándose y empleando una gran cantidad de técnicas innovadoras para que sus ataques tengan mayor probabilidad de éxito y pasen desapercibidos por las técnicas de detección actuales. Algunas de las técnicas utilizadas principalmente por las botnets son: [6]

- El uso de técnicas como Fast Flux
- Generación dinámica de dominios
- La ofuscación de las comunicaciones
- Uso de comunicaciones P2P y de la red Tor
- Implementación de medidas antireversing

Para hacer frente a las técnicas anteriores, se definen una serie de técnicas pasivas y activas para ayudar en la detección de estas botnets, como se puede ver a continuación. [6]

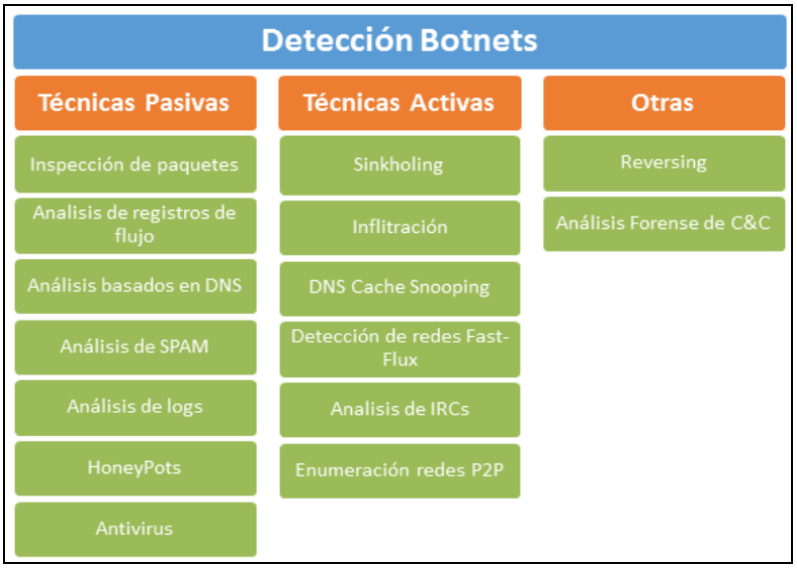

Ilustración 4. Técnicas de detección de botnets.

Las **técnicas pasivas** consisten en la observación, sin modificar ni interferir en los elementos implicados, simplemente aplicando métodos de análisis de comportamiento y firmas. Dentro de las técnicas pasivas, el método más empleado de análisis es el de **inspección de paquetes**, el cual consiste en inspeccionar los paquetes del tráfico de red en busca de patrones o características definidas para detectar amenazas.

Los sistemas y aplicaciones existentes enfocados en realizar inspección de paquetes son los IDS e IPS. Por tanto, podemos entender la gran importancia de los sistemas IDS en arquitectura de seguridad en organizaciones hoy en día para ampliar la seguridad de los sistemas, y entre otras capacidades, **detectar conexiones sospechosas a botnets** conocidas e identificadas en blacklist.

En este proyecto, se pretende desarrollar esta funcionalidad en Zeek, para ser capaz de detectar conexiones sospechosas hacia destinos de tipo Command and Control.

3. Tecnologías utilizadas.

En este capítulo se expondrán las diferentes tecnologías y herramientas que han sido necesarias para el trabajo y desarrollo de este proyecto.

3.1 Zeek IDS.

La historia de Zeek se remonta a hasta hace más de dos décadas, cuando Vern Paxson diseñó e implementó la versión inicial de esta solución, conocida entonces como Bro IDS.

A lo largo de los años se han ido sumando numerosas organizaciones y entidades en el proyecto, que han permitido un gran avance en la herramienta. Dentro de estas organizaciones, destacan el Laboratorio Nacional Lawrence Berkeley (LBNL), la Fundación Nacional de Ciencia (NSF) o el instituto internacional de ciencias de la computación (ICSI). En la imagen siguiente se muestra el avance cronológico en la historia de la herramienta Zeek. [7]

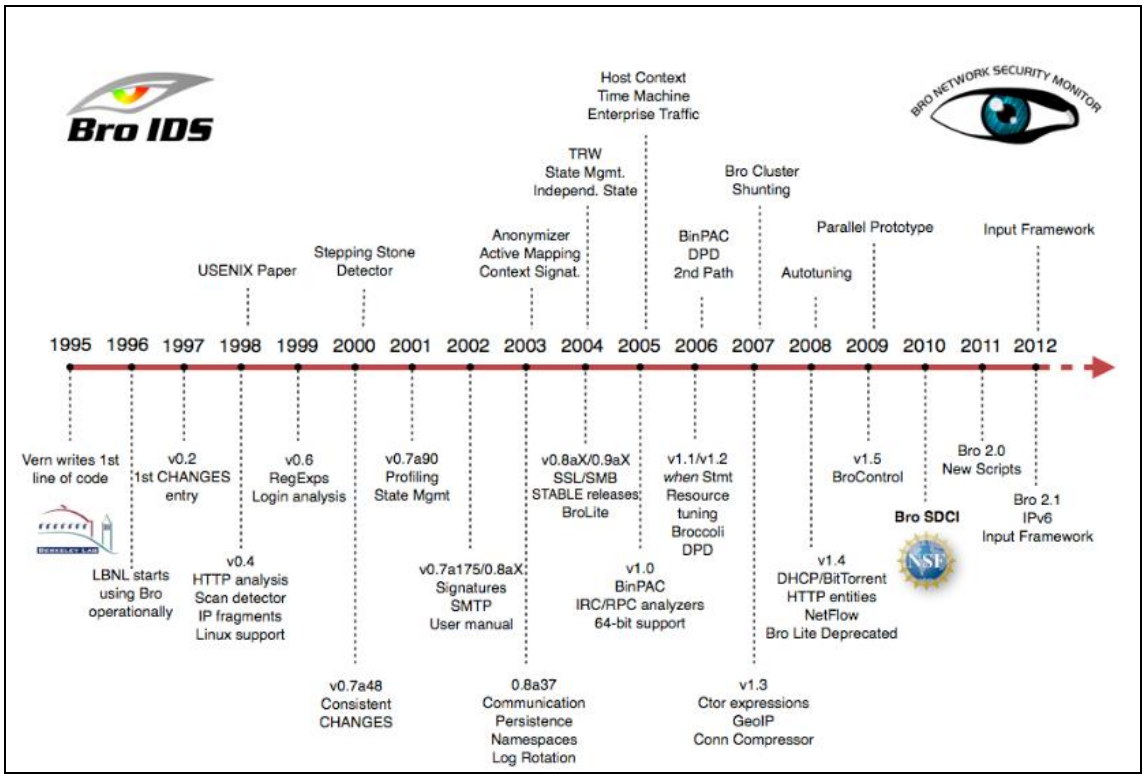

Ilustración 5. Cronología de la historia de Zeek.

La experiencia previa con Zeek consistía en una herramienta con una capacidad de detección y potencial enorme, dando un gran valor en materias de seguridad en aquellos entornos donde se implementaba. La principal desventaja que tenía antes esta herramienta era su gran complejidad de personalización y su elevada curva de aprendizaje. Para solucionar este problema, se trabajó en conjunto con el centro nacional de aplicaciones de supercomputación (NCSA), con el cual se desarrolló una nueva versión de la herramienta, llamada desde entonces Zeek 2.0. Desde esta versión, el uso de Zeek ha crecido enormemente, dada la flexibilidad que ofrecía con capacidad

de incluir nuevas implementaciones adaptadas a cada entorno con un lenguaje de programación propio de la herramienta, no tan complejo y fácilmente entendible, además de proporcionar una documentación y manual de uso muy completa. [7]

Zeek es un sistema de detección de intrusiones de tipo NIDS. Trata de una herramienta de código abierto (licencias BSD), para el análisis de trafico de red cuyo objetivo es detectar actividades sospechosas.

Esta herramienta dispone de una seria de **scripts o políticas** escritos en lenguaje nativo de Zeek. Estas políticas definen las actividades que se consideran como sospechosas, y en base a estas, se realizan las detecciones oportunas en el tráfico de red monitorizado. Además, estas políticas, generan logs de tráfico normal o legítimo. La herramienta incluye la capacidad de permitir desarrollar políticas propias y específicas para cada entorno de red, buscando ataques concretos que se pretendan detectar.

Las **principales características** de Zeek son las siguientes: [7]

- **Despliegue:**
  - Se ejecuta en hardware de sistemas estándar de tipo UNIX (Linux, FreeBSD y MacOS).
  - Análisis de tráfico completamente pasivo fuera de un puerto de red o monitoreo.
  - Utilización de libpcap para capturar paquetes.
  - Permite realizar análisis en tiempo real y offline.
  - Permite realizar implementaciones con clúster a gran escala.
  - Dispone de un marco de administración unificado.

- **Análisis**:
  - Zeek puede ser ejecutado para la detección en tiempo real de anomalías configurando una interfaz de red a la escucha o mediante la lectura de capturas de tráfico especificas o pcap, permitiendo realizar estudios sobre posibles incidentes de seguridad ocurridos.
  - Permite realizar un análisis independiente de puertos de protocolos de capa de aplicación (DNS, FTP, HTTP, IRC, SMTP, SSH, SSL).
  - Analiza los contenidos de archivos para la toma de huellas digitales a través del cálculo MD5 / SHA1.
  - Incluye soporte completo de IPv6.
  - Zeek decapsula los túneles y luego analiza su contenido como si no hubiera ningún túnel en su lugar.
  - Incluye la capacidad de detectar coincidencias de patrones de estilo IDS.

- **Lenguaje de escritura:**
  - Lenguaje completo de Turing para la expresión de tareas de análisis arbitrario.
  - Modelo de programación basada en eventos.

- Tipos de datos específicos del dominio, como direcciones IP (manejo transparente de IPv4 e IPv6), números de puerto y temporizadores.
- Soporte para rastrear y administrar el estado de la red.

- **Interfaz**
  - Salida predeterminada a registros ASCII bien estructurados.
  - Backends alternativos para Elasticsearch y DataSeries
  - Integración en tiempo real de entrada externa en análisis.
  - Biblioteca externa de C para intercambiar eventos de Zeek con programas externos.
  - Capacidad para desencadenar procesos externos arbitrarios desde el lenguaje de script.

En cuanto a la **arquitectura de Zeek**, se basa en dos componentes principales: [8]

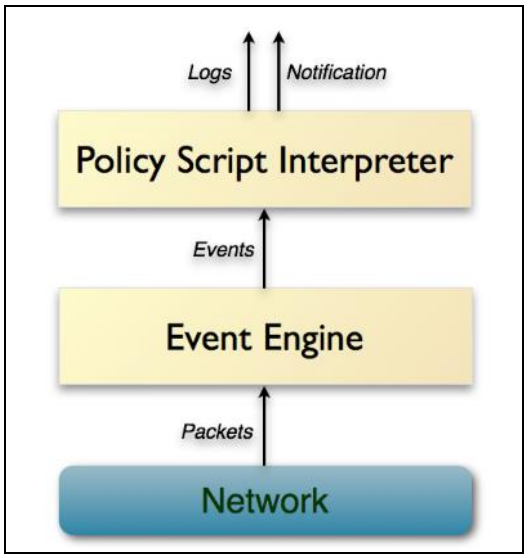

Ilustración 6. Arquitectura interna de Zeek.

- **Motor de eventos** (event engine): Reduce el flujo de paquetes, organizándolos para ser llevados a un nivel superior.

- **Interprete de Scripts** (policy script interpreter): acciones a tomar cuando se detecta una actividad determinada.

En capítulos posteriores, comentaremos el proceso detallado de instalación de esta herramienta, así como las configuraciones necesarias para la detección de las anomalías de seguridad definidas en el plan de pruebas.

3.2 ELK.

ELK (Elasticsearch, Logstash y Kibana) es un conjunto de herramientas de gran potencial de código abierto que se combinan para crear una solución de administración de registros, permitiendo la monitorización, consolidación y análisis de logs generados por múltiples fuentes de datos. [9]

Las tres herramientas que componen principalmente ELK pueden ser utilizadas en soluciones de forma independiente, pero utilizando todas ellas en conjunto, forman una solución muy completa que permite sacar el máximo partido a las funcionalidades ofrecidas por estas.

Las **principales funcionalidades que ofrece** ELK son las siguientes: [9]

- Recolección de logs de eventos y de aplicaciones.
- Se encarga de procesa toda la información anterior y la pone a disposición de los usuarios que la necesiten.
- Incluye distintos módulos a implementar de seguridad, permitiendo que un usuario acceda solo a aquella información que pueda o deba administrar.
- Realiza un formateo o parseo de los campos de los eventos recogidos, convirtiéndolos en opciones de búsqueda y filtrado.
- Toda la información se presenta en visualizaciones o dashboard, donde se pueden realizar búsquedas, filtrados, agregaciones, alarmados, generación de reportes, etc. Es decir, permite una representación de datos muy completa, facilitando el trabajo de los analistas que deben de trabajar con estos datos.

Por tanto, el funcionamiento general de ELK se puede resumir en la siguiente imagen, donde a partir de una fuente de datos, Logstash se encarga de recoger estos datos y procesarlos antes de ser almacenados en un índice de Elasticsearch, siendo toda esta información representada por medio de visualizaciones y búsquedas en Kibana. [9]

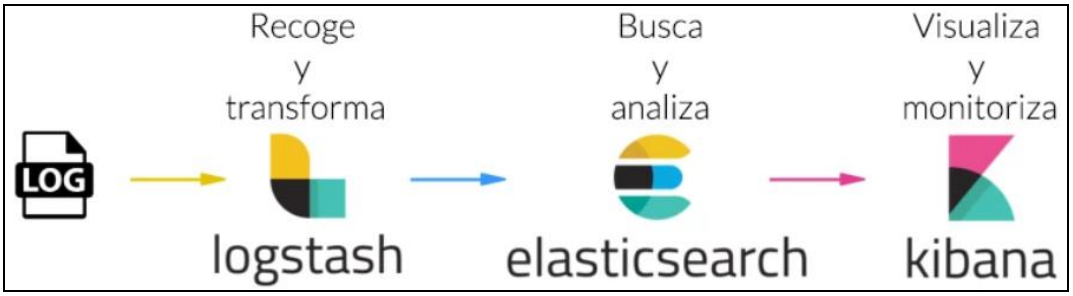

Ilustración 7. Funcionamiento general de ELK.

Logstash

Esta herramienta es la encargada de **definir una serie de inputs** y orígenes de datos, como pueden ser logs, ficheros o documentos, y sobre los cuales, realiza una serie de **acciones de pre procesado** antes de ser enviado a la instancia de Elasticsearch y ser almacenado en un índice concreto de datos. [9]

Logstash debe de ser configurada con Elasticsearch para sincronizar el origen de datos y el destino de estos. Estas configuraciones las comentaremos en apartados posteriores de esta memoria.

Elasticsearch

Elasticsearch trata de una **base de datos distribuida**, donde toda la información es distribuida a todos los nodos disponibles, garantizando así la disponibilidad de datos en todo momento y la tolerancia a fallos. [9]

Además, también es capaz de distribuir el procesamiento de datos, ya que cuando se realiza una búsqueda, será cada nodo de elastic el que procese dicha información y devuelva los resultados, siendo por tanto, un procesamiento distribuido, que garantice los mejores rendimientos posibles.

En esta base de datos, se definen los llamados **índices**, y las configuraciones o mappings de estos, dónde de configuran todos sus campos o parámetros posibles. Estos índices serán aquellos dónde se guarde la información recogida y procesada por el componente Logstash.

Elasticsearch debe de ser configurada con Kibana para sincronizar ambos elementos y permitir la representación de los datos. Estas configuraciones las comentaremos en apartados posteriores de esta memoria.

Kibana

Kibana es el elemento de ELK donde se **generan las visualizaciones y dashboards** sobre la información almacenada en la base de datos de Elasticsearch. [9]

Además de las representaciones comentadas, se pueden generar búsquedas de datos a través de un portal web muy intuitivo y potente, además de definir búsquedas que generen alertas, generación de reportes, módulo de desarrollo para las pruebas necesarias, así como, la gestión avanzada de los índices de datos y sus plantillas.

3.3 Python.

Python fue creado y desarrollado por Guido van Rossum a finales de los años ochenta. Se trata de **un lenguaje de programación interpretado** característico por su facilidad tanto a la hora de programar como su fácil legibilidad. Además, se trata de un lenguaje multiparadigma, es decir, es un lenguaje orientado a objetos que permite una programación imperativa y funcional. Python, cuenta con un extenso abanico de librerías que se podrán utilizar y combinar en nuestros programas, para permitir ampliar aún más la potencia de este lenguaje y ampliar sus horizontes, enriqueciendo los programas creados. [10]

En este proyecto, se trabajará con Python con el fin de desarrollar unos sencillos script que permitan la transformación de los logs generados por la herramienta Zeek a un formato de tipo JSON, para que estos pueden ser tratados de forma más sencilla por Logstash y ser almacenados en la solución

ELK. En concreto, utilizaremos la librería ParseZeekLogs, y cuyo desarrollo comentamos más adelante en este documento.

Todas estas increíbles ventajas y características que nos ofrece Python hacen de este lenguaje uno de los más potentes y utilizados en la actualidad, siendo su uso previsto de un gran crecimiento en un futuro. Por último, cabe destacar que Python es considerado uno de los lenguajes de programación característicos en lo que al área de Seguridad Informática se refiere, gracias a su capacidad para permitir estructurar una idea y demostrarla en diferentes pruebas de concepto. [11]

3.4 Virtual Box.

La herramienta **Oracle VM VirtualBox es un software de virtualización** para arquitecturas de tipo x86/amd64. Lo que ofrece esta herramienta es la capacidad de instalar en ellas cualquier tipo de sistema operativo, conocido como “sistema invitado” dentro del sistema anfitrión, que se trata del que tiene el computador del usuario, creando diferentes entornos virtuales. Por tanto, es como si dentro de una misma máquina, tuviésemos disponibles varios tipos de máquinas diferentes virtualizadas, pudiendo trabajar en todas ella de forma simultánea. [12]

Los sistemas operativos que ofrece esta herramienta para virtualizar son:

- GNU/Linux (Ubuntu, Debian, CentOS)
- Mac OS X
- OS/2 Warp
- Microsoft Windows
- Solaris/OpenSolaris

Para el proyecto, será necesario desplegar dos máquinas virtuales en Virtual Box, como se puede ver en la imagen siguiente, cuyas características comentaremos en el capítulo de configuración del entorno de trabajo.

4. Estado del arte.

En este capítulo realizaremos un análisis de las diferentes herramientas existentes actualmente en el mundo tecnológico que se asemejen a la funcionalidad de la solución llevada a cabo en este proyecto. En concreto, se han realizado una serie de investigaciones sobre el estado actual del mercado para los productos de sistemas de detección de intrusos o IDS, así como para los productos SIEM.

