Universitat Oberta de Catalunya

uoc.edu

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

Alumno: Carlos Mezquida Salva
Plan de Estudios: “Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones”. (MISTIC)
Área del trabajo: Hacking

Nombre Consultor/a: Borja Guaita Perez
Nombre Profesor/a responsable de la asignatura: Victor Garcia Font

Fecha Entrega: 28/12/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Desplegar la herramienta "Bro IDS" y su posterior explotación para el análisis de actividades sospechosas en la red*
Nombre del autor:	*Carlos Mezquida Salva*
Nombre del consultor/a:	*Borja Guaita Perez*
Nombre del PRA:	*Victor Garcia Font*
Fecha de entrega (mm/aaaa):	12/2019
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*TFM-Hacking*
Idioma del trabajo:	*Castellano*
Palabras clave	*Seguridad de red, reputación, análisis de red*

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

Cada vez es más habitual leer noticias sobre ataques de robo de datos, cifrado de ordenadores, etc en organizaciones y empresas de todos los tamaños. Es una realidad que la seguridad es un pilar importante para la estabilidad de una organización o empresa.

Por este motivo es necesario implementar todas las posibles medidas para identificar las amenazas Una de las medidas más interesante a nivel de red son los NIDS. Los NIDS son herramientas de detección de intrusión de red que nos permites analizar la red de la organización. Entre ellas se encuentran las herramientas Snort, Suricada y Zeek (anteriormente conocida como Bro IDS).

Este Trabajo analiza la herramienta Zeek para la detección temprana de amenazas y su posterior exploración mediante un panel de control que nos ofrece el stack de Elastic (Elasticsearch, Logstash y Kibana).

Para detección de amenazas se realiza la consulta de fuentes externas de reputación mediante la herramienta IntelMQ que facilita la lista de ip de reputación a nuestro sistema.

Como mejora para la detección se desarrollan mecanismos de alarma para notificar de posibles eventos de actividad maliciosa en la red.

El método de trabajo seguido consiste la definición de una serie de etapas y

objetivos de investigación, implantación y entrega.

El resultado obtenido del Trabajo es la implantación e integración de las herramientas Zeek y listas de reputación para su explotación posterior mediante un panel de control sencillo.

En cuanto a las conclusiones del Trabajo se puede afirmar que las herramientas NIDS son herramientas muy útiles para la detección temprana de anomalías en la red y que pueden ser clave para identificación de amenazas.

Abstract (in English, 250 words or less):

News about attacks, data theft, computer encryption, ... in organizations and companies is becoming more frequent. Security tasks are an important pillar for the stability of organizations and companies.

For this reason it is necessary to implement all possible security measures to identify threats. One of the most interesting measures at the network level is the NIDS. NIDS are network intrusion detection tools that allow us to analyze the network of our organizations. Some NIDS tools are Snort, Suricada and Zeek tools (formerly known as Bro IDS).

This Project analyzes the Zeek tool for the early detection of threats and their subsequent explotation through a control panel offered by the Elastic stack (Elasticsearch, Logstash and Kibana).

I use IntelMQ as a tool of intelliegence to add a reputation list to the system.

Alarm mechanisms have been developed to notify of possible events of malicious activity in the network as an improvement to the detection

I defined a series of stages, a general objective and the objectives of research, implementation and delivery of documentation.

The result obtained from the Project is the implementation and integration of Zeek tools and reputation lists for later exploitation through a simple control panel.

As for the conclusions of the Project, it can be affirmed that NIDS tools are very useful tools for the early detection of network anomalies and that they can be key to threat identification.

Índice

Lista de figuras

Listado de Imágenes y Figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

El número de usuarios con acceso a internet cada día es mayor. No han pasado muchos años desde el inicio de la primera red de ordenadores, conocida como ARPANET[1] para la conexión entre instituciones académicas y estatales de los Estados Unidos (1983) con carácter académico hasta las redes de nueva generación 5G que ahora conocemos, pasando por sus hermanas menores 4G, 3G, 2G, etc ofreciendo el acceso a una red de información multiservicio.

Con la llegada de internet se ha abierto un mar de posibilidades a todos los niveles. En el ámbito de la educación, donde es posible realizar cursos de formación sin necesidad de desplazamientos innecesarios, en medicina, donde la conexión de dispositivos permite poder emitir diagnósticos mediante la visualización de imágenes con calidad de radiodiagnóstico, en transporte, donde el control logístico es posible gestionarlo de un modo más óptimo tanto en el origen como en el destino o en energía, donde se pueden estimar los consumos eléctricos remotamente y adaptar su generación según la tendencia. Pero no todas estas nuevas oportunidad y ventajas se pueden considerar como positivas, ya que aparecen nuevos retos, en algunos casos transformados en peligros que deben ser tenidos en cuenta, por este motivo es fundamental una concienciación y educación de seguridad de la información.