Dentro de los sistemas de detecciones de intrusiones (IDS) se han encontrado los siguientes productos más conocidos y empleados en la actualidad, centrándonos en soluciones de código abierto:

- **Snort**

Es una solución libre y gratuita. Se trata de una de las herramientas dentro de los IDS con mayor tiempo de existencia en el mercado, convirtiéndose en un estándar básico en sistemas de detección de intrusiones (IDS) e incluso en sistemas de prevención de intrusiones (IPS). Este sistema ofrece la capacidad de almacenamiento de bitácoras en archivos de texto y en bases de datos abiertas, como MySQL. Tiene un lenguaje de reglas de detección muy potente y útil. [13]

Entre las **principales ventajas de snort**, podemos destacar que se trata de un proyecto estable y con bastantes años de existencia, lo cual asegura su implementación evitando problemas futuros. Cabe destacar también el gran poder de la comunidad de snort creciente en los últimos años, siendo de gran ayuda en aquellos que utilicen esta herramienta por primera vez.

En cambio, la **principal desventaja** que presenta snort en la inexistencia de una interfaz gráfica que ayude a su administración y configuración. [14]

- **Suricata**

Este producto cumple la finalidad de sistema de detección de intrusos (IDS) y de sistema de prevención de intrusos (IPS). Fue desarrollado por la Open Information Security Foundation. Su funcionamiento es muy similar a snort, utilizando ambos las mismas bases de datos de firmas. Dentro de las **principales ventajas** de suricata encontramos: [14]

  - Dispone de capacidad de ejecución en multi hilo, aumentando considerablemente su rendimiento.
  - Permite utilizar tarjetas gráficas para inspeccionar el tráfico de red.
  - Realiza una extracción de ficheros completa para poder detectar posibles descargas de malware.
  - Mediante LuaJIT, permite aumentar la funcionalidad mediante una serie de reglas basadas en scripting.
  - Es capaz de analizar paquetes, certificados, protocolos, capa de aplicación, etc.

- **Zeek IDS**

Zeek, también conocido como Bro IDS, es un marco de análisis de red de software libre y de código abierto. Se puede usar como un sistema de detección de intrusos en la red (NIDS), contando con la principal funcionalidad de análisis en vivo de eventos de red. Es un producto muy potente, que permite extraer gran información sobre el tráfico de red, permitiendo obtener una gran cantidad de inteligencia sobre las amenazas de red más actuales. [13]

Las **principales ventajas que ofrece Zeek** es el gran potencial de interprete de políticas basadas en script con un lenguaje propio de administración, que ofrece unas posibilidades muy interesantes al administrador de la herramienta. Además, permite una automatización en las tareas de análisis y recolección de datos a un gran nivel, siendo capaz de extraer ficheros sospechosos, realizar un análisis de malware, notificar si existen problemas e incluso, alimentar una blacklist.

La **principal desventaja que presenta Zeek** es la elevada curva de aprendizaje, pero gracias al creciente apoyo de la comunidad y su documentación, cada vez resulta más sencillo adaptarse a una de las herramientas más potentes del ámbito IDS. [14]

- **Kismet**

Es una herramienta que se ha convertido en el estándar de los IDS de tipo Wireless, los cuales no se centran tanto en la carga de paquetes de red, si no en el análisis de los eventos que suceden en la red.

La principal ventaja que ofrece Kismet es su capacidad de encontrar puntos de acceso falsos o simulados que puedan provocar una brecha de seguridad. Sin embargo, como **desventaja**, se trata de una herramienta que ofrece menos funcionalidades como IDS que las anteriormente mencionadas. [14]

- **Ossec**

Trata de un sistema de detección de intrusiones basados en host (HIDS). Dentro de los IDS de este tipo, esta herramienta es el referente que utilizar, dada su versatilidad de ejecución en cualquier sistema operativo y utilizar una arquitectura de tipo cliente-servidor. Por tanto, a parte de su versatilidad, **la gran ventaja** que ofrece ossec es su capacidad de enviar alertas y registros a un servidor centralizado donde poder llevar a cabo los análisis de datos, además de poder controlar una serie de agentes desde un mismo sistema. [14]

La principal desventaja de esta herramienta es que al tratarse de un HIDS, la propia solución podría verse afectado por un ataque de seguridad, por lo que es necesaria, que todos los datos recopilados por la herramienta sean enviados a un servidor seguro.

Sobre los productos de sistemas de gestión de eventos de seguridad (SIEM), algunos de los más populares en la actualidad son los siguientes:

- **IBM QRadar**

Trata de una solución SIEM de IBM que permite a los equipos de seguridad detectar con precisión y priorizar amenazas en toda la empresa, además de proporcionar información inteligente para que los equipos respondan rápidamente y así reducir el impacto de los incidentes. [15]

Se trata de una solución con un precio razonable en el mercado, pero su complejidad de implementación puede que no sea la solución adecuada para algunas organizaciones.

Las principales **características o ventajas** que ofrece esta solución es la capacidad de aplicar correlación de eventos de seguridad en tiempo real, su compatibilidad con Windows, Linux y Mac OS, además de las numerosas integraciones de aplicaciones de terceros que facilitar el monitorizar una gran diversidad de datos.

En cambio, entre sus **principales desventajas**, encontramos su curva de aprendizaje algo elevada, ya que la plataforma puede resultar confusa por no ser muy intuitiva, siendo necesaria una formación previa. [16]

- **Splunk Enterprise Security**

Es una de las soluciones SIEM más potentes y utilizadas en la actualidad, ya que permite un análisis de eventos de datos y un correlado de estos a gran nivel, asegurando la detección de un gran porcentaje de incidentes de seguridad. Es utilizado por clientes muy potentes como, por ejemplo, el gobierno estadounidense. Este SIEM permite centralizar logs para análisis forense y Threat Hunting, además de ofrecer un potente motor de correlación que permitan generar alertas en tiempo real. [15]

La principal ventaja de esta herramienta se basa en la gran flexibilidad que ofrece al administrador para buscar y analizar eventos de seguridad sin ningún tipo de restricción, además de contar con una amplia base de librerías que permite detectar las anomalías principales de seguridad. Por último, cabe destacar los dashboards y cuadros de mandos personalizables, así como sus numerosas integraciones con APIs que permiten monitorizar una gran cantidad de fuentes de datos distintas.

La **principal desventaja** de Splunk Enterprise Security es su alto coste de licencia, que dificulta su implementación en pequeñas o medianas empresas.

- **Elastic Stack**

Es un conjunto de paquetes, en su mayoría de código abierto, que funcionan de forma conjunta para ofrecer esta solución SIEM. Dispone de un motor de búsqueda, un colector de datos y agentes que envíen estos datos, además de

un sistema de visualización que permite la gestión de alertas de seguridad, reportes, monitorización, etc. [16]

Dentro de las **ventajas** de esta herramienta, encontramos su gran capacidad modular y su bajo coste, al ser la mayoría de los elementos que lo componen de código abierto. Cada vez dispone de mayores clientes debido a los altos precios del resto de soluciones SIEM.

En cambio, encontramos **algunas desventajas**, ya que, en su forma básica, no cumple la funcionalidad de SIEM, no dispone de soporte y tampoco viene con reglas de seguridad predefinidas que faciliten el trabajo y detección a los administradores.

- **LogRhythm SIEM**

Es una solución que permite a las organizaciones detectar, responder y neutralizar rápidamente los posibles daños causados por incidentes de seguridad. Esta herramienta unifica gestión de logs, SIEM, análisis forense de redes y puertos finales, así como analítica de seguridad avanzada. [15]

Las principales ventajas de esta herramienta es que es más fácil de implementar que algunos de los otros productos SIEM, así como su facilidad de uso y capacidad de monitorización en tiempo real. Su **principal desventaja** es que no es muy escalable, por lo que no encajaría bien en entornos grandes de producción que trabajen con grandes volúmenes de datos.

- **AlienVault USM**

Trata de una plataforma unificada y diseñada para proporcionar una defensa completa de las amenazas de seguridad más recientes. Es una herramienta con pocos años en el mercado, pero con un precio muy razonable y centrado sobre todo para pequeñas y medianas empresas. [16]

Entre las **principales ventajas** que ofrece este SIEM, podemos numerar las siguientes:

- Inteligencia contra amenazas de seguridad de forma continuada.
- Despliegue rápido y sencillo
- Funciones múltiples de seguridad en una sola consola.
- Presentación de informes y reportes muy completos, además de disponer de una interfaz muy intuitiva.

En cambio, entre sus **desventajas**, podemos encontrar que puede quedar como una solución insuficiente para grandes empresas, o que, requerimientos más técnicos de amenazas y de seguridad, pueden ser más complicados de detectar o monitorizar.

Por tanto, una vez presentado el estado actual de las diferentes soluciones existentes en la actualidad en el ámbito de la seguridad informática, podemos concluir, que la herramienta Zeek IDS con la que trabajaremos trata de una de las soluciones cada vez más utilizadas y emergentes en la actualidad, dado su

gran potencial de detención de anomalías y su uso como análisis forense avanzado, todo ello, de forma gratuita.

En cambio, la herramienta ELK, aunque no la utilizaremos formalmente como un SIEM, analizaremos en ella las detecciones realizadas por Zeek en base a una serie de alertas y dashboards. En base al estudio anteriormente realizado, podemos observar, como no sería una solución SIEM robusta como otras. Sin embargo, podría utilizarse en entornos de pequeñas o medianas empresas, que quieran incluir las funcionalidad y requisitos básicos o mínimos de seguridad.

5. Configuración del entorno de trabajo.

En este capítulo se mostrará la configuración del entorno llevada a cabo, en el cual se realizará el desarrollo y pruebas de este proyecto.

Para la **planificación del entorno**, se han tenido en cuenta las siguientes posibilidades:

- Crear dos **máquinas virtuales** a través de la herramienta VirtualBox, en las cuales se ofreciera la funcionalidad de Zeek y ELK, respectivamente.

- Trabajar con **Docker**, y crear dos contenedores distintos de sistema operativo con versión reducida, en los cuales se preste la funcionalidad de Zeek y ELK, respectivamente.

Tras analizar ambas posibilidades, se ha decidido desplegar dos máquinas virtuales, por la facilidad y experiencia previa en el trabajo con entornos de virtualización, así como la facilidad de depurar fallos entre ellas y visualizar de forma más clara las conexiones y obtención de resultados.

Las características técnicas de estas máquinas son las siguientes:

- **Tamaño de disco duro (VDI):** 80 GB
- **Memoria RAM**: 2048 MB
- **Sistema Operativo**: Debian (64-bit)
- **Distribución**: Kali Linux
- **Configuración de red:** Adaptador puente

En concreto, se han creado las siguientes máquinas virtuales para nuestro entorno de trabajo.

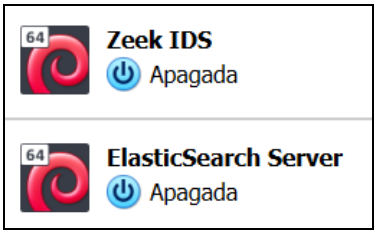

Ilustración 8. Máquinas virtuales en VirtualBox.

- **Zeek IDS:** Sera la máquina virtual donde se instale y configure Zeek IDS y la herramienta Logstash de la solución ELK.

- **Elasticsearch Server**: En esta máquina virtual se instalará y configuraran las herramientas Elasticsearch y Kibana de la solución ELK.

De esta forma, se podrán enviar los logs generados por el IDS a la plataforma dónde se encuentre instalada Elasticsearch, para el almacenado de estos logs en índices, y su posterior representación en Kibana.

Por tanto, la configuración definitiva del entorno se muestra en el siguiente diagrama, el cual se adapta con la solución y objetivos de este proyecto.

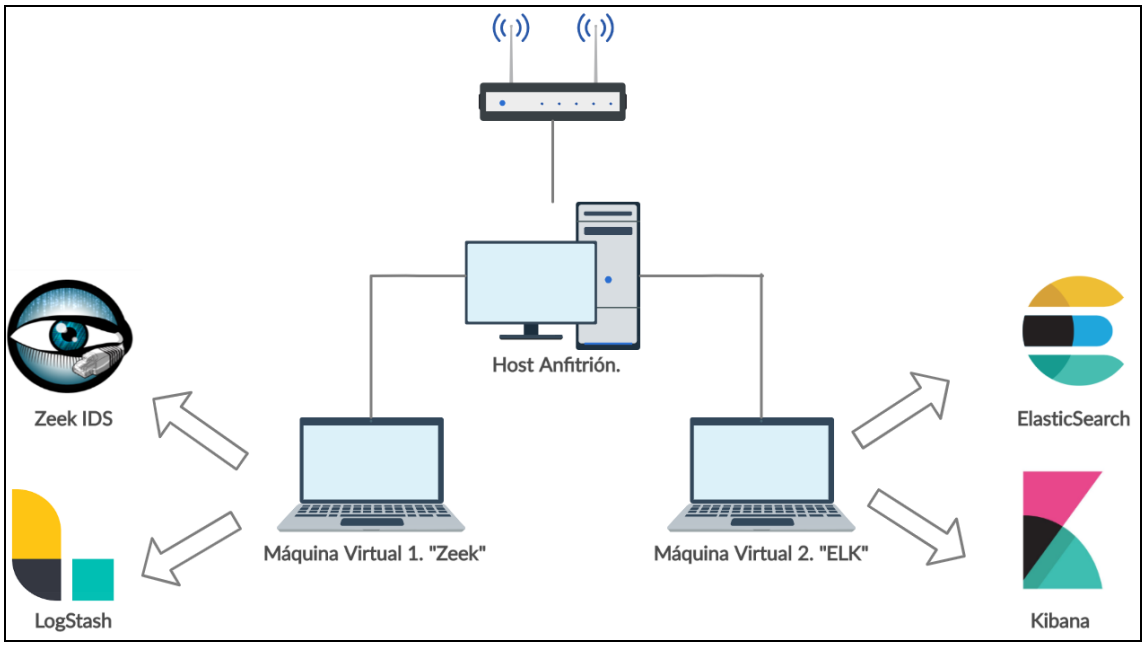

Ilustración 9. Diagrama de arquitectura del entorno.

6. Proceso de instalación de tecnologías.

A continuación, se comenta la fase de instalación de todas las tecnologías necesarias en cada uno de los entornos de virtualización comentados en el capítulo anterior.

En concreto, se comentarán los pasos seguidos para la fase de instalación, así como el inicio de los servicios instalados.

6.1 Instalación de Zeek IDS

Antes de comenzar con la instalación de Zeek, es necesario instalar y resolver algunas **dependencias** de esta herramienta dentro de la máquina. Por tanto, es necesario instalar: [17]

- Libpcap
- OpenSSL libraries
- BIND8 library
- Libz
- Bash (for ZeekControl)
- Python 2.6 or greater (for ZeekControl)
- CMake 3.0 or greater
- Make
- C/C++ compiler with C++17 support (GCC 7+ or Clang 4+)
- SWIG
- Bison 2.5 or greater
- Flex (lexical analyzer generator)
- Libpcap headers
- OpenSSL headers
- zlib headers

Esto lo realizamos con el comando:

```
root@kali:/home/kali#
sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libssl-
dev python-dev swig zlib1g-dev
```

Una vez instaladas las dependencias, procedemos a instalar la herramienta desde la página web donde se encuentra su código fuente (https://github.com/zeek/zeek): [17]

```
root@kali:/home/kali# git clone --recursive https://github.com/zeek/zeek
```

A continuación, **compilamos e instalamos** la herramienta:

```
root@kali:/home/kali# ./configure
root@kali:/home/kali# make
root@kali:/home/kali# make install
```

Por último, antes de poder iniciar la herramienta Zeek IDS, deberemos de configurar la **variable de entorno** para asegurar que poder utilizar todos los comandos predefinidos para gestionar Zeek y su consola. [17] [18]

```
root@kali:/home/kali# export PATH=/usr/local/zeek/bin:$PATH
```

Para **iniciar el servicio**, lanzamos el comando siguiente.

```
root@kali:/home/kali# zeekctl
[ZeekControl] > deploy
```

Por tanto, queda Zeek IDS arrancado y monitorizando por defecto la interfaz eth0 de red, en busca de anomalías. En capítulos posteriores, mostraremos las configuraciones realizadas en Zeek para el plan de pruebas definido.

6.2 Instalación de Logstash

Para la descarga e instalación de la herramienta **Logstash de la solución ELK**, deberemos hacerlo desde el sitio web, donde encontramos el centro de descargas oficial. (https://www.elastic.co/es/downloads/) [19]

La **principal dependencia** de este tipo de herramientas es la necesidad de tener instalado Java en la máquina en la cual se realice la instalación. En los entornos virtualizados creados de Kali Linux, viene instalada por defecto la versión Java 11 JDK.

En caso de ser necesario, se instalaría Java de la siguiente forma:

```
root@kali:/home/kali# sudo apt-get install openjdk-11-jre
```

Descargamos el paquete de instalación en formato .deb con el siguiente comando: [19]

```
root@kali:/home/kali# curl –O
https://artifacts.elastic.co/downloads/logstash/logstash-7.6.1.deb
```

Una vez descargado, procedemos a **instalarlo** mediante el descomprimido del paquete anterior.

```
root@kali:/home/kali# dpkg –i logstash-7.6.1.deb
```

Una vez instalado, ya podemos configurar y arrancar Logstash. Su configuración, se mostrará más en detalle en capítulos posteriores. Para **arrancar el servicio** ejecutamos:

```
root@kali:/home/kali# service logstash start
```

6.3 Instalación de Elasticsearch

Para la descarga e instalación de la herramienta **Elasticsearch de la solución ELK**, deberemos hacerlo desde el sitio web, donde encontramos el centro de descargas oficial. (https://www.elastic.co/es/downloads/) [19]

La **principal dependencia** de este tipo de herramientas es la necesidad de tener instalado Java en la máquina en la cual se realice la instalación. En caso de ser necesario, se instalaría Java de la siguiente forma:

```
root@kali:/home/kali# sudo apt-get install openjdk-11-jre
```

Descargamos el paquete de instalación en formato .deb con el siguiente comando: [19]

```
root@kali:/home/kali# curl –O
https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-
7.6.1.deb
```

Una vez descargado, procedemos a **instalarlo** mediante el descomprimido del paquete anterior.

```
root@kali:/home/kali# dpkg –i elasticsearch-7.6.1.deb
```

Una vez instalado, ya podemos configurar y arrancar Logstash. Su configuración, se mostrará más en detalle en capítulos posteriores. Para **arrancar el servicio** ejecutamos:

```
root@kali:/home/kali# service elasticsearch start
```

6.4 Instalación de Kibana

Para la descarga e instalación de la herramienta **Kibana de la solución ELK**, deberemos hacerlo desde el sitio web, donde encontramos el centro de descargas oficial. (https://www.elastic.co/es/downloads/) [19]

La **principal dependencia** de este tipo de herramientas es la necesidad de tener instalado Java en la máquina en la cual se realice la instalación. En caso de ser necesario, se instalaría Java de la siguiente forma:

```
root@kali:/home/kali# sudo apt-get install openjdk-11-jre
```

Descargamos el paquete de instalación en formato .deb con el siguiente comando: [19]

```
root@kali:/home/kali# curl –O https://artifacts.elastic.co/downloads/kibana
/kibana-7.6.1-amd64.deb
```

Una vez descargado, procedemos a **instalarlo** mediante el descomprimido del paquete anterior.

```
root@kali:/home/kali# dpkg –i kibana-7.6.1-amd64.deb
```

Una vez instalado, ya podemos configurar y arrancar Logstash. Su configuración, se mostrará más en detalle en capítulos posteriores. Para **arrancar el servicio** ejecutamos:

```
root@kali:/home/kali# service kibana start
```

7. Escenarios de pruebas de seguridad.

En este capítulo se presentan los principales escenarios de pruebas con los que se evaluarán las prestaciones de la herramienta Zeek IDS. Es decir, a partir de una serie de casos de uso de seguridad de prueba, basados en capturas de tráfico de eventos de seguridad reales, se verificará si Zeek IDS es capaz de detectar las principales anomalías de seguridad planteadas en cada uno de ellos.

Se harán las pruebas de este proyecto con capturas de tráfico reales, ya que de esta forma, será más sencillo simular comportamientos reales en la red y poder sacar el máximo partido posible a Zeek IDS.

Antes de presentar estos escenarios de pruebas, es necesario conocer cuáles son las principales anomalías de seguridad en la red que podemos encontrarnos en un entorno real, ya que los escenarios de prueba propuestos estarán basados en algunas de estas anomalías.

Normalmente, existen **cuatro tipos de amenazas en la seguridad de la red**: [20]

- **Amenazas no estructuradas**. Basadas en ataques provenientes de usuarios inexpertos que utilizan herramientas sencillas de ataque disponibles en internet, normalmente basados en Shell scripts o password crackers.

- **Amenazas estructuradas**: Estas provienen de hackers, que presentan una motivación y son más hábiles tecnológicamente. Este tipo de atacantes, conocen los sistemas y sus vulnerabilidades, y tratan de crear exploits que puedan provocar un ataque con mayor probabilidad de éxito de pasar desapercibido.

- **Amenazas externas**: Los ataques externos pueden provenir de individuos u organizaciones externas a la compañía, como por ejemplo, grupos APT.

- **Amenazas internas:** Estos ataques provienen de alguien que tiene acceso autorizado a los sistemas de la organización, y una vez dentro de esta, intenta extraer información o atacar servidores y sistemas dentro de la misma red.

Una vez conocidas las principales amenazas de seguridad en la red, podemos destacar los siguientes **tipos de ataques más frecuentes** en el segmento de red de las organizaciones: [20]

- **Reconocimiento:** Consiste en un descubrimiento y mapeo de los sistemas, servicios y posibles vulnerabilidades existentes en una organización. Se trata de la fase previa a un ataque, en la cual, se

recolecta la mayor cantidad de información posible del objetivo del ataque.

- **Acceso**: Este tipo de ataque consiste en la capacidad de un intruso en lograr ganar acceso a un dispositivo, sin estar autorizado para ello.

- **Denegación de servicio**: Este ataque consiste en deshabilitar o corromper servicios de red con la intención de que usuarios normales de la red no puedan hacer uso de ellos. Existe una gran variedad de este tipo de ataques, y pueden suponer grandes pérdidas a la organización afectada.

- **Virus, Gusanos y Troyanos**. El software malicioso es capaz de insertarse en un sistema de la red y una vez este infectado, es capaz de propagarse al resto de sistemas que componen la red, con el fin de dañar el conjunto de sistemas de la red. En los últimos años, el más famoso de estos ataques ha sido el ransomware, y la mayoría de ellos, suelen provenir de botnets de tipo Command and Control, cuyo objetivo es lograr persistencia e infectar el conjunto de sistemas de la red.

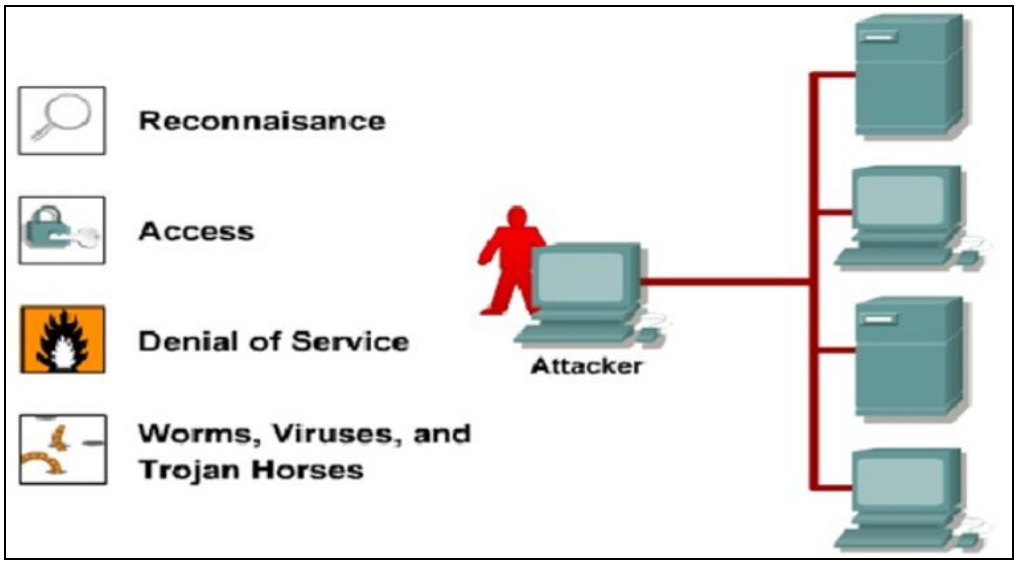

Ilustración 10. Principales ataques en la red.

Por tanto, al conocer los principales ataques y anomalías de seguridad existentes en la red, podemos destacar la importancia que tienen herramienta como lo sistemas de detecciones de intrusiones o IDS, ya que pretenden detectar a tiempo este tipo de ataques.

Por esta razón, vamos a evaluar Zeek IDS con las siguientes capturas de trafico de red o PCAP, basadas en ataques reales de seguridad, y verificar la correcta detección de los sucesos ocurridos en cada uno de estos casos. Estas capturas de tráfico han sido obtenidas del sitio web oficial de Malware-Traffic-Analysis (https://www.malware-traffic-analysis.net/training-exercises.html)

PCAP 1. Análisis del protocolo SMB.

En esta captura de tráfico, el **incidente de seguridad** consiste en un movimiento lateral por el protocolo SMB, causado a través de la extracción, creación y ejecución de un servicio o ejecutable malicioso como admin en un host infectado, a través de la botnet Trickbot. En resumen, se presentan los siguientes tipos de ataque:

- Conexión externa a sitios web con certificados SSL inválidos.
- Conexión externa a sitios maliciosos categorizado como botnet.
- Creación de servicio EXE malicioso.
- Ejecución de servicio EXE malicioso.
- Intento de propagación del malware en la red.
- Movimiento Lateral en la red.

Los **equipos locales** afectados son los siguientes:

- 10.6.26.110
- 10.6.26.6

Los posibles **sitios maliciosos** que hayan podido causar este ataque de seguridad son:

- 144.217.50.241
- 185.203.242.137
- 187.58.56.26
- 190.152.4.210

PCAP 2. Análisis Mondogreek.

En esta captura de tráfico, el **incidente de seguridad** consiste en que un equipo local Windows 10 de una organización, es infectado por el malware Trickbot, y desde ese equipo, se localizan intentos de propagación del malware a otros elementos de la red. En concreto, se presentan los siguientes tipos de ataque:

- Conexión externa a sitios web con certificados SSL inválidos.
- Conexión externa a sitios maliciosos categorizado como botnet.
- Detección de fichero cuyo hash es considerado Malware.
- Escaneos de red locales desde el equipo infectado.
- Intentos de propagación de malware en la red.

Los **equipos locales** afectados son los siguientes:

- 10.3.11.194
- 10.3.11.218
- 10.3.11.119

Los posibles **sitios maliciosos** que hayan podido causar este ataque de seguridad son:

- 45.148.120.153
- 51.254.164.244
- 190.214.13.2

- 185.14.31.98
- 203.176.135.102
- 64.44.133.131
- 199.247.13.144

PCAP 3. Análisis Spraline.

En esta captura de tráfico, el **incidente de seguridad** consiste en un equipo local que ha sido infectado por malware de tipo Urnsif, conocido también como Gozi, y Trickbot. En concreto, se presentan los siguientes tipos de ataque:

- Conexión externa a sitios web con certificados SSL inválidos.
- Conexión externa a sitios maliciosos categorizado como botnet.

Los **equipos locales** afectados son los siguientes:

- 10.8.20.101

Los posibles **sitios maliciosos** que hayan podido causar este ataque de seguridad son:

- 94.103.87.160
- 94.103.86.146
- 185.193.141.166
- 139.198.5.65
- 206.189.74.47
- 68.183.185.221
- 89.105.203.184
- 185.117.75.41
- 185.37.181.152
- 185.248.87.88
- 170.238.117.187
- 185.183.98.232

La comprobación del funcionamiento y las detecciones realizadas por Zeek se muestran en capítulos posteriores, donde nos permitirá evaluar adecuadamente las prestaciones de este IDS, así como la visualización de datos a través de ELK.

8. Proceso de configuración de tecnologías.

En esta sección se explicarán las diferentes configuraciones realizadas en el entorno de trabajo de este proyecto. En concreto, las configuraciones realizadas para Zeek IDS y el entorno Elasticsearch, Logstash y Kibana.

Estas configuraciones han sido realizadas con el fin de cumplir los objetivos de este proyecto, así como, lograr las detecciones específicas de los casos de uso de pruebas planteados en el apartado anterior.

8.1 Zeek IDS.

Tras el proceso de instalación seguido de Zeek IDS en el apartado 6.1 de este documento, podemos comenzar a realizar las configuraciones básicas que necesitamos para este proyecto.

La ruta predefinida de archivos de configuración de Zeek se encuentran en el siguiente path:

/usr/local/zeek/

Configuración de interfaz de red

En primer lugar, deberemos de configurar la **interfaz de red en la que Zeek permanecerá a la escucha** y analizará el tráfico entrante y saliente, en busca de patrones sospechosos de seguridad. En nuestro caso, configuramos la interfaz eth0, como la interfaz que se monitorizará en el IDS.

```
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly chang
ing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=eth0
```

/usr/local/zeek/etc/node.cfg

Integración de BZAR en Zeek IDS

En este proyecto, se va a realizar una **integración en Zeek IDS con el proyecto BZAR de Mitre ATT&CK**.

MITRE es una corporación no gubernamental cuyo objetivo es intentar resolver problemas que contribuyan a un mundo más seguro. Disponen de un framework llamado MITRE ATT&CK (por sus siglas en inglés, **Tácticas, Técnicas y Conocimiento Común de Adversarios**), consistente en una plataforma que organiza y categoriza los distintos tipos de ataques, amenazas y procedimientos realizados por los distintos atacantes en el mundo digital y que permite identificar vulnerabilidades en los sistemas informáticos. Este framework es muy útil, ya que permite modelar los posibles atacantes y los métodos utilizados para una organización concreta, permitiendo detectar **técnicas asociadas a grupos APT**. [21]

Un ejemplo de la matriz de Mitre ATT&CK anteriormente descrita:

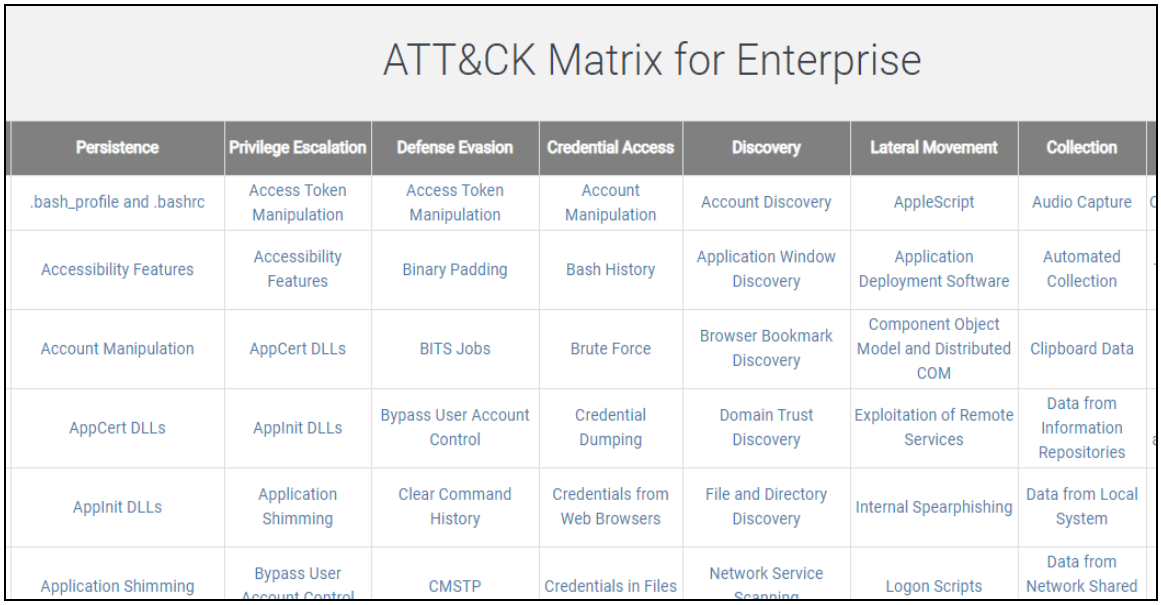

Ilustración 11. MITRE ATT&CK Matrix for Enterprise

El **proyecto BZAR** que integraremos en Zeek, consiste en un repositorio de scripts públicos escritos en lenguaje Zeek, con el cual se pretende añadir una capa más de inteligencia al IDS, permitiendo categorizar los patrones detectados por el IDS en un conjunto de técnicas y tácticas de ataque de seguridad. [21]

En concreto, el proyecto BZAR, nos permite detectar el siguiente conjunto de tácticas de ataque:

- Ejecución
- Persistencia
- Evasión de defensa
- Acceso de credenciales
- Descubrimiento
- Movimiento Lateral

Y para cada una de estas tácticas, el siguiente conjunto de técnicas específicas de ataque.

Execution	Persistence	Defense Evasion	Credential Access	Discovery	Lateral Movement
T1035 Service Execution	T1004 Winlogon Helper DLL	T1070 Indicator Removal Host	T1003 Credential Dumping	T1016 System Network Configuration	T1077 Windows Admin Shares
T1047 Windows Mgmt Instrum. (WMI)	T1013 Port Monitors			T1049 System Networks Connections	T1105 Remote File Copy
T1053 Scheduled Task				T1018 Remote System	
				T1033 System Owner/User	
				T1069 Permission Groups	
				T1082 System Info	
				T1083 File and Directory	
				T1087 Account	
				T1124 System Time	
				T1135 Network Share	

Ilustración 12. Bzar. Tácticas y técnicas MITRE.

Para su integración, deberemos de copiar el conjunto de scripts disponible en (https://github.com/mitre-attack/bzar), dentro de la ruta de configuración de políticas que cargará y utilizará Zeek IDS. Por tanto, dentro del path ***/usr/local/zeek/share/zeek/policy/bzar/*** definimos los siguientes scripts:

```
root@kali:/usr/local/zeek/share/zeek/policy/bzar# ls -lrth
total 144K
-rwxr-xr-x 1 root root  367 Mar 15 06:31 __load__.zeek
-rwxr-xr-x 1 root root  302 Mar 15 06:31 dpd.sig
-rwxr-xr-x 1 root root 9.5K Mar 15 06:31 bzar_smb.zeek
-rwxr-xr-x 1 root root 1.9K Mar 15 06:31 bzar_files.zeek
-rwxr-xr-x 1 root root 8.4K Mar 15 06:31 bzar_dce-rpc.zeek
-rwxr-xr-x 1 root root  99K Mar 15 06:31 bzar_dce-rpc_consts.zeek
-rwxr-xr-x 1 root root 7.3K Mar 15 06:39 main.zeek
```

Ilustración 13. Scripts BZAR para Zeek IDS.

Estos scripts, a la hora de ejecutar Zeek, generarán un "***notice.log***", en el cual podremos ver toda la información sobre el ataque detectado en la captura de tráfico analizada.

Integración feeds de botnets

Para la integración de lista de reputación de botnets, se ha desarrollado un script en Python, el cual se encarga de obtener la información más detallada sobre botnet del sitio (https://abuse.ch/), en concreto, se obtienen los siguientes feeds:

- **Blacklist IP de Botnet**: Consiste en una lista de bloqueo de direcciones IP asociadas con dichos C&C de botnet que se pueden usar para detectar y bloquear el tráfico C2 de botnet desde máquinas infectadas hacia Internet. Algunas de las botnets más frecuentes son Dridex, Heodo (también conocido como Emotet) y TrickBot.

- **Blacklist SSL**: Consiste en un proyecto de abuse.ch con el objetivo de detectar conexiones SSL maliciosas, mediante la identificación y la lista

negra de certificados SSL utilizados por los servidores de C&C de botnet. Además, identifica las huellas digitales JA3 que le ayudan a detectar y bloquear la comunicación C&C de la botnet de malware en la capa TCP.

El contenido del script es el siguiente:

```
import urllib.request
# BOTNET IP C&C
response = urllib.request.urlretrieve("https://feodotracker.abuse.ch/downl
oads/ipblocklist.csv","/usr/local/zeek/feeds/raw_botnet.txt")

file = open ("/usr/local/zeek/feeds/raw_botnet.txt","r")
alldata = file.readlines()
file.close()

file2write = open("/usr/local/zeek/feeds/botnet.txt","w")
file2write.write("#fields\tindicator\tmeta.desc\tindicator_type\tmeta.sour
ce\tmeta.url\n")
for data in alldata:
    if ("#" not in data):
        file2write.write(str(data.split(",")[1].strip())+"\t")
        file2write.write("botnet_feed\t")
        file2write.write("Intel::ADDR\t")
        file2write.write(str(data.split(",")[4].strip())+"\t")
        file2write.write("https://feodotracker.abuse.ch/downloads/ipblockl
ist.csv\n")
file2write.close()

# BOTNET SSL C&C
response = urllib.request.urlretrieve("https://sslbl.abuse.ch/blacklist/ss
lblacklist.csv","/usr/local/zeek/feeds/raw_botnet_ssl.txt")

file = open ("/usr/local/zeek/feeds/raw_botnet_ssl.txt","r")
alldata = file.readlines()
file.close()

file2write = open("/usr/local/zeek/feeds/botnet_ssl.txt","w")
file2write.write("#fields\tindicator\tmeta.desc\tindicator_type\tmeta.sour
ce\tmeta.url\n")
for data in alldata:
    if ("#" not in data):
        file2write.write(str(data.split(",")[1].strip())+"\t")
        file2write.write("botnet_feed\t")
        file2write.write("Intel::CERT_HASH\t")
        file2write.write(str(data.split(",")[2].strip())+"\t")
        file2write.write("https://feodotracker.abuse.ch/downloads/ipblockl
ist.csv\n")
file2write.close()
```

/usr/local/zeek/feeds/get_feeds_botnet.py

Para habilitar el módulo de Inteligencia de Zeek IDS, deberemos de desarrollar una regla en lenguaje Zeek, mediante la cual, se carguen los feeds obtenidos mediante el script. Esta regla tiene el siguiente contenido.

```
export {
    redef Intel::read_files += {
        "/usr/local/zeek/feeds/botnet.txt",
        "/usr/local/zeek/feeds/botnet_ssl.txt"
    };
}
```

/usr/local/zeek/share/zeek/policy/uoc/load_feeds.zeek

Esta regla permitirá que cuando se ejecute Zeek IDS, nos genere un "***Intel.log***", el cual contendrá la información de todos aquellos casos en los que se encuentre una coincidencia con las listas de reputación de botnets.