Entre los posibles peligros que existen actualmente en la red (internet) encontramos sofware malicioso (*malware*), *botnes*, *ramsomware*, fraudes de *phishing*, p*harming, spoofing, spyware, etc.* Estos elementos, acompañados de la baja concienciación en términos de seguridad de la información, son los causantes del incremento de delitos telemáticos.

Una de las ideas base de la seguridad de la información es que la seguridad al 100% no existe y por tanto no existe un herramienta que asegure la protección completa de los activos de una organización.

No existe una herramienta hoy en día que pueda realizar todas las tareas de seguridad que se puedan idear de una manera eficiente y eficaz debido a que los usuarios con malas intenciones siempre se adelantan a las implementaciones de las medidas de seguridad que surgen en el mercado.

Por ello, una buena práctica es tener diferentes herramientas que puedan realizar de manera óptima y coordinada entre ellas gran parte de las tareas de seguridad. Una de los herramientas más interesantes relacionada con la seguridad son los sistemas de detección de intrusión de red. Estas herramientas capturan la información que circula por la red y la almacenan en un repositorio donde es posible analizar los flujos de comunicación de los usuarios para conocer los comportamientos y tendencias. Este tipo de herramientas se desplieguen en todo tipo de entornos para identificar posibles anomalías.

Los Sistemas de Detección de Intrusión de Red (NIDS, *Network Intrusion Detection Systems*), son sistemas que ayudan a identificar los comportamientos de red en las organizaciones. Estos sistemas se pueden

1 ARPANET: https://es.wikipedia.org/wiki/ARPANET

agrupar según el modelo que análisis que utilizan. Ya sea basado en firmas o mediante el análisis del comportamiento de los flujos de comunicaciones.

Los NIDS más conocidos actualmente en el entorno Open Source son Snort[2], Suricata[3] y Bro IDS[4].

Uno de los Sistemas de Detección de Intrusión de Red más conocidos es “Bro IDS” que desde 2018 cambió su nombre por Zeek. Zeek es un framework de análisis de red que facilita la identificación de amenazas. Esta herramienta está integrada en muchas distribuciones de seguridad (sistemas operativos de orientados a seguridad) como una aplicación a tener en cuenta y ser usada en análisis forense además de en situaciones para la detección de anomalías en las redes de comunicaciones. Puede utilizarse tanto a nivel online y offline.

Zeek recolecta la información a nivel de protocolo en la interface de red del host en el que se despliega y almacena la información en ficheros en base al protocolo. De este modo cataloga la información para su posterior análisis. La herramienta configurada de manera óptima es capaz de gestionar un gran volumen de tráfico en tiempo real.

Otra de las funcionalidades importantes de Zeek es la utilización de los eventos y la política de scripts. Éstos ofrecen una serie de funcionalidades con gran potencial para los analistas de seguridad.

Zeek no muestra la información de manera amigable mediante un panel de control, por lo que requiere de otro elemento que pueda aportar esa funcionalidad. Para poder realización una representación sencilla y clara se utiliza el Stack de ELK, es decir, el conjunto de aplicaciones de Elastic comúnmente llamadas Elasticsearch, Logstash y Kibana para ofrecer un extra a Zeek en cuanto a:

- Recolección, transformación y almacenamiento de log (Logstash)
- Indexación y búsqueda de información (Elasticsearch)
- Visor de información (Kibana)

Mediante la implementación de Zeek como sistema de identificación y análisis del tráfico de red sin interferir en el normal funcionamiento de la red y ubicada de manera estratégica en la infraestructura de red, se obtiene un sistema que observa los flujos de comunicaciones permitiendo conocer la actividad de red de una organización.

Con la combinación de ambos sistemas (Zeek y ELK) es posible conocer si el tráfico que circula por la red es adecuado o es necesario revisar algún equipo informático o dispositivo electrónico que haya podido ser comprometido y se haya identificado algún proceso de exfiltración de información o la conexión con algún recurso externo a la organización como podría ser un servidor de Comando y Control (*C&C*) para realizar un ataque dirigido.

Una vez se dispone de información realista de la actividad de la organización es posible analizar y valorar los diferentes tipos de datos que circular por la red y tratar de identificar y separar el tráfico malicioso del tráfico habitual.