Configuración archivo Local.zeek

Se deberá de configurar también el archivo **local.zeek**, en el cual, definiremos todas aquellas políticas y configuraciones propias creadas para que a la hora de iniciar y ejecutar Zeek, disponga de estas configuraciones y aplique los métodos de análisis y detección deseados.

En concreto, deberemos de añadir las referencias a:

- Scripts del proyecto BZAR.
- Script de inteligencia sobre feeds de botnets.

El contenido de este fichero es el siguiente:

```
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

@load misc/loaded-scripts
@load tuning/defaults
@load misc/capture-loss
@load misc/stats
@load frameworks/software/vulnerable
@load frameworks/software/version-changes
@load-sigs frameworks/signatures/detect-windows-shells

# Default scripts.
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software
@load protocols/dns/detect-external-names
@load protocols/ftp/detect
@load protocols/conn/known-hosts
@load protocols/conn/known-services
@load protocols/ssl/known-certs
@load protocols/ssl/validate-certs
@load protocols/ssl/log-hostcerts-only
@load protocols/ssh/geo-data
@load protocols/ssh/detect-bruteforcing
@load protocols/ssh/interesting-hostnames
@load protocols/http/detect-sqli
```

```
@load frameworks/files/hash-all-files
@load frameworks/files/detect-MHR
@load policy/frameworks/notice/extend-email/hostnames

#Customize scripts
#BZAR
@load bzar/main.zeek
@load-sigs bzar/dpd.sig
@load bzar/bzar_smb.zeek
@load bzar/bzar_files.zeek
@load bzar/bzar_dce-rpc_consts.zeek
@load bzar/bzar_dce-rpc.zeek

#FEEDS & INTEL
@load uoc/load_feeds.zeek
@load policy/frameworks/intel/seen
```

/usr/local/zeek/share/zeek/site/local.zeek

Configuración File System

Para la mejor organización de las detecciones realizadas por Zeek una vez se analicen cada una de las capturas de trafico de los escenarios de prueba, se han creado los siguientes directorios dentro del path de configuraciones de Zeek.

- Para los resultados de análisis del PCAP 1:
  - /usr/local/zeek/smb_lateral_movement
- Para los resultados de análisis del PCAP 2:
  - /usr/local/zeek/mondogreek
- Para los resultados de análisis del PCAP 3:
  - /usr/local/zeek/spraline

Ejecución de Zeek IDS

Una vez iniciado el servicio Zeek IDS, procederemos a analizar las capturas de tráfico mencionadas en el capítulo anterior, dentro del directorio creado para cada uno de ellos.

La forma de ejecutar Zeek y analizar los PCAP será de la siguiente forma:

```
root@kali:/usr/local/zeek/mondogreek # zeek -r /home/kali/Downloads/2020-03-
14-traffic-analysis-exercise.pcap /usr/local/zeek/share/zeek/site/local.zeek
```

Zeek generara una serie de logs en base al análisis y detecciones realizadas, como se muestra a continuación.

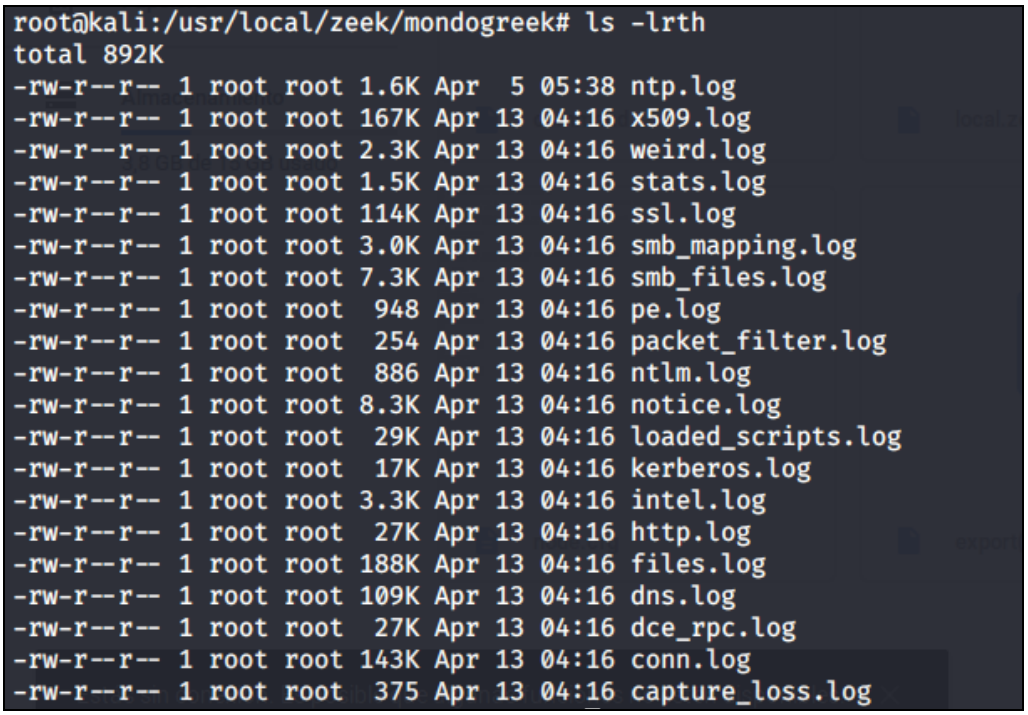

Ilustración 14. Muestra de logs generados por Zeek IDS.

De todos los logs generados por Zeek, nos centraremos en **notice.log** e **intel.log**, ya que son los logs que más información aportan sobre los ataques de seguridad ocurridos y motivos del incidente de seguridad analizado.

En los apartados siguientes, comentaremos las configuraciones realizadas para el envío de estos logs a la otra máquina de nuestro entorno donde se almacenarán e indexarán dichos logs.

8.2 Logstash.

Logstash se encuentra instalado en la máquina virtual 1 del entorno de trabajo, junto a Zeek IDS.

Mediante Logstash, vamos a definir el envío de logs generados por Zeek hacia la máquina virtual 2 del entorno, para que estos logs sean almacenados en Elasticsearch, y poder tratarlos y analizarlos en Kibana.

Para lograr el funcionamiento descrito anteriormente, deberemos de realizar una serie de configuraciones que iremos describiendo a continuación.

El path por defecto para las configuraciones de Logstash es el siguiente:

/etc/logstash/

Configuración de archivo logstash.yml

Este archivo contiene la configuración básica de funcionamiento de la herramienta Logstash.

En nuestro caso, este archivo no será necesario modificarlo, ya que la configuración proporcionada por defecto se adapta a las necesidades del entorno de este proyecto.

En concreto, se pueden configurar ciertos parámetros, como los siguientes:

- Data path:
  - /var/lib/logstash
- Debugging settings:
  - /var/log/logstash
- Node identity.
- Pipeline settings and configuration.
- Module settings.
- Cloud settings.
- X-pack settings.
- Queue settings.
- Metrics settings.

Configuración de archivo pipelines.yml

Por defecto logstash, al iniciarse y ejecutarse, utiliza para todos los envíos de logs definidos una única tubería o pipeline para realizar esos envíos de datos.

En el caso de ser necesario disponer más de un pipeline para el envío de datos, será necesario configurarlos en el archivo mostrado a continuación.

Dado que en nuestro caso, vamos a enviar dos tipos de logs diferentes y cada uno de ellos tendrá un tratado distinto, definiremos dos pipelines.

```
# This file is where you define your pipelines. You can define multiple.
# For more information on multiple pipelines, see the documentation:
#   https://www.elastic.co/guide/en/logstash/current/multiple-
pipelines.html

- pipeline.id: notice
  path.config: "/etc/logstash/conf.d/notice_log.conf"

- pipeline.id: intel
  path.config: "/etc/logstash/conf.d/intel_log.conf"
```

/etc/logstash/pipelines.yml

Configuraciones de envío de logs de Zeek IDS

Necesitaremos enviar dos tipos de logs diferentes generados por Zeek IDS.

- Notice.log
- Intel.log

Para ello, dentro del path de configuraciones personalizadas para logstash, definimos dos archivos de configuración para definir el envío de estos logs

hacia el índice definido en Elasticsearch. Estas configuraciones las haremos en:

/etc/logstash/conf.d/

Dentro de estos archivos de configuración deberemos de definir:

- **Inputs**: Serán aquellos logs generados por Zeek que se pretender leer y enviar a Elasticsearch. En nuestro caso tendremos 3 tipos de logs notice, y tres tipos de logs Intel, uno por cada una de las capturas de tráfico analizadas.

- **Filter**: En este apartado definiremos como se tratarán y parsearán los logs anteriores, teniendo en cuenta, que los logs generados por Zeek tienen formato CSV, utilizando como separador el tabulador.
    - Excluir del envío todas aquellas líneas del log que comiencen por #.

    - Aplicar filtros grok, mediante los cuales, se realice el parseo de los diferentes campos que componen el log, para que a la hora de indexarlo en Elasticsearch, se guarden los campos correspondientes.

    - Aplicar filtro geoip, para añadir información de geolocalización a aquellas direcciones IP destino encontradas en las comunicaciones.

    - Aplicar como fecha de indexación la que proporciona la captura de tráfico sobre la fecha real de los eventos, y evitar como fecha, la del tiempo de indexación en Elasticsearch,

- **Outputs**: Definiremos el destino donde se pretenden enviar los logs, en nuestro caso:
    - Hacia el host de la máquina virtual 2 donde se encuentra Elasticsearch.

    - El índice de Elasticsearch donde se almacenarán cada uno de los logs.

El contenido de los archivos de configuración para los logs de tipo Notice.log e Intel.log se muestran en detalle en los anexos de este documento.

8.3 Elasticsearch

En Elasticsearch definiremos todas las configuraciones necesarias para almacenar e indexar todos los logs provenientes de Zeek IDS. El path por defecto de configuración de Elasticsearch es el siguiente:

/etc/elasticsearch/

En concreto definiremos las siguientes configuraciones.

Configuración de índices de datos

Un índice en Elasticsearch se define como una base de datos relacional, la cual dispone de un mapeo que define múltiples tipos. Es decir, un índice es un espacio de nombres lógico que se asigna a uno o más fragmentos primarios y puede tener cero o más fragmentos de réplica.

En este proyecto, trabajaremos con **dos índices de datos**:

- Logstash-notice-log
- Logstash-intel-log

Estos índices serán creados de forma automática al proceder de logstash, a los cuales se les aplicará el mapeo de configuraciones por defecto definido para logs de tipo “logstash”.

Configuración de archivo elasticsearch.yml

En este archivo de configuración se defines todas aquellas configuraciones para que Elasticsearch sea capaz de establecer la **conexión con el recolector de logs llamada Logstash** dispone en la máquina virtual 1 de este entorno, así como para definir las **configuraciones de email del módulo x-pack para el envío de alertas mediante el módulo Watcher** a través de un correo electrónico creado para tal efecto.

Dispondremos de dos correos electrónicos para las alertas:

- Un correo electrónico creado específico para el envío de alertas desde Watcher.
- Correo electrónico propio personal, el cual recibirá todas aquellas alertas configuradas.

Este archivo de configuración se muestra en detalle en los anexos de este documento:

Para poder realizar el envío de correos electrónicos mediante alertas en el entorno ELK, además de la configuración añadida en el archivo anterior, deberemos de configurar en Elasticsearch la password de dicho correo electrónico. Para ello, ejecutamos la siguiente sentencia en línea de comandos, e introducimos la contraseña de dicho correo electrónico:

```
root@kali:/usr/share/elasticsearch/bin/elasticsearch-keystore add
xpack.notification.email.account.gmail_account.smtp.secure_password
```

8.4 Kibana

En Kibana nos encargaremos de definir todas aquellas **visualizaciones y Dashboards** que nos permitan analizar todas las anomalías de seguridad detectadas en los logs proporcionados por Zeek IDS, los cuales se encontraran ya almacenados en sus índices correspondientes en Elasticsearch. Además, se definirán el conjunto de **alertas sobre detecciones de conexiones hacia sitios maliciosos** categorizados de tipi botnet.

El path por defecto de configuraciones de Kibana:

/etc/kibana/

Configuración de archivo kibana.yml

En este archivo de configuración procederemos a definir como se realizará la conexión entre Elasticsearch y Kibana para que podamos ver en este todos los datos almacenados en los índices de datos de Elasticsearch.

El contenido del archivo de configuración se muestra en el anexo de este documento.

Configuración de plantillas de índices

Una vez se haya ejecutado Zeek y se haya realizado el envío y almacenado de los logs por medio de Logstash hacia Elasticsearch, se definirán los llamados **index pattern**, a través de los cuales podremos realizar búsquedas de datos dentro de la pestaña "**Discover**" de Kibana, así como utilizar estos modelos o plantillas, a la hora de crear visualizaciones.

En concreto, crearemos dos plantillas para los índices de datos:

- Logstash-notice
- Logstash-intel

Para crearlos, deberemos de hacerlo desde la opción "**Managment**" y en la sección "Index Pattern", el cual, automáticamente, detectará los índices de datos que definimos en el apartado anterior.

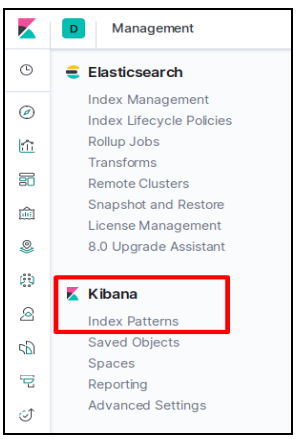

Ilustración 15. Menú Kibana. Index Patterns

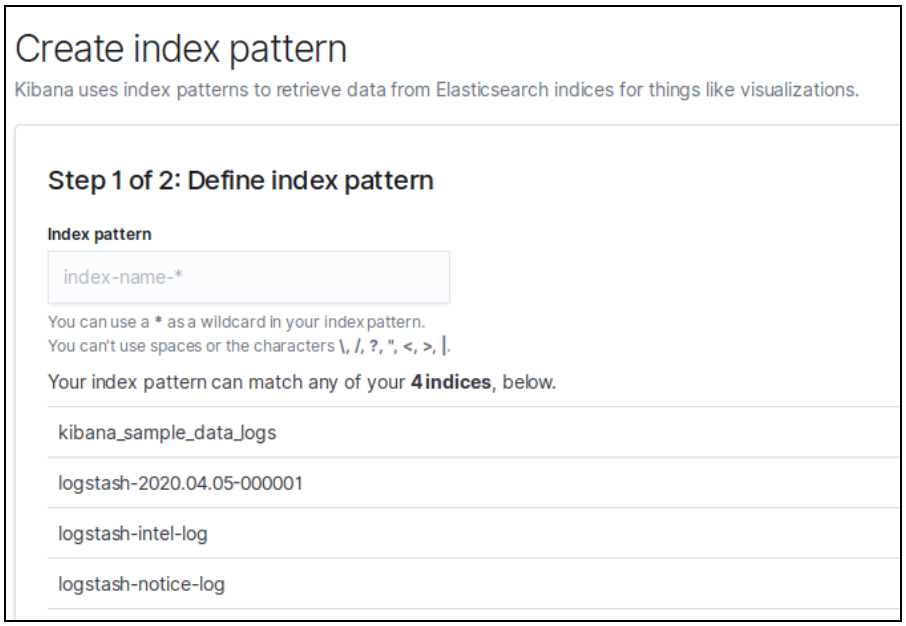

Ilustración 16. Creación de Index Patterns en Kibana.

Definición de búsquedas de datos

Una vez creados los index pattern anteriores, podremos realizar búsquedas de datos a través de ellos, y mediante filtros, guardar una serie de búsquedas, para que estas puedan ser utilizadas también a la hora de crear visualizaciones más personalizadas.

Para guardar búsquedas de datos personalizados, debemos de hacerlo desde la pestaña "**Discover**" de Kibana.

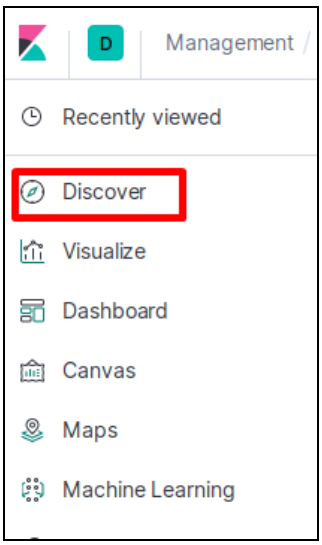

Ilustración 17. Menú Kibana. Discover.

Desde el cuadro de mandos abiertos, podremos realizar búsquedas y filtros personalizados en base a cada uno de los campos que componen los logs. En nuestro caso, deberemos de guardar seis búsquedas personalizadas, filtradas

por el atributo **path**, para diferenciar el tipo de log y a que captura de trafico de red pertenece.

En concreto, hemos creado las siguientes búsquedas:

- **Notice_smb**: Para el notice.log del PCAP 1 analizado.
- **Notice_mondogreek**: Para el notice.log del PCAP 2 analizado.
- **Notice_spraline**: Para el notice.log del PCAP 3 analizado.
- **Intel_smb**: Para el intel.log del PCAP 1 analizado.
- **Intel_mondogreek:** Para el intel.log del PCAP 2 analizado.
- **Intel_spraline**: Para el intel.log del PCAP 3 analizado.

Para guardar este tipo de búsquedas, deberemos de añadir el filtro por path, deseado, y seleccionar la opción "**Save**".

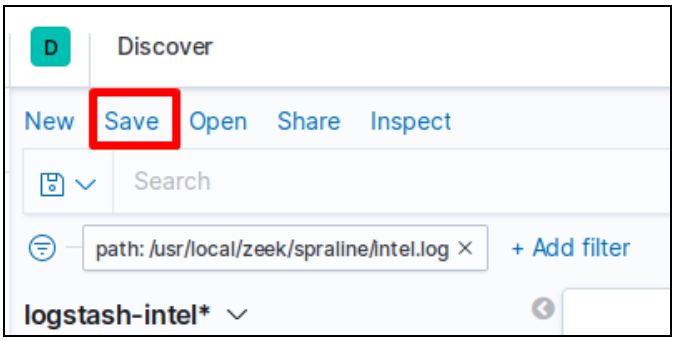

Ilustración 18. Guardar búsquedas de datos en Kibana.

Configuración de Dashboards y visualizaciones

Para realizar un análisis de los resultados obtenidos por Zeek IDS, procedemos a crear una serie de Dashboard formados por una serie de visualización concretas de datos.

En primer lugar, creamos todas las visualizaciones que deseemos añadir posteriormente a un Dashboard. Para ellos, deberemos de seleccionar la opción "**Visualize**" del menú de Kibana, y dentro de este, seleccionar el tipo de visualización que más se adapte a lo que se pretenda representar.

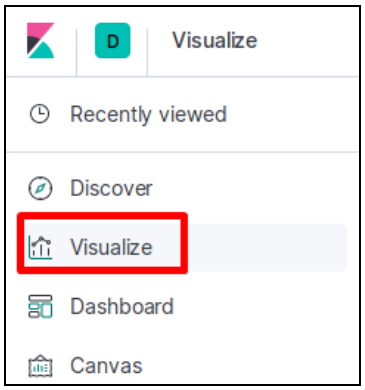

Ilustración 19. Menú Kibana. Visualizaciones.

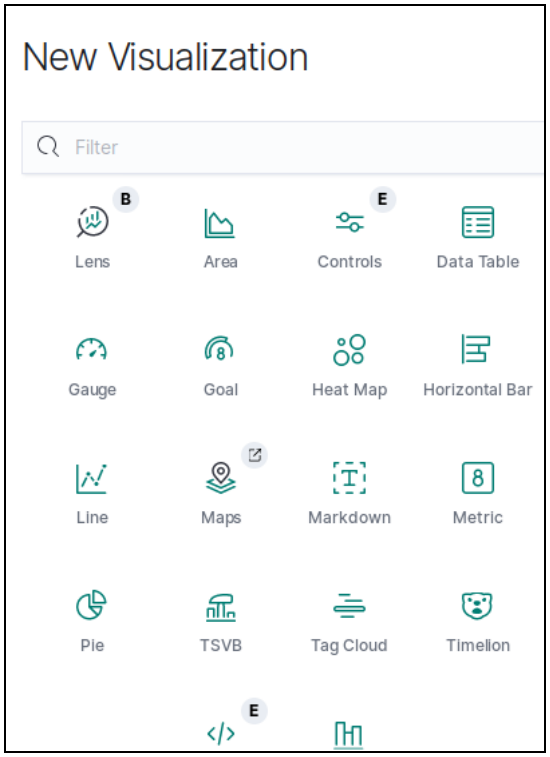

Ilustración 20. Tipos de visualizaciones en Kibana.

Una vez creadas todas las visualizaciones, para crear un Dashboard, deberemos de seleccionar la pestaña "**Dashboard**" del menú de Kibana, y dentro de él, se procederá a añadir el conjunto de visualizaciones creadas mediante el método anteriormente descrito.

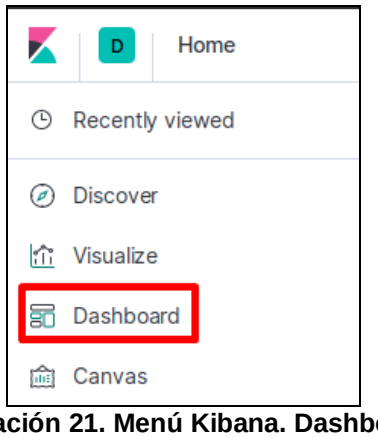

Ilustración 21. Menú Kibana. Dashboard.

En este proyecto, se van a crear dos tipos de Dashboard, para cada uno de los PCAP analizados en los escenarios de prueba. En concreto, se han creado los siguientes tipos de Dashboard:

- **Dashboard sobre logs de tipo Notice.** Los cuales están formados por las siguientes visualizaciones:
    - **Gráfico circular**: Representando los tipos de ataques de seguridad detectados.
    - **Gráfico circular**: Mostrando el mensaje principal de los ataques de seguridad detectados.
    - **Mapa del mundo**: Mostrando la ubicación de las direcciones IP destino con las que se han establecido conexiones en la captura de trafico de red.
    - **Tablas de datos:** Mediante las cuales se representa información del tipo:
        - Direcciones IP origen de comunicaciones.
        - Puertos origen de comunicaciones.
        - Direcciones IP destino de comunicaciones.
        - Puertos destino de comunicaciones.
        - Posibles ficheros maliciosos detectados.

- **Dashboard sobre logs de tipo Intel.** Los cuales están formados por:
    - **Gráfico circular**: Mostrando los tipos de botnets a las cuales se han detectado conexiones en el tráfico de red analizado.
    - **Mapa del mundo:** Representando la geolocalización de las direcciones IP maliciosas categorizadas como botnet.
    - **Tabla de datos**: Mostrando aquellas direcciones IP origen que establecieron comunicaciones con botnet, y que por tanto, posiblemente se encuentren infectadas.

Los resultados obtenidos y las muestras de los Dashboard creados se mostrarán en el capítulo siguiente, donde se evalúen los resultados y datos obtenidos.

Configuración de alertas.

Por último, en Kibana, se definirán una serie de alertas para que están envíen notificaciones sobre las detecciones de conexiones a botnets detectadas en cada una de las capturas de tráfico de red analizadas en el escenario de pruebas.

Se pretende dar un funcionamiento de tipo SIEM al entorno ELK en base a este conjunto de alertas configuradas.

Estas alertas se configuran y gestionan en Kibana desde la pestaña “**Managment**”, dentro del módulo llamado “Watcher”.

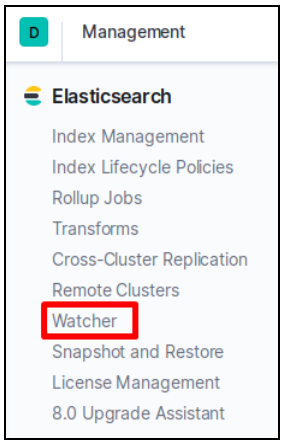

Ilustración 22. Menú Kibana. Watcher

Previamente de habilitar y configurar alertas, es necesario configurar la licencia de prueba gratuita de Kibana para que se pueda utilizar el módulo de alarmado de tipo Watcher, tal y como se muestra a continuación.

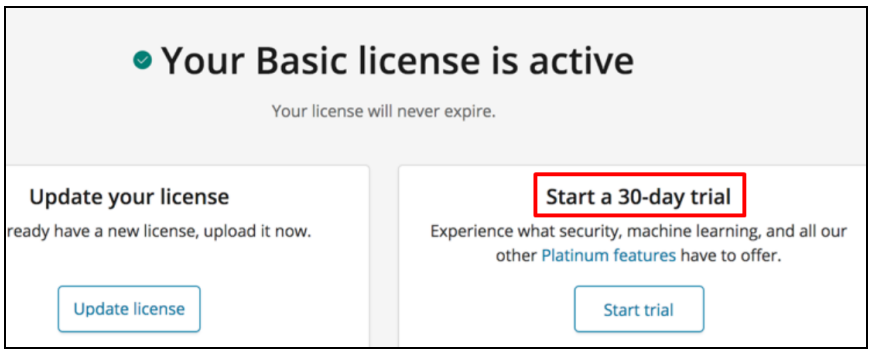

Ilustración 23. Activar licencia Kibana.

Por tanto, se han creado tres tipos de alertas. Los archivos de configuración de estas alertas se muestran en detalle en los anexos de este documento:

- **[ELK] Movimiento lateral en protocolo SMB**: Esta alerta notificará sobre las conexiones a botnets detectadas en el PCAP 1 analizado.

- **[ELK] Malware detectado en PCAP Mondogreek:** Esta alerta notificará sobre las conexiones a botnets detectadas en el PCAP 2 analizado.

- **[ELK] Malware detectado en PCAP Spraline**: Esta alerta notificará sobre las conexiones a botnets detectadas en el PCAP 3 analizado.

Estas alertas, además de un texto descriptivo incluyendo las botnets hacia las cuales se han detectado las conexiones, incluye un reporte PDF con las direcciones IP internas que se han visto infectadas por estas conexiones.

El resultado de estas alertas se mostrará en el capítulo posterior donde se evaluarán y analizarán todos los resultados obtenidos por la herramienta.

9. Evaluación de prestaciones de la herramienta.

En este capítulo, se mostrarán las pruebas realizadas sobre los escenarios planteados en el capítulo 7 de este documento, en las cuales se analizarán los paquetes de trafico de red seleccionados.

Estos serán analizados con Zeek IDS, el cual detectará los principales problemas de seguridad existentes en ellos, generando una serie de logs que contienen toda esta información.

Dichos logs, serán enviados mediante Logstash al entorno de la segunda máquina virtual, donde se encuentra instalado Elasticsearch y Kibana, mediante los cuales, podremos realizar un análisis de los resultados obtenidos, así como la configuración de Dashboard y alertas de seguridad.

En cada uno de los subapartados siguientes, se mostrará para cada paquete de tráfico a analizar, el proceso de envío y recepción de datos, así como la visualización de los Dashboard configurados y los resultados de las alertas configuradas.

9.1 Análisis del protocolo SMB.

El paquete de trafico de red a analizar se llama "**smb_spread-lateralmovement_activity.pcap**" el cual se encuentra en el directorio de descargas del entorno de trabajo de la máquina virtual 1.

Procedemos a **ejecutar Zeek IDS**, para el que utilizaremos el siguiente comando.

```
zeek -r /home/kali/Downloads/smb_spread-lateralmovement_activity.pcap /usr/local/zeek/share/zeek/site/local.zeek
```

Ilustración 24. Ejecución Zeek IDS análisis protocolo SMB

Los **logs generados por Zeek** son los siguientes:

```
drwxr-xr-x 2 root root  28K Apr 13 04:23 extract_files
-rw-r--r-- 1 root root  15K Apr 13 04:23 x509.log
-rw-r--r-- 1 root root  805 Apr 13 04:23 weird.log
-rw-r--r-- 1 root root 1.8K Apr 13 04:23 stats.log
-rw-r--r-- 1 root root  13K Apr 13 04:23 ssl.log
-rw-r--r-- 1 root root 2.3K Apr 13 04:23 smb_mapping.log
-rw-r--r-- 1 root root 4.7K Apr 13 04:23 smb_files.log
-rw-r--r-- 1 root root 1.7K Apr 13 04:23 pe.log
-rw-r--r-- 1 root root  254 Apr 13 04:23 packet_filter.log
-rw-r--r-- 1 root root 2.0K Apr 13 04:23 ntp.log
-rw-r--r-- 1 root root  499 Apr 13 04:23 ntlm.log
-rw-r--r-- 1 root root  65K Apr 13 04:23 notice.log
-rw-r--r-- 1 root root  29K Apr 13 04:23 loaded_scripts.log
-rw-r--r-- 1 root root 5.8K Apr 13 04:23 kerberos.log
-rw-r--r-- 1 root root 2.6K Apr 13 04:23 intel.log
-rw-r--r-- 1 root root 4.3K Apr 13 04:23 http.log
-rw-r--r-- 1 root root  19K Apr 13 04:23 files.log
-rw-r--r-- 1 root root  29K Apr 13 04:23 dns.log
-rw-r--r-- 1 root root 1.2K Apr 13 04:23 dhcp.log
-rw-r--r-- 1 root root 4.4K Apr 13 04:23 dce_rpc.log
-rw-r--r-- 1 root root  39K Apr 13 04:23 conn.log
-rw-r--r-- 1 root root  458 Apr 13 04:23 capture_loss.log
```

Ilustración 25. Logs Zeek IDS. Análisis protocolo SMB.

De todos ellos, mediante Logstash se enviarán los siguientes logs a la máquina virtual 2 del entorno de trabajo:

- Notice.log
- Intel.log

Podemos observar cómo mediante Logstash, se encuentran creados los **pipelines** que se encargarán de enviar los datos hacia Elasticsearch.

```
root@kali:/usr/local/zeek/smb_lateral_movement# lsof -i -P -n
COMMAND     PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
NetworkMa   481     root   22u  IPv4  18039      0t0  UDP 192.168.100.146:68→192.168.100.1:67
java       1731 logstash   83u  IPv6  27968      0t0  TCP 192.168.100.146:34076→192.168.100.147:9200 (ESTABLISHED)
java       1731 logstash   84u  IPv6  28965      0t0  TCP 192.168.100.146:34078→192.168.100.147:9200 (ESTABLISHED)
```

Ilustración 26. Pipelines de conexión entre Logstash y Elasticsearch.

Una vez ejecutado Zeek IDS, y realizado el envío de datos de forma transparente mediante Logstash, verificamos en el buscador de datos de Kibana, como los datos han sido indexados y almacenados correctamente en el índice "**logstash-notice-log**" y "**logstash-intel-log**".

Ilustración 27. Logs Zeek IDS visibles en Elasticsearch y Kibana. Análisis protocolo SMB.

Tras verificar el funcionamiento correcto del flujo de envío de datos mencionado anteriormente, procedemos a analizar los resultados mostrados por los dashboard configurados.

Sobre el **dashboard [Notice] SMB Lateral Movement** obtenemos los siguientes resultados:

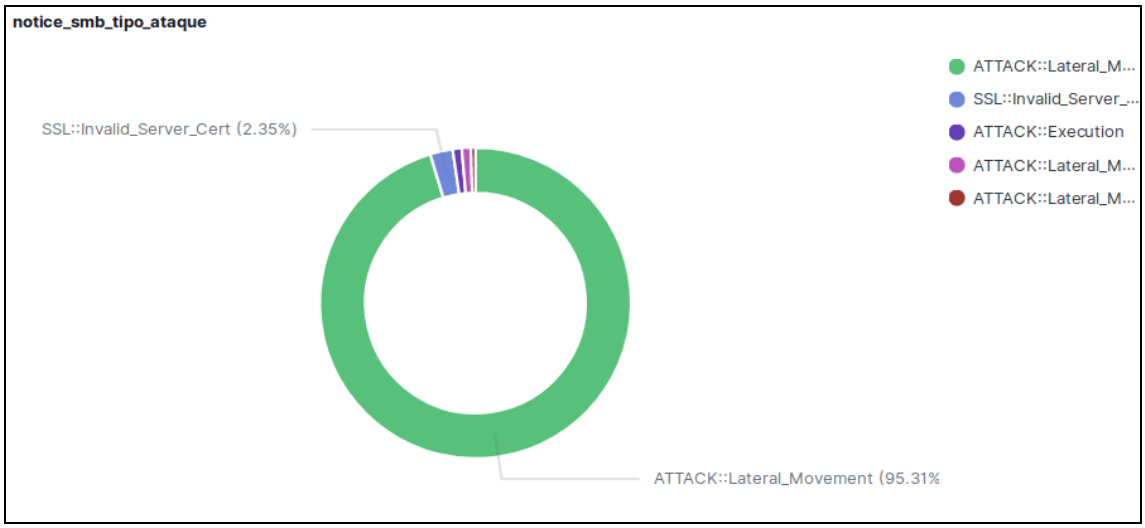

Ilustración 28. Análisis del protocolo SMB. Notice Log. Tipo de ataque.

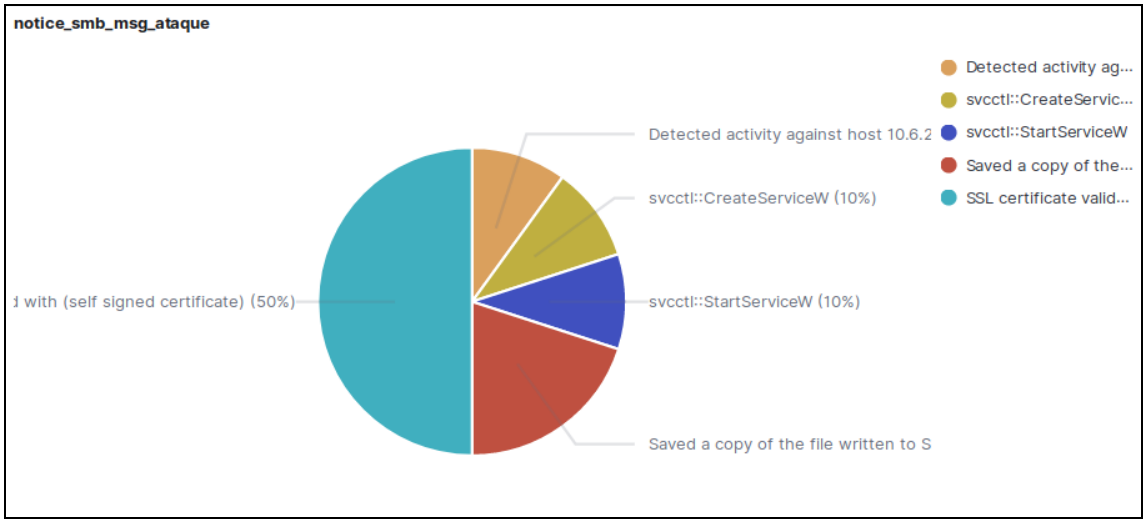

Ilustración 29. Análisis del protocolo SMB. Notice Log. Información del ataque.