2 Snort: https://www.snort.org/
3 Suricata: https://suricata-ids.org/
4 Bro IDS: https://www.zeek.org/

Para poder diferencial el tipo de tráfico es necesario compararlo con tráfico sospechoso. Para ello, es necesario consultar fuentes externas para determinar si el tráfico que circula en la red de la organización se puede considerar malicioso. En internet existen listas de reputación que son obtenidas por proveedores de servicios y entidades sin ánimo de lucro con el objetivo de informar y prevenir de las actividades maliciosas por lo que es posible consultarlas, en algunos casos, de manera abierta.

Mediante la consulta de las listas de reputación y la comparación con los datos de red de la organización se puede generar la automatización necesaria para que se genere un evento cuando se produce una coincidencia y posteriormente realizar las acciones de control preventivo que sean necesarias con el objetivo de mitigar una situación de riesgo.

Por tanto el Trabajo de Fin de Master "*Despliegue de la herramienta "Bro IDS" y su posterior explotación para el análisis de actividades sospechosas en la red*" aborda el despliegue de las herramientas Zeek y ELK para la identificación de amenazas en tiempo real a partir de fuentes de información públicas.

1.2 Objetivos del Trabajo

El objetivo principal del presente trabajo es la integración de las herramientas Zeek y listas de reputación de "*malware*, *botnets*, etc" con ELK para la detección de conexiones sospechosas y generar las alarmas de seguridad para que puedan tomarse posteriormente las medidas oportunas. Para completar el objetivo general se definen una sería de objetivos de investigación, implantación y entrega que se especifican a continuación:

Objetivo de investigación:

- Investigar y analizar la arquitectura y estructura de la herramienta Zeek.
- Investigar y analizar la arquitectura y estructura de las herramientas que componen el *stack*[5] de ELK (Elasticsearch, Logstash y Kibana).
- Analizar los diferentes métodos de integración de la herramienta Zeek y ELK para la transferencia de información.
- Analizar cómo crear paneles de control para mostrar información en la herramienta Kibana.
- Investigar qué son y cómo se obtienen las listas de reputación.
- Analizar los diferentes modos de consulta e integración de las listas de reputación con el stack ELK.
- Investigar los diferentes mecanismos para generar eventos y alarmas dentro de la herramienta ELK.
- Investigar cómo generar alarmas cuando se produzca una coincidencia en la información proporcionada por Zeek y la obtenida de las listas de reputación.

Objetivo de implantación:

- Instalar la herramienta Zeek y sus dependencias en un host virtual.
- Comprobar el correcto funcionamiento de Zeek y la visualización de los logs de tráfico de la herramienta.

5 *Stack*: Se nombra "*Stack*" (pila) al conjunto de las herramientas que componen ELK (Elasticsearch, Logstash y Kibana).

- Instalar y comprobar el funcionamiento de las herramientas Elasticsearch, Logstash y kibana, así como sus dependencias en un host virtual.
- Instalar las herramientas y dependencias necesarias para la integración de Zeek y ELK.
- Configurar un panel de control para la visualización de información proporcionada por Zeek.
- Integrar las fuentes de información de reputación en ELK.
- Configurar un panel de control para la visualización de información proporcionada por las listas de reputación.
- Configurar los eventos necesarios en ELK o/y Zeek para la notificación cuando se produzcan coincidencias en el comportamiento del tráfico y las listas de reputación.

Objetivo de entrega:

- Cumplir los plazos de entrega acordados en el Plan de Trabajo en tiempo y forma.
- Desarrollar la documentación de entrega.
  - Entrega 1 - PEC 1. Plan de trabajo.
  - Entrega 2 - PEC2. Seguimiento.
  - Entrega 3 - PEC3. Seguimiento.
  - Entrega 4 - PEC4 (Memoria final).
  - Entrega 5 – Presentación en video.
- Entregar el documento de Trabajo de Fin de Master junto con el video de presentación.

1.3 Enfoque y método seguido

El enfoque y la metodología a seguir para cumplir con los objetivos marcados en el Trabajo de Fin de Master vienen definidos por una serie de etapas que se identifican a continuación.

Definición del plan de trabajo

Se trata de una de las fases más importante ya que es la columna vertebral sobre la que se sustenta el Trabajo. En esta fase se define y contextualiza la necesidad de desarrollar los objetivos de una manera sencilla y clara. Se define el objetivo general, así como los objetivos específicos. También se define la metodología utilizada y las diferentes tareas que son necesaria para cumplir con los objetivos marcados. Además se identifican los diferentes riesgos que puedan hacer peligrar el cumplimiento de los objetivos. Al mismo tiempo se realiza una planificación temporal de las tareas y se enumeran los diferentes documentos que son entregados.