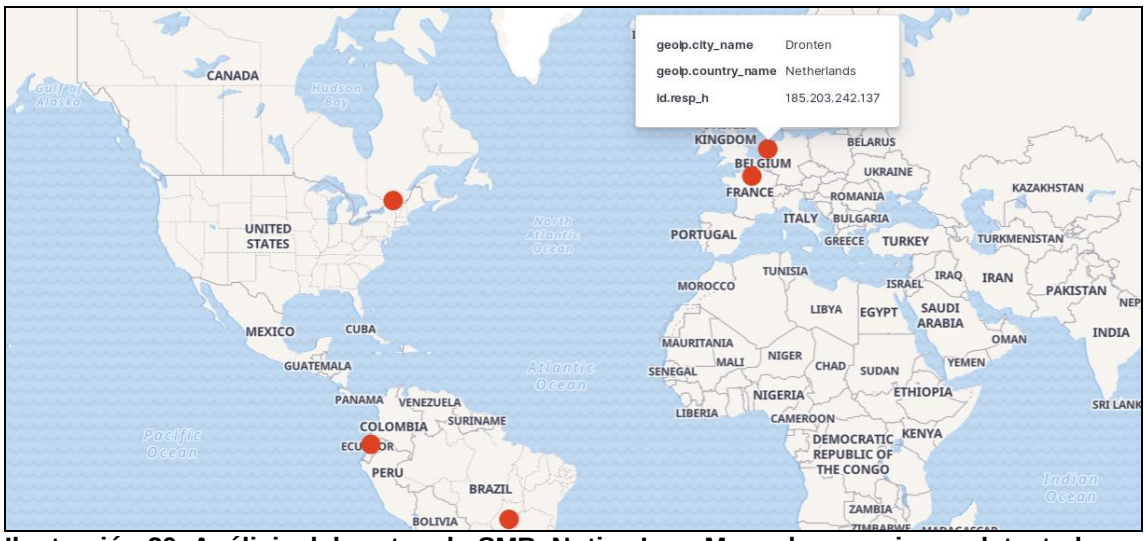

Ilustración 30. Análisis del protocolo SMB. Notice Log. Mapa de conexiones detectadas.

notice_smb_src_ip

Source IP	Count
10.6.26.110	210
10.6.26.6	2
	212

Ilustración 31. Análisis del protocolo SMB. Notice Log. Direcciones IP Origen.

notice_smb_source_port

Source Port	Count
49250	207
49207	1
49223	1
49568	1
50396	1
	211

Ilustración 32. Análisis del protocolo SMB. Notice Log. Puertos Origen.

notice_smb_destination_ip

Destination IP	Count
10.6.26.6	207
144.217.50.241	1
185.203.242.137	1
187.58.56.26	1
190.152.4.210	1
	211

Ilustración 33. Análisis del protocolo SMB. Notice Log. Direcciones IP Destino.

notice_smb_destination_port

Destination Port	Count
445	207
447	2
449	2
443	1
	212

Ilustración 34. Análisis del protocolo SMB. Notice Log. Puertos Destino.

notice_smb_files

Malicious Files	Count
\\WINDOWS\\d0p2nc6ka3f_fixhohlycj4ovqfcy_smchzo_ub83urjpphrwahjwhv_o5c0fvf6.exe	1
\\WINDOWS\\oiku9bu68cxqenfmcsos2aek6t07_guuisgxhllixv8dx2eemqddnhyh46l8n_dl.exe	1
	2

Ilustración 35. Análisis del protocolo SMB. Notice Log. Ficheros maliciosos.

En base a los análisis de los resultados obtenidos por estos dashboard, podemos concluir, que en esta captura de tráfico el principal ataque de seguridad detectado trata de **un ataque de movimiento lateral, causado principalmente por la creación y ejecución de un servicio/proceso malicioso** a través de uno de los ficheros maliciosos detectados. Además, podemos comprobar las principales direcciones IP origen y destino que intervienen en las comunicaciones.

En el **dashboard [Intel] SMB Lateral Movement**, procedemos a analizar los resultados del log intel para este paquete de trafico de red, donde encontramos los siguientes dashboard configurados:

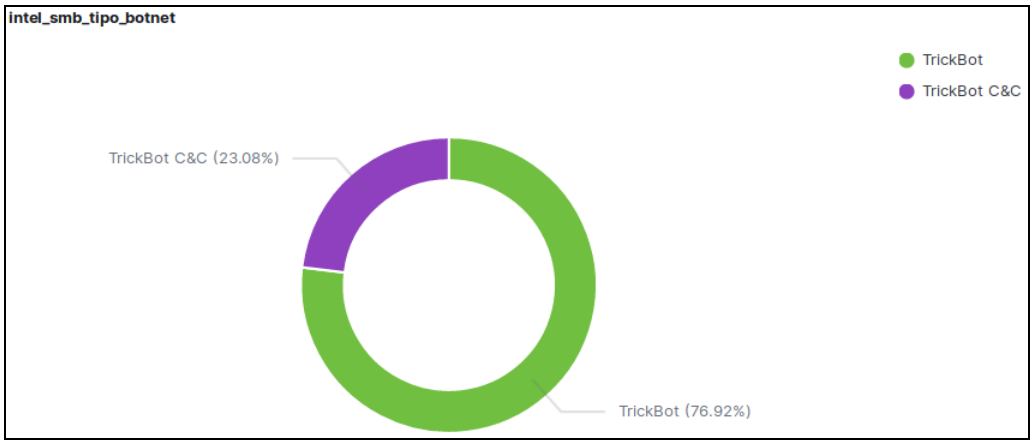

Ilustración 36. Análisis del protocolo SMB. Intel Log. Tipo de botnet detectada.

intel_smb_src_ip

Source IP	C&C connections
10.6.26.110	10
10.6.26.6	3
	13

Ilustración 37. Análisis del protocolo SMB. Intel Log. Direcciones IP Origen infectadas.

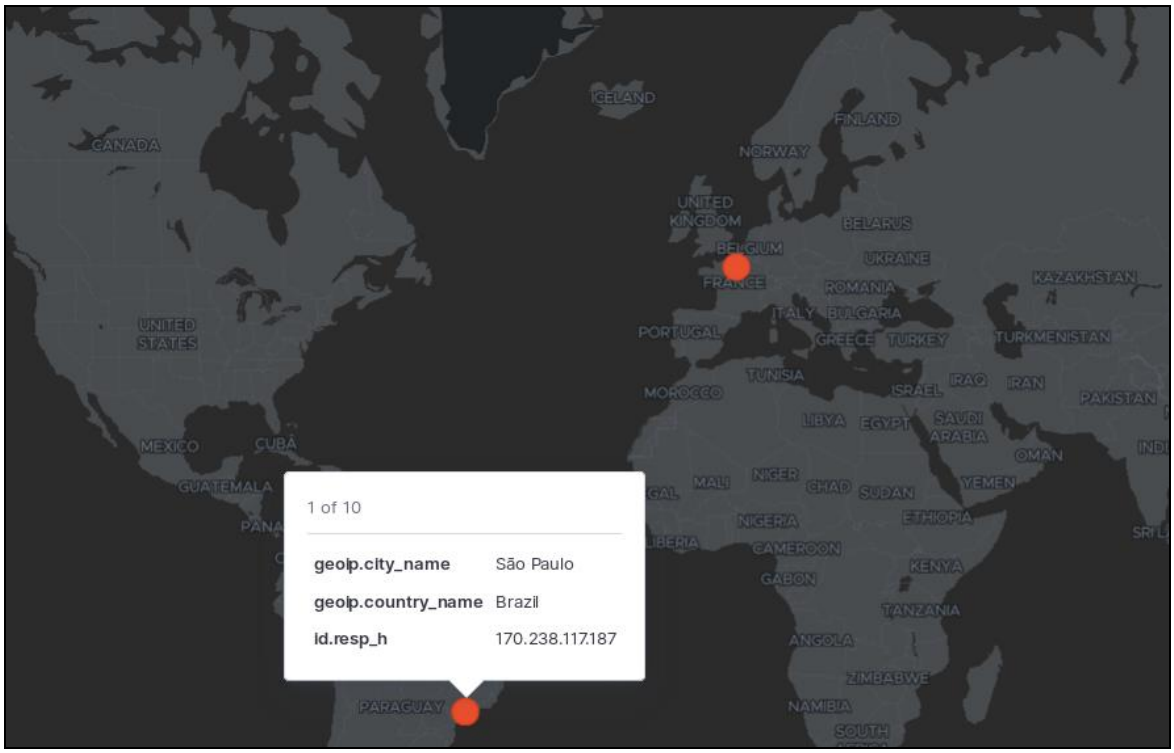

Ilustración 38. Análisis del protocolo SMB. Intel Log. Geolocalización de botnets.

En estas visualizaciones podemos verificar que los ficheros maliciosos anteriormente detectados, han sido obtenidos en base a las conexiones a sitios maliciosos de tipo **botnet, categorizados como Trickbot**, a través del cual se ha detectado el ataque mencionado anteriormente.

Por último, se definió una **alerta de tipo Watcher en Kibana** a través de la cual se notificará mediante correo electrónico de las detecciones de botnet realizadas en el análisis de logs proporcionados por Zeek IDS. La alerta configurada generó el siguiente correo electrónico.

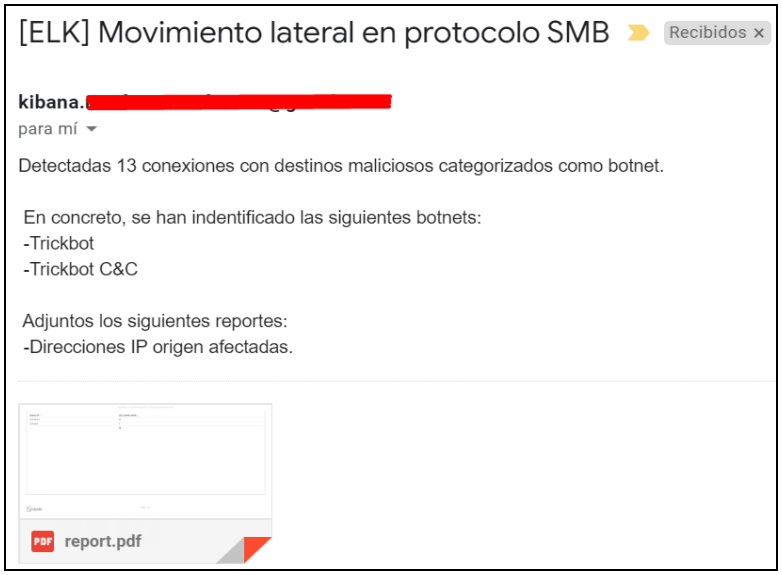

Ilustración 39. Análisis del protocolo SMB. Notificación email.

9.2 Análisis del paquete de red “Mondogreek”.

El paquete de trafico de red a analizar se llama “**2020-03-14-traffic-analysis-exercise.pcap”** el cual se encuentra en el directorio de descargas del entorno de trabajo de la máquina virtual 1.

Ejecutamos Zeek IDS mediante el comando:

```
zeek -r /home/kali/Downloads/2020-03-14-traffic-analysis-exercise.pcap /usr/local/zeek/share/zeek/site/local.zeek
```

Ilustración 40. Ejecución Zeek IDS análisis Mondogreek

Una vez ejecutado Zeek IDS, y realizado el envío de datos de forma transparente mediante Logstash, verificamos en el buscador de datos de Kibana, como los datos han sido indexados y almacenados correctamente en el índice “**logstash-notice-log**” y “**logstash-intel-log**”.

Ilustración 41. Logs Zeek IDS visibles en Elasticsearch y Kibana. Análisis Mondogreek

Sobre el dashboard **[Notice] Mondogreek** obtenemos los siguientes resultados:

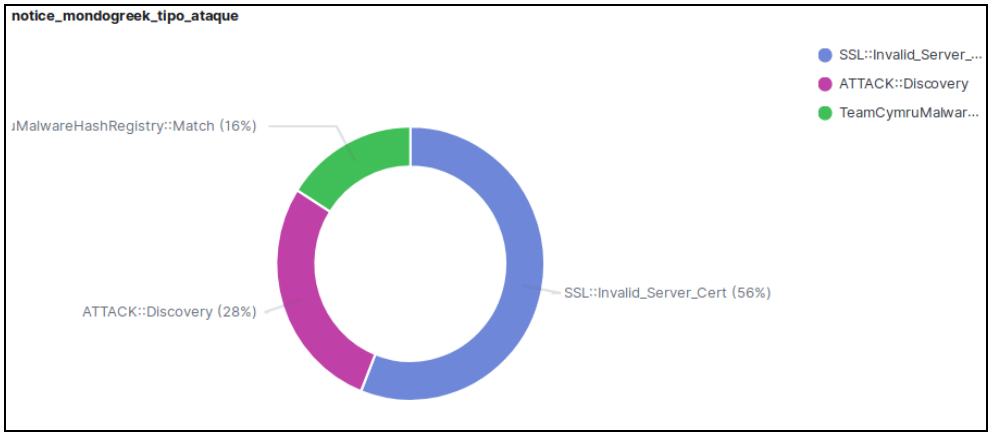

Ilustración 42. Análisis Mondogreek. Notice Log. Tipo de ataque

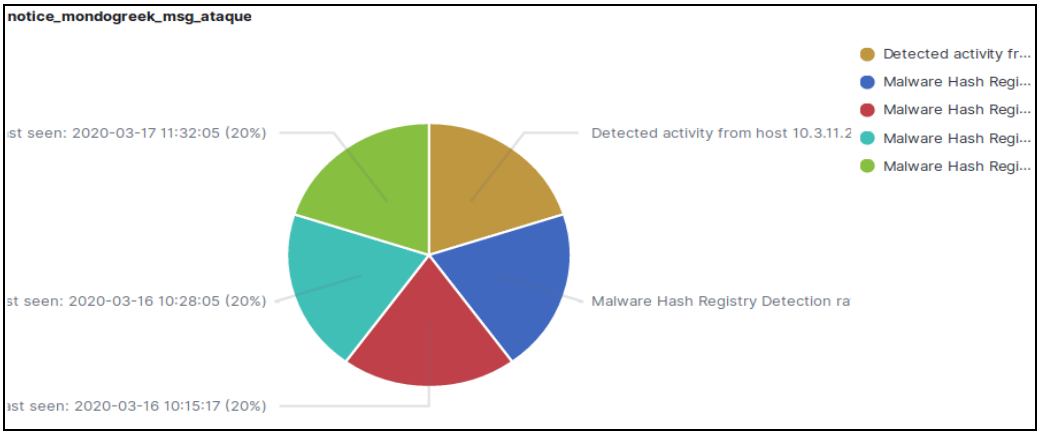

Ilustración 43. Análisis Mondogreek. Notice Log. Información del ataque.

Ilustración 44. Análisis Mondogreek. Notice Log. Mapa de conexiones detectadas.

notice_mondogreek_src_ip

Source IP	Count
10.3.11.194	10
10.3.11.218	6
10.3.11.119	2
	18

Ilustración 45. Análisis Mondogreek. Notice Log. Direcciones IP Origen.

notice_mondogreek_src_port

Source Port	Count
49816	2
49699	1
49709	1
49710	1
49715	1
	6

Ilustración 46. Análisis Mondogreek. Notice Log. Puertos Origen.

notice_mondogreek_dst_ip

Destination IP	Count
64.44.133.131	3
111.221.29.40	1
13.68.93.109	1
13.78.168.230	1
13.78.177.144	1
	7

Ilustración 47. Análisis Mondogreek. Notice Log. Direcciones IP Destino.

notice_mondogreek_dst_port

Destination Port	Count
443	12
80	4
447	1
449	1
	18

Ilustración 48. Análisis Mondogreek. Notice Log. Puertos Destino.

notice_mondogreek_files

Malicious Files	Count
http://64.44.133.131/images/cursor.png	2
http://64.44.133.131/images/imgpaper.png	1
http://bolton-tech.com/YAS20.exe	1
	4

Ilustración 49. Análisis Mondogreek. Notice Log. Ficheros maliciosos.

En base a los análisis de los resultados obtenidos por estos dashboard, podemos concluir, que en esta captura de tráfico el principal ataque de seguridad detectado consiste en la **infección por malware de un conjunto de equipos internos, a causa de la descarga y ejecución de los ficheros maliciosos** mostrados anteriormente.

También se detecta como este malware se intenta propagar en base a una serie de actividad de escaneos detectados posteriormente a la infección de malware.

Además, podemos comprobar las principales direcciones IP origen y destino que intervienen en las comunicaciones.

En el dashboard **[Intel] Mondogreek** encontramos los siguientes dashboard configurados:

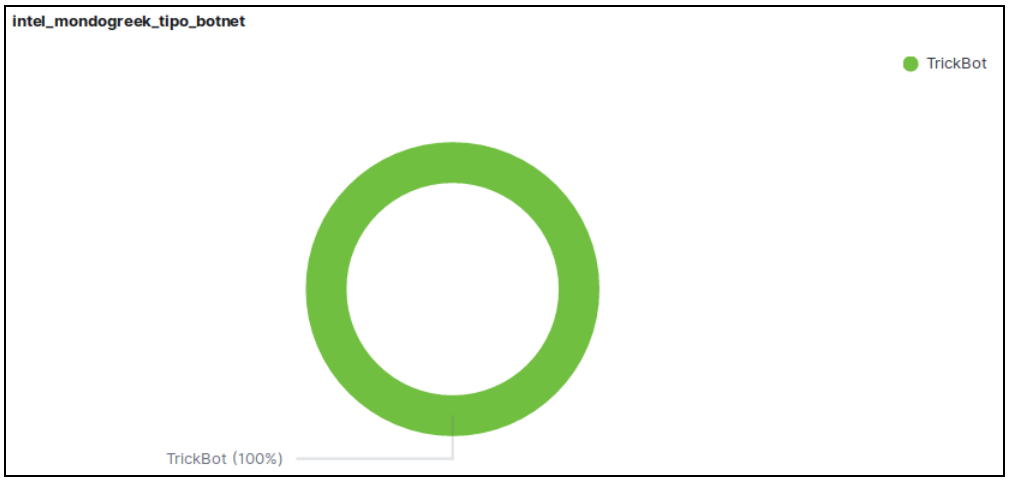

Ilustración 50. Análisis Mondogreek. Intel Log. Tipo de botnet detectada.

Ilustración 51. Análisis Mondogreek. Intel Log. Direcciones IP Origen infectadas.

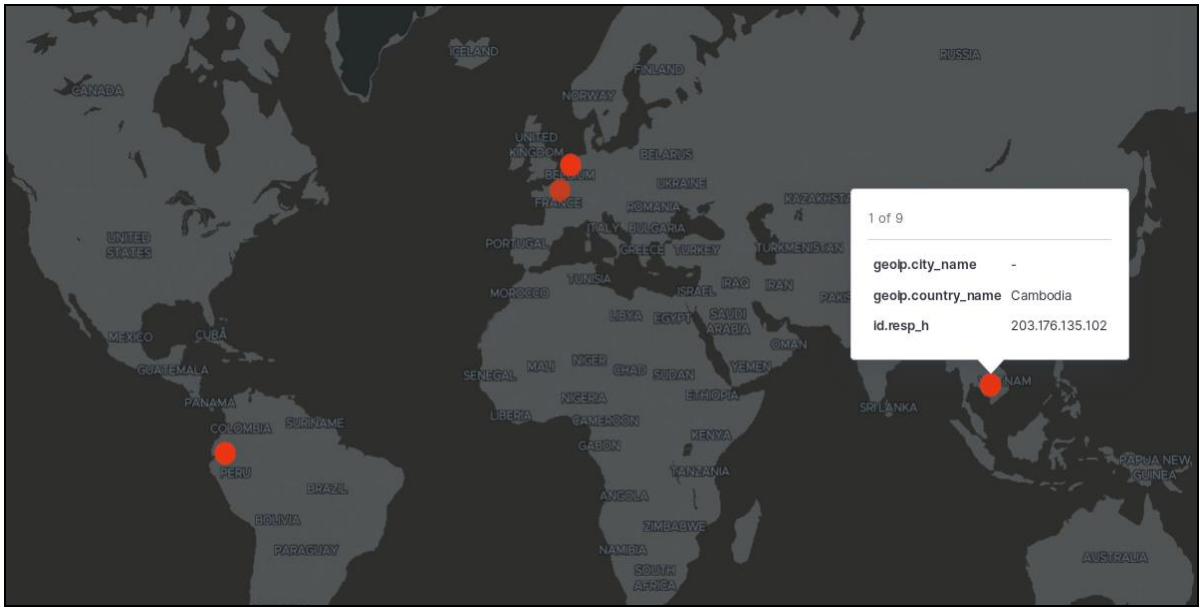

Ilustración 52. Análisis Mondogreek. Intel Log. Geolocalización de botnets.