Análisis de la herramienta Zeek (Bro IDS)

En esta fase se explica qué es Zeek (anteriormente Bro IDS). Cuáles son sus funciones los pasos seguidos para poder implementar Zeek en un entorno de laboratorio controlado.

Análisis de la herramienta ELK (Elasticsearch, Logstahs, Kibana)

En esta fase se explica que es ELK y que elementos lo componen, así como los detalles que la hacen una de las herramientas utilizadas para la gestión de datos en internet más relevantes.

Métodos de integración de Zeek y ELK

En esta fase se definen los métodos de integración que se pueden utilizar para transferencia de información entre Zeek y el stack de ELK.

Representación de datos de Zeek en ELK

En esta fase se realiza la configuración necesaria para la representación de los datos obtenidos a través de Zeek en un panel de control claro y simple que permita entender el contenido de este de manera ágil.

Análisis de las listas de reputación

Que son las listas de reputación, cuál es su objetivo y como obtenerlas.

Consulta e integración de listas de reputación

En esta fase se analizan los métodos de consulta de las listas de reputación y como se pueden integrar con nuestro sistema de recolección.

Creación de alertas en ELK

En esta fase se analiza cómo crear alarmas en ELK y como se puede cruzar la información del análisis de tráfico proporcionada por Zeek con la información de las listas de reputación obtenidas y consultadas previamente para generar alertas.

Representación conjunta de datos Zeek y listas de reputación en Kibana

En esta fase se pretende crear un panel de control donde se pueda visualizar, tanto la información aportada por Zeek como de información de actividad sospechosa aportada por fuentes externas.

Conclusiones y trabajo futuro

En esta última fase se presentan las conclusiones del desarrollo del Trabajo Final de Master además de las posibles nuevas líneas de trabajo futuro que puedan desarrollarse como consecuencia del mismo.

1.4 Listado de Tareas

En este aparatado se detallan las tareas que son realizadas en cada una de las fases indicadas en el apartado anterior.

Documentación

Se estudia el contexto y justificación del trabajo planteado.

Orientación del trabajo

Se define el objetivo general y objetivos concretos para la realización del trabajo.

Ámbito del trabajo

Se define el ámbito en el que se engloba el trabajo, así como las tareas a realizar.

Plan de trabajo

Contexto y justificación: Se expone la necesidad de las herramientas de análisis de red para la detección de comportamientos y actividades sospechosas.

Objetivo: Se detallan los pasos seguidos para poder disponer de un sistema que pueda ser capaz de identificar accesos sospechosos mediante el despliegue de varias herramientas que puedan ayudar a la identificación en las comunicaciones.

Metodología: Se identifican las fases y la manera en las que se divide el trabajo para conseguir los objetivos definidos.

Listado de tareas: Mediante la lista de tareas se obtiene metas alcanzables.

Planificación: Se presenta la estimación temporal de las tareas definidas, teniendo en cuenta los plazos acordados para las entregas parciales del Trabajo de Fin de Master.

Recursos necesarios: Se identifican los recursos que se utilizan para el desarrollo del trabajo.

Análisis de riesgos: Se definen los factores que pueden hacer peligrar el cumplimiento de los objetivos definidos.

Productos obtenidos: Se detallan los productos obtenidos con la realización del trabajo.

Entrega 1 - PEC 1 - Plan de trabajo: Se trata del punto de control para la revisión de la evolución del trabajo mediante la entrega de documentación.

Análisis de la herramienta Zeek (Bro IDS)

Descripción: Se describe las funcionalidades de la herramienta.

Estructura: Se define la estructura y características de la herramienta para su funcionamiento.

Requisitos técnicos: Se identifican los requisitos para el despliegue de la herramienta.

Despliegue: Se detallan los pasos seguidos para la puesta en marcha de la herramienta en un entorno controlado.

Comprobación: Se enumeran los servicios que son necesario para el funcionamiento adecuado de la herramienta.

Análisis de la herramienta ELK (Elasticsearch, Logstahs, Kibana)

Descripción: Se describe las funcionalidades de las herramientas Elasticsearch, Logstahs y Kibana (*stack* de ELK) tanto a nivel general como individual.

Estructura: Se define la estructura y características de las herramientas para su funcionamiento dentro del *stack* ELK.

Requisitos técnicos: Se identifican los requisitos para el despliegue de las herramientas

Despliegue: Se detallan los pasos seguidos para la puesta en marcha de las aplicaciones Elasticsearch, Logstahs y Kibana en un entorno controlado.

Comprobación: Se enumeran los servicios que son necesario para el funcionamiento adecuado de las herramientas.