En estas visualizaciones podemos verificar que el malware proviene de las conexiones realizadas a la **botnet categorizada como Trickbot**.

Por último, se definió una **alerta de tipo Watcher en Kibana** a través de la cual se notificará mediante correo electrónico de las detecciones de botnet realizadas en el análisis de logs proporcionados por Zeek IDS. La alerta genera el siguiente correo electrónico.

Ilustración 53. Análisis Mondogreek. Notificación email.

9.3 Análisis del paquete de red “Spraline”.

El paquete de trafico de red a analizar se llama “**2019-08-20-traffic-analysis-exercise.pcap**”. Procedemos a ejecutar Zeek IDS mediante el comando:

```
zeek -r /home/kali/Downloads/2019-08-20-traffic-analysis-exercise.pcap /usr/local/zeek/share/zeek/site/local.zeek
```

Ilustración 54. Ejecución Zeek IDS análisis Spraline.

Tras ejecutar Zeek IDS, verificamos en el buscador de datos de Kibana, como los datos han sido indexados y almacenados correctamente en el índice “**logstash-notice-log**” y “**logstash-intel-log**” tras ser estos datos enviados por Logstash.

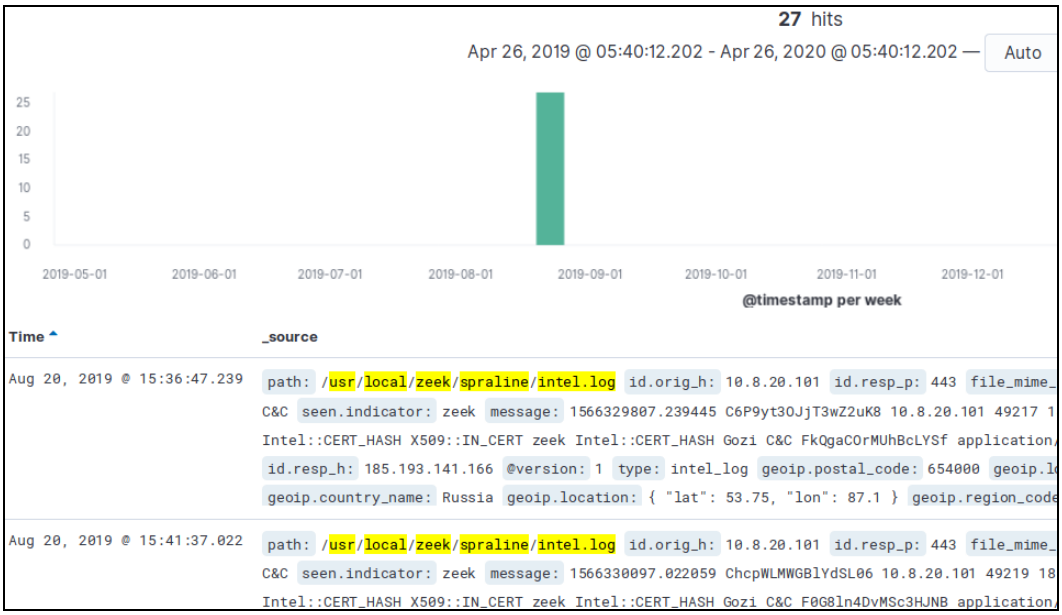

Ilustración 55. Logs Zeek IDS visibles en Elasticsearch y Kibana. Análisis Spraline.

Sobre el dashboard **[Notice] Spraline** obtenemos los siguientes resultados:

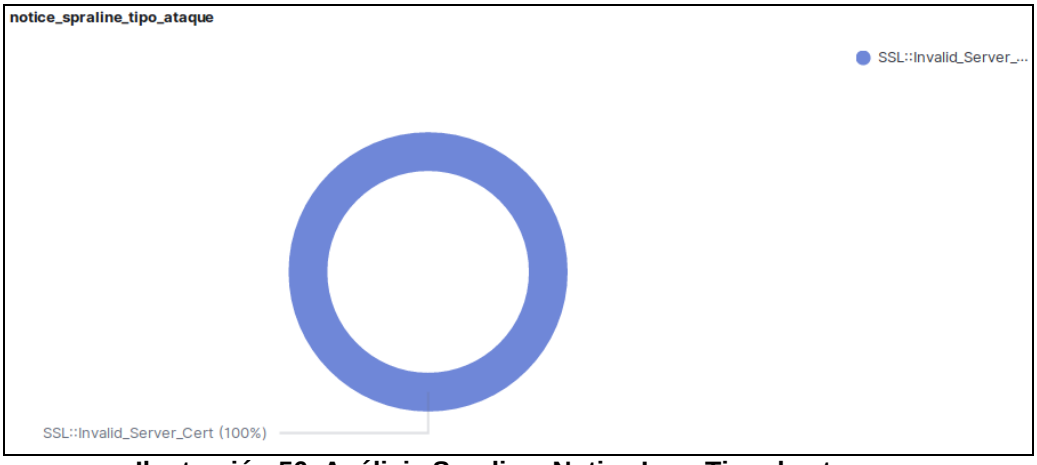

Ilustración 56. Análisis Spraline. Notice Log. Tipo de ataque.

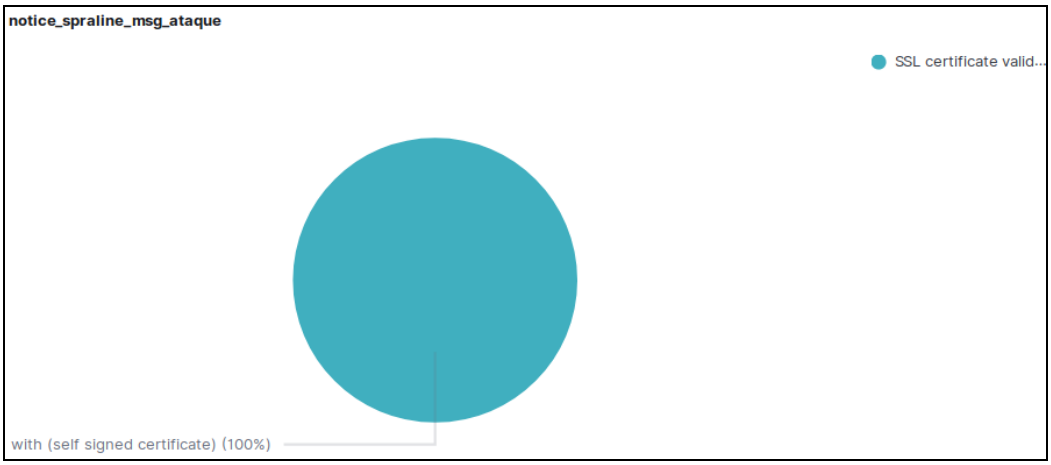

Ilustración 57. Análisis Spraline. Notice Log. Información del ataque.

Ilustración 58. Análisis Spraline. Notice Log. Mapa de conexiones detectadas.

notice_spraline_src_ip

Source IP	Count
10.8.20.101	4
	4

Ilustración 59. Análisis Spraline. Notice Log. Direcciones IP Origen.

notice_spraline_src_port

Source Port	Count
49217	1
49222	1
49224	1
49231	1
	4

Ilustración 60. Análisis Spraline. Notice Log. Puertos Origen.

notice_spraline_dst_ip

Destination IP	Count
185.117.75.41	1
185.193.141.166	1
191.37.181.152	1
89.105.203.184	1
	4

Ilustración 61. Análisis Spraline. Notice Log. Direcciones IP Destino.

notice_spraline_dst_port

Destination Port	Count
443	2
447	1
449	1
	4

Ilustración 62. Análisis Spraline. Notice Log. Puertos Destino.

En base a los análisis de los resultados obtenidos por estos dashboard, podemos intuir, que en esta captura de tráfico el principal ataque de seguridad detectado trata de una **infección por malware basada en la visita a sitios web con certificados SSL inválidos y con firmas propias**, los cuales suelen ser métodos empleados por botnet conocidas.

Además, podemos comprobar las principales direcciones IP origen y destino que intervienen en las comunicaciones, así como la geolocalización de estas.

En el dashboard **[Intel] Spraline**, procedemos a analizar los resultados del log intel para este paquete de trafico de red, donde encontramos
los siguientes dashboard configurados:

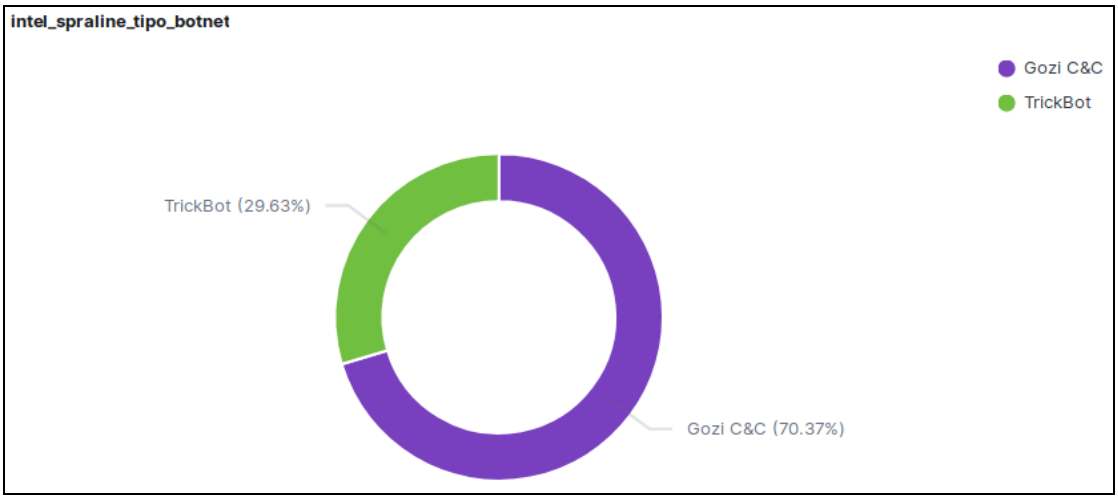

Ilustración 63. Análisis Spraline. Intel Log. Tipo de botnet detectada.

intel_spraline_src_ip

Source IP	C&C connections
10.8.20.101	27
	27

Ilustración 64. Análisis Spraline. Intel Log. Direcciones IP Origen infectadas.

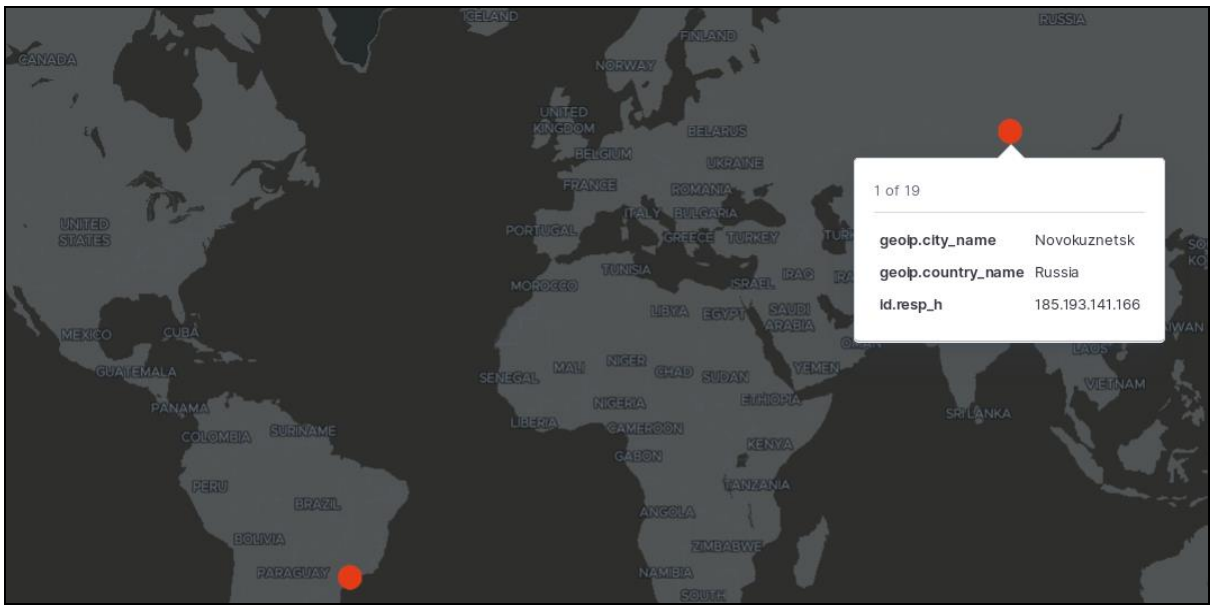

Ilustración 65. Análisis Spraline. Intel Log. Geolocalización de botnets.

En los logs de inteligencia generados por Zeek IDS se verifica las conexiones existentes a las **botnet Gozi C&C (Urnsif) y Trickbot**. Dadas estas conexiones, se produce la infección por malware en el equipo local detectado.

Para finalizar este análisis, se define una **alerta de tipo Watcher en Kibana** a través de la cual se notificará mediante correo electrónico de las detecciones de botnet realizadas en el análisis de logs proporcionados por Zeek IDS. La alerta generada es la siguiente:

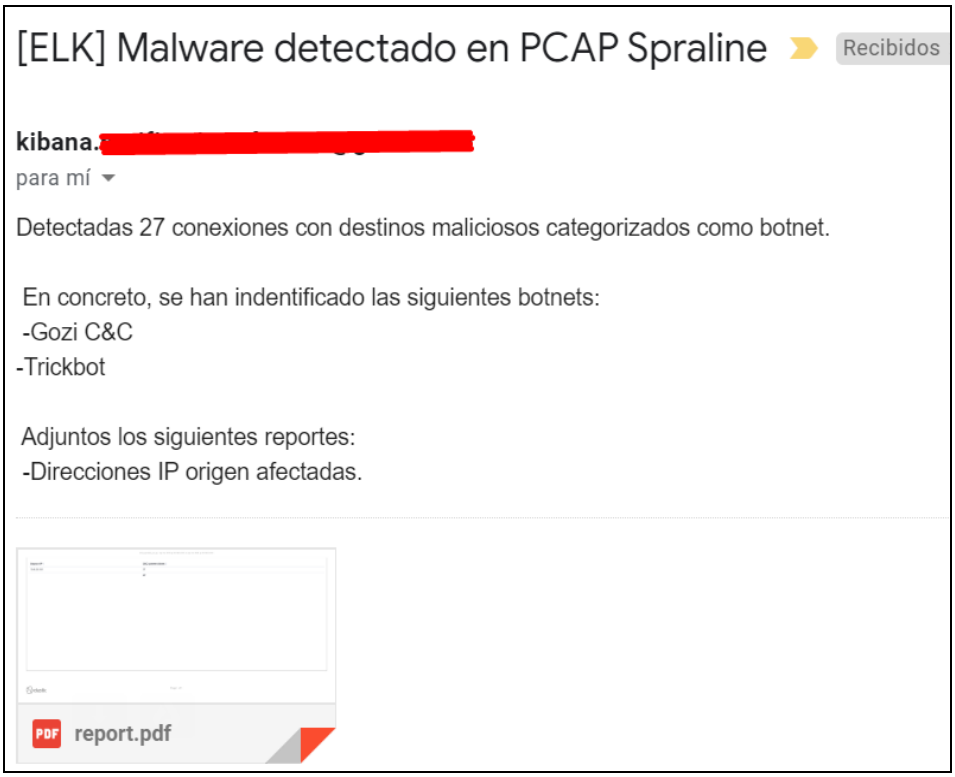

Ilustración 66. Análisis Spraline. Notificación email.

10. Conclusiones.

El desarrollo de este proyecto ha sido una experiencia totalmente positiva. Se han tratado de unos meses de mucho trabajo y esfuerzo, que han sido recompensados con un nivel de aprendizaje muy alto en las competencias previstas del mundo de la seguridad de la información. A continuación se describen las principales conclusiones obtenidas tras la elaboración de este trabajo final de master.

- En la puesta en práctica del producto implementado, se ha podido verificar la utilidad y necesidad que presentan los IDS en el entorno actual de la seguridad informática. Además, se muestra lo útil que es tener una solución como ELK, para disponer de los datos a analizar disponibles de forma centralizada y analizarlos de forma sencilla.

- Se ha podido conocer en detalle el funcionamiento de Zeek IDS, el cual trata de un producto muy potente, aportando una visibilidad sobre anomalías en el tráfico de red a un nivel que otras herramientas IDS no pueden alcanzar. Zeek IDS es una solución muy personalizable, en las cuales puedes crear todo tipo de reglas que se adapten mejor al entorno que se pretende monitorizar. Además, permite su utilización como herramienta de monitorización en tiempo real, o como herramienta de análisis forense.

- El entorno ELK es una solución que permite el envío, almacenamiento y visualización de datos de manera muy completa. Tras trabajar con esta solución, es comprensible que muchas empresas lo utilicen, debido a su bajo coste y el gran rendimiento ofrecido. Además, las visualizaciones y dashboard obtenidos en Kibana permiten realizar un análisis de datos rápido y certero, que puede ser de gran ayuda a los analistas de seguridad en la actualidad.

- El proyecto BZAR de Mitre aporta un grado más de detalle en las detecciones realizadas por Zeek IDS, categorizando los tipos de ataques detectados, ofreciendo una mayor visibilidad y detalle de las anomalías detectadas.

- Por último, este proyecto es fácilmente reproducible en pequeñas y medianas empresas, que podrían aportar una capa más de seguridad a sus organizaciones con soluciones de seguridad de este estilo.

10.1 Evaluación de objetivos cumplidos.

Para completar este apartado de conclusiones, se va a realizar una comparativa con los objetivos planteados al comienzo del proyecto, para verificar a la finalización del proyecto, como todos estos objetivos han sido cumplidos de la forma prevista.

- **Definir un plan de trabajo que permita llevar a cabo el proyecto a tiempo y consiguiendo los objetivos definidos. Seguir una metodología de trabajo que se adapte al plan de trabajo definido.** Se ha seguido el plan de trabajo definido, cumpliendo tareas y tiempos establecidos para cada uno de ellos. La metodología de trabajo seguida de investigación y aplicación posterior de los conocimientos adquiridos me ha permitido conseguir un gran nivel de aprendizaje en las herramientas con las que he trabajado en este proyecto.

- **Estudiar el funcionamiento e integración de Zeek IDS en un entorno de pruebas.** El aprendizaje de Zeek IDS ha sido muy amplio, conociendo todas las posibilidades que ofrece, incluso en el proyecto se han realizado implementaciones personalizadas para ampliar su funcionalidad en el entorno de pruebas preparado. Gracias a esto, también se ha conseguido el subobjetivo de **comprender el funcionamiento de un sistema de detección de intrusiones**.

- **Conocer e investigar los patrones de eventos sospechosos de seguridad que se suelen dar en la red**. Se han analizado y estudiado las principales anomalías de red de seguridad en la actualidad, las cuales se han pretendido analizar en Zeek IDS.

- **Construir un entorno de trabajo adecuado**. El entorno de trabajo construido ha permitido la integración de las herramientas prácticamente sin problemas y de la forma prevista, permitiendo además la exportación y aplicación del producto en otros entornos.

- **Crear reglas de detección de eventos sospechosos de seguridad en Zeek IDS**. Debido al potencial de Zeek, este objetivo venia prácticamente integrada en el producto de forma predeterminada, a pesar de las integraciones personalizadas realizadas.

- **Estudiar el funcionamiento e integración de ELK en un entorno de pruebas**. **Crear graficas que permitan analizar anomalías de seguridad en el tráfico de red.** ELK es un producto muy completo, fácil y sencillo de implementar. Mediante este, hemos podido analizar las detecciones de seguridad de forma muy completa.

- **Investigar sobre fuentes de datos fiables para conocer los indicadores de compromiso (IOCs) sobre malware/botnet que existen en la actualidad.** Hay multitud de fuentes que aporten información útil sobre IOCs y firmas maliciosas actualizadas, que son muy fácil de integrar en el entorno de cada organización.

- **Crear un sistema de alertado en ELK mediante correlación con un feed de reputaciones de elementos maliciosos.** Mediante el módulo watcher de ELK se permite un sistema de alarmado muy completo, capaz de integrar con una gran variedad de buzones de correo electrónico.

- **Realizar una documentación sobre la solución implementada**. Mediante este documento, se ha tratado de explicar con el máximo detalle posible todos los pasos seguidos a lo largo de este proyecto para llegar a la solución final planteada al principio del proyecto.

Por tanto, debido a la consecución de forma paulatina de cada uno de estos subjetivos planteados al inicio de este proyecto, ha permitido alcanzar el objetivo final de implementar una solución mediante Zeek IDS y ELK en el cual se analice de forma completa las anomalías de seguridad en el tráfico de red.

10.2 Problemas encontrados durante el desarrollo del proyecto.

Algunos de los problemas detectados durante la implementación y desarrollo son los siguientes:

- Debido al extraño formato de los logs de Zeek IDS, fue costoso llegar a la configuración optima de logstash que permitiese el indexado de los logs de forma correcta, respetando todos sus campos y valores, así como el timestamp del log.

- Para añadir más detalle a los logs, se añadió en logstash la configuración para geolocalizar las direcciones IP destino de los logs de Zeek IDS. Lograr el funcionamiento de esta parte fue algo costoso y requirió más tiempo que a otras partes del proyecto.

- La configuración del alarmado mediante Watcher, requería de una configuración específica, que impedía al principio la recepción de correos electrónicos de forma adecuada.

10.3 Trabajos futuros.

Por último, algunos de los trabajos futuros que se pueden plantear, para hacer de esta solución una más completa y adaptable en un futuro, son:

- En Zeek IDS, se podrían elaborar más reglas de detección personalizadas, que permitan habilitar otros módulos que contiene Zeek, así como ampliar la información de algunos logs ofrecidos que faciliten el análisis posterior.

- Ampliar el tipo de logs generados por Zeek IDS que sean enviados a ELK, como por ejemplo “pe.log” que permite conocer la estructura de ficheros ejecutables, y mediante el punto anterior, se puede conocer en detalle si el fichero en cuestión se trata de un malware o no. También existen otros logs interesantes como “dns.log” o “http.log”.

- Ampliar el tipo de visualizaciones y dashboard en Kibana que ofrezcan una mayor capacidad de análisis.

- Integración de ELK con una herramienta de ticketing, para permitir su adaptación en la pequeña y mediana empresa, donde cada ticket, corresponda con una de las alertas generadas que requieran de un análisis posterior.

11. Glosario

- IDS: Sistema de detección de intrusiones.
- NIDS: Sistema de detección de intrusiones de tipo red.
- HIDS: Sistema de detección de intrusiones de tipo host.
- IPS: Sistema de prevención de intrusiones.
- SIEM: Security Information and Event Monitoring.
- ELK: Entorno de Elasticsearch, Logstash y Kibana.
- Botnet: Nombre genérico que denomina a cualquier grupo de PC infectados y controlados por un atacante de forma remota
- Kali: Distribución basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática en general
- APT: Una amenaza persistente avanzada es un conjunto de procesos informáticos sigilosos orquestados por un tercero con la intención y la capacidad de atacar de forma avanzada y continuada en el tiempo, un objetivo determinado.
- Mitre: Plataforma que organiza y categoriza los distintos tipos de ataques, amenazas y procedimientos realizados por los distintos atacantes en el mundo digital y que permite identificar vulnerabilidades en los sistemas informáticos
- PCAP: El pcap es una interfaz de una aplicación de programación para captura de paquetes

12. Bibliografía

[1] ***Maldonado, L.*** (2019). ***Blog Everis***. Obtenido de https://blog.everis.com/es/blog/negocio/ciberataques-y-ciberseguridad-evoluci%C3%B3n-en-paralelo

[2] ***Iris-cert.*** (2008). ***Rediris*** Obtenido dehttps://www.rediris.es/cert/doc/unixsec/node26.html

[3] ***Martinez, J. L.*** (2018). ***campusvirtual.uclm.es (Asignatura Seguridad de la Información - Introduccion a la Seguridad).*** Obtenido de https://campusvirtual.uclm.es/pluginfile.php/1852228/mod_resource/content/1/1.2%20-%20Seguridad%20-%20Introduccio%CC%81n%20a%20la%20Seguridad.pdf.

[4] ***Wikipedia - Sistema de deteccion de intrusos*** (2019). Obtenido de https://es.wikipedia.org/wiki/Sistema_de_detecci%C3%B3n_de_intrusos

[5] ***Micro Focus Enterprise Security Team. (***2017) ***Security Information and Event Monitoring (SIEM)***. Obtenido de https://learn.techbeacon.com/tracks/siem

[6] ***Cantón, D. Detección de botnets mediante análisis de paquetes.*** (2014) ***Incibe-cert***. Obtenido de https://www.incibe-cert.es/blog/botnets-analisis-paquetes

[7] ***Documentación oficial Zeek IDS***. (2019). ***Docs-Zeek***. Obtenido de https://docs.zeek.org/en/current/intro/index.html

[8] ***Sanz, J.*** (2015). ***BRO IDS: "el ojo que todo lo ve…".*** Obtenido de https://www.securityartwork.es/2015/07/28/bro-ids-el-ojo-que-todo-lo-ve/

[9] ***Losada, S.*** (2018). ***OpenWebinars. ¿QUÉ ES ELK? ElasticSearch, Logstash y Kibana***. Obtenido de https://openwebinars.net/blog/que-es-elk-elasticsearch-logstash-y-kibana/

[10] ***Wikipedia - Python.*** (2017). Obtenido de https://es.wikipedia.org/wiki/Python.

[11] ***Montoya, D. E.*** (2014). ***Python para Pentesters. 0xWORD***.

[12] ***Wikipedia - VirtualBox***. (2017). Obtenido de https://es.wikipedia.org/wiki/VirtualBox.

[13] ***Ortego, D***. (2017). ***OpenWebinars. Las 8 mejores herramientas open source de detección de intrusión***. Obtenido de https://openwebinars.net/blog/las-8-mejores-herramientas-open-source-de-deteccion-de-intrusion/

[14] ***Alejandro*** (2017). ***protegermipc. Mejores IDS Opensource para Detección de Intrusiones***. Obtenido de https://protegermipc.net/2017/02/22/mejores-ids-opensource-deteccion-de-intrusiones/

[15] ***Robb, D***. (2018) ***esecurityplanet. Top SIEM Products***. Obtenido de https://www.esecurityplanet.com/products/top-siem-products.html

[16] ***Scarfone, K.*** (2018). ***searchsecurity. Seven criteria for evaluating today's leading SIEM tools***. Obtenido de https://searchsecurity.techtarget.com/feature/Comparing-the-best-SIEM-systems-on-the-market

[17] ***Documentación oficial Zeek IDS***. (2019). ***Install guide***. Obtenido de https://docs.zeek.org/en/current/install/install.html

[18] ***Ali Alwashali***. (2019). ***Youtube. Complete guide to install and configure Zeek IDS (formerly Bro IDS)***. Obtenido de https://www.youtube.com/watch?v=T89n561_PmU

[19] ***Documentación oficial Elastic***. (2019). ***Install guide***. Obtenido de https://www.elastic.co/guide/en/elastic-stack-get-started/7.6/get-started-elastic-stack.html#_make_sure_elasticsearch_is_up_and_running

[20] ***Fundamentos seguridad redes*** (2019). ***Monografías***. Obtenido de https://www.monografias.com/docs115/fundamentos-seguridad-redes/fundamentos-seguridad-redes.shtml

[21] ***Mitre Attack. Bzar Project.*** (2020). ***Github***. Obtenido de https://github.com/mitre-attack/bzar

13. Anexo. Archivos de configuración.

/etc/logstash/conf.d/notice_log.conf

```
input {
  file {
    type => "notice_log"
        start_position => "beginning"
        path => "/usr/local/zeek/smb_lateral_movement/notice.log"
  }

  file {
        type => "notice_log"
        start_position => "beginning"
        path => "/usr/local/zeek/mondogreek/notice.log"
  }

  file {
        type => "notice_log"
        start_position => "beginning"
        path => "/usr/local/zeek/spraline/notice.log"
  }

}

filter {

  if [message] =~ /^#/ {
    drop { }
  }

  else{
    if [type] == "notice_log" {
            grok {
                match => [ "message", "(?<ts>(.*?))\t(?<uid>(.*?))\t(?<id.orig_h>(
.*?))\t(?<id.orig_p>(.*?))\t(?<id.resp_h>(.*?))\t(?<id.resp_p>(.*?))\t(?<fuid>(.*?
))\t(?<file_mime_type>(.*?))\t(?<file_desc>(.*?))\t(?<proto>(.*?))\t(?<note>(.*?))
\t(?<msg>(.*?))\t(?<sub>(.*?))\t(?<src>(.*?))\t(?<dst>(.*?))\t(?<p>(.*?))\t(?<n>(.
*?))\t(?<peer_descr>(.*?))\t(?<actions>(.*?))\t(?<suppress_for>(.*?))\t(?<dropped>
(.*?))" ]
            }
    }
 }

  geoip {
        source => "id.resp_h"
        target => "geoip"
  }

  date {
    match => [ "ts", "UNIX" ]
  }
}

output {
  elasticsearch {
    hosts => ["192.168.100.145"]
    index => "logstash-notice-log"
  }
}
```

/etc/logstash/conf.d/intel_log.conf

```
input {
  file {
    type => "intel_log"
        start_position => "beginning"
        path => "/usr/local/zeek/smb_lateral_movement/intel.log"
  }

  file {
        type => "intel_log"
        start_position => "beginning"
        path => "/usr/local/zeek/mondogreek/intel.log"
  }

  file {
        type => "intel_log"
        start_position => "beginning"
        path => "/usr/local/zeek/spraline/intel.log"
  }

}

filter {

  if [message] =~ /^#/ {
    drop { }
  }

  else{
    if [type] == "intel_log" {
             grok {
                 match => [ "message", "(?<ts>(.*?))\t%{DATA:uid}\t(?<id.orig_h>(.*
?))\t(?<id.orig_p>(.*?))\t(?<id.resp_h>(.*?))\t(?<id.resp_p>(.*?))\t%{DATA:fuid}\t
%{DATA:file_mime_type}\t%{DATA:file_desc}\t(?<seen.indicator>(.*?))\t(?<seen.indic
ator_type>(.*?))\t(?<seen.where>(.*?))\t%{NOTSPACE:sources}" ]
             }
    }
  }

  geoip {
    source => "id.resp_h"
    target => "geoip"
  }
  date {
    match => [ "ts", "UNIX" ]
  }
}


output {
  elasticsearch {
    hosts => ["192.168.100.145"]
    index => "logstash-intel-log"
  }
}
```

/etc/elasticsearch/elasticsearch.yml

```
# ======================== Elasticsearch Configuration =========================
# ---------------------------------- Cluster -----------------------------------
# Use a descriptive name for your cluster:
#cluster.name: my-application
# ------------------------------------ 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
# ---------------------------------- Network ----------------------------------
# Set the bind address to a specific IP (IPv4 or IPv6):
network.host: 0.0.0.0
# Set a custom port for HTTP:
http.port: 9200
# --------------------------------- Discovery ---------------------------------
# Pass an initial list of hosts to perform discovery when this node is started:
discovery.seed_hosts: ["127.0.0.1", "[::1]", "192.168.100.147"]
# ---------------------------------- Gateway ----------------------------------
# Block initial recovery after a full cluster restart until N nodes are started:
#gateway.recover_after_nodes: 3
# ---------------------------------- Various ----------------------------------
# Require explicit names when deleting indices:
#action.destructive_requires_name: true
# Email alerts configuration:
xpack.notification.email.account:
    gmail_account:
        profile: gmail
        smtp:
            auth: true
            starttls.enable: true
            host: smtp.gmail.com
            port: 587
            user: CORREO ELECTRONICO KIBANA@gmail.com
```

/etc/kibana/kibana.yml

```
# 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.
server.host: "localhost"

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

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

# When this setting's value is true Kibana uses the hostname specified in the serv
er.host
# setting. When the value of this setting is false, Kibana uses the hostname of th
e host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "home"
#elasticsearch.username: "kibana"
#elasticsearch.password: "pass"

# Enables SSL and paths to the PEM-
format SSL certificate and SSL key files, respectively.
#server.ssl.enabled: false
#server.ssl.certificate: /path/to/your/server.crt
#server.ssl.key: /path/to/your/server.key

# Optional settings that provide the paths to the PEM-
format SSL certificate and key files.
#elasticsearch.ssl.certificate: /path/to/your/client.crt
#elasticsearch.ssl.key: /path/to/your/client.key

# Optional setting that enables you to specify a path to the PEM file for the cert
ificate
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

# Logs queries sent to Elasticsearch. Requires logging.verbose set to true.
#elasticsearch.logQueries: false

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

# Enables you specify a file where Kibana stores log output.
#logging.dest: stdout

# Set the value of this setting to true to suppress all logging output.
#logging.silent: false

# Set the value of this setting to true to suppress all logging output other than
error messages.
#logging.quiet: false

# Set the value of this setting to true to log all events, including system usage
information
#logging.verbose: false

# Set the interval in milliseconds to sample system and process performance metric
s
#ops.interval: 5000
```

Configuración alerta: Movimiento lateral en protocolo SMB:

```
{
  "trigger": {
    "schedule": {
      "interval": "30m"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "logstash-intel-log"
        ],
        "rest_total_hits_as_int": true,
        "body": {
          "query": {
            "terms": {
              "path.keyword": [
                "/usr/local/zeek/smb_lateral_movement/intel.log"
              ]
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "standard",
        "attachments": {
          "report.pdf": {
            "reporting": {
              "url": "http://localhost:5601/api/reporting/generate/printablePdf?jo
bParams=(browserTimezone:America%2FNew_York,layout:(dimensions:(height:554.5,width
:1498),id:preserve_layout),objectType:visualization,relativeUrls:!(%27%2Fapp%2Fkib
ana%23%2Fvisualize%2Fedit%2F94335fd0-7cae-11ea-ac0c-45709cbc5f17%3F_g%3D)))”
            }
          }
        },
        "to": [
          "CORREO PERSONAL@yyy.com"
        ],
        "subject": "[ELK] Movimiento lateral en protocolo SMB",
        "body": {
          "text": "Detectadas {{ctx.payload.hits.total}} conexiones con destinos m
aliciosos categorizados como botnet. \n\n En concreto, se han indentificado las si
guientes botnets: \n -Trickbot \n -
Trickbot C&C \n\n Adjuntos los siguientes reportes: \n -
Direcciones IP origen afectadas."
        }
      }
    }
  }
}
```

Configuración alerta: Malware detectado en PCAP Mondogreek.

```
{
  "trigger": {
    "schedule": {
      "interval": "30m"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "logstash-intel-log"
        ],
        "rest_total_hits_as_int": true,
        "body": {
          "query": {
            "terms": {
              "path.keyword": [
                "/usr/local/zeek/mondogreek/intel.log"
              ]
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "standard",
        "attachments": {
          "report.pdf": {
            "reporting": {
              "url": "http://localhost:5601/api/reporting/generate/printablePdf?jo
bParams=(browserTimezone:America%2FNew_York,layout:(dimensions:(height:554.5,width
:1498),id:preserve_layout),objectType:visualization,relativeUrls:!(%27%2Fapp%2Fkib
ana%23%2Fvisualize%2Fedit%2F9aecd060-7cd9-11ea-ac0c-45709cbc5f17%3F_g%3D)))"
            }
          }
        },
        "to": [
          "CORREO PERSONAL@yyy.com"
        ],
        "subject": "[ELK] Malware detectado en PCAP Mondogreek",
        "body": {
          "text": "Detectadas {{ctx.payload.hits.total}} conexiones con destinos m
aliciosos categorizados como botnet. \n\n En concreto, se han indentificado las si
guientes botnets: \n -Trickbot \n\n Adjuntos los siguientes reportes: \n -
Direcciones IP origen afectadas."
        }
      }
    }
  }
}
```

Configuración alerta: Malware detectado en PCAP Spraline.

```
{
  "trigger": {
    "schedule": {
      "interval": "1m"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "logstash-intel-log"
        ],
        "rest_total_hits_as_int": true,
        "body": {
          "query": {
            "terms": {
              "path.keyword": [
                "/usr/local/zeek/spraline/intel.log"
              ]
            }
          }
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
    "send_email": {
      "email": {
        "profile": "standard",
        "attachments": {
          "report.pdf": {
            "reporting": {
              "url": "http://localhost:5601/api/reporting/generate/printablePdf?jo
bParams=(browserTimezone:America%2FNew_York,layout:(dimensions:(height:554.5,width
:1498),id:preserve_layout),objectType:visualization,relativeUrls:!(%27%2Fapp%2Fkib
ana%23%2Fvisualize%2Fedit%2F48a586f0-7caf-11ea-ac0c-45709cbc5f17%3F_g%3D)))"
            }
          }
        },
        "to": [
          "CORREO PERSONAL@yyy.com"
        ],
        "subject": "[ELK] Malware detectado en PCAP Spraline",
        "body": {
          "text": "Detectadas {{ctx.payload.hits.total}} conexiones con destinos m
aliciosos categorizados como botnet. \n\n En concreto, se han indentificado las si
guientes botnets: \n -Gozi C&C \n-
Trickbot \n\n Adjuntos los siguientes reportes: \n -
Direcciones IP origen afectadas."
        }
      }
    }
  }
}
```