Métodos de integración de Zeek y ELK

Formato de logs: Se identifican los formatos que son usados o tratados por las diferentes herramientas.

Análisis: Se analizan los diferentes aspectos relevantes de integración entre herramientas.

Comprobación funcional: Se identifican las configuraciones realizadas para la integración de información.

Descripción: Se define la solución final adoptada para la integración de las herramientas.

Representación de datos de Zeek en ELK

Investigación: Se analizan las diferentes opciones que se dispone en la herramienta Kibana para la representación de datos en paneles de control.

Comprobación: Se genera un panel de control amigable para catalogar la información que se obtiene de Zeek.

Descripción: Se definen los pasos seguidos para poder obtener el panel de control y como personalizarlo.

Análisis de las listas de reputación

Estudio de listas de reputación: Se investiga que se entiende por lista de reputación, donde se pueden consultar, como se gestionan, quien las gestiona y con qué finalidad.

Entrega 2 - PEC 2 - Seguimiento: Se trata del punto de control para la revisión de la evolución del trabajo mediante la entrega de documentación.

Consulta e integración de listas de reputación

Investigación: Se analiza la información existente sobre integraciones con fuentes externas.

Análisis de los mecanismos de consulta: Se definen los modos en los que se puede realizar consultas a fuentes públicas conocidas de información reputacional.

Análisis de los mecanismos de integración: Se explican los diferentes mecanismos disponibles para obtener la información de interés para integrarla con nuestro sistema ELK.

Configuración del modelo de integración: Se define y detalla el modelo de integración elegido.

Creación de alertas en ELK

Definición de alertas: En este apartado que explica que se entiende por una alerta en ELK.

Modelos de alerta en ELK: Se enumeran algunos de los mecanismos para generación de alertas en ELK.

Elección del modelo de alerta: Se define y explica los pasos a realizar para la configuración de los alertas en ELK.

Representación conjunta de datos Zeek y listas de reputación en Kibana

Investigación: Se analizan las diferentes opciones que se dispone en la herramienta Kibana para la representación de datos en paneles de control.

Comprobación: Se genera un panel de control amigable para catalogar la información que se obtiene de Zeek.

Descripción: Se definen los pasos seguidos para poder obtener el panel de control y como personalizarlo.

Entrega 3 - PEC 3 - Seguimiento: Se trata del punto de control para la revisión de la evolución del trabajo mediante la entrega de documentación.

Conclusiones y trabajo futuro

Conclusiones: Se enumeran y detallan las conclusiones que se desprenden al completar los objetivos definidos en el trabajo.

Trabajo futuro: Se identifican posibles trabajos que puedan derivarse del trabajo presentado.

Redacción de la Memoria

Redacción de la memoria: Se realiza la memoria completa donde se engloba toda la información objeto del trabajo.

Entrega 4 - PEC4 (Memoria final): Se trata del punto de control de entrega de la memoria completa del Trabajo de Fin de Master.

Realización del video de presentación

Preparación del video: Se identifican los puntos más relevantes a destacar en la presentación del video mediante una estructura ordenada y clara.

Grabación del video: Se realiza la grabación de la presentación en video dando relevancia a los puntos más importantes del Trabajo de Fin de Master.

Entrega 5 – Presentación en video: Se trata del punto de control para la entrega del video de presentación en el aula virtual.

Defensa del Trabajo de Fin de Master

Se realiza la defensa del Trabajo de Fin de Master de forma virtual en el foro del aula donde se plantean cuestiones por parte del tribunal, las cuales deben ser contestadas en un plazo no superior a 24 horas.

1.5 Planificación del Trabajo

A continuación de muestra la planificación temporal del desarrollo de Trabajo de Fin de Master en diferentes formatos para su comprensión. No se ha tenido en cuentas los periodos vacacionales ni fines de semana, ya que se considera que las tareas definidas en el plan de trabajo se realizan de manera organizada con el tiempo asignado a la actividad laboral y familiar. Aun así, se ha estimado un periodo de reserva de 10 días aproximadamente por si surgiera algún imprevisto que impidiera completar el Trabajo en los plazos indicados.

NOMBRE DE LA TAREA	INICIO	FIN
Trabajo de Fin de Master	18/09/19	17/01/20
Documentación	18/09/19	19/09/19
Orientación del trabajo	20/09/19	20/09/19
Ámbito del trabajo	21/09/19	21/09/19
Plan de trabajo	22/09/19	30/09/19
Contexto y justificación	22/09/19	23/09/19
Objetivo	24/09/19	24/09/19
Metodología	25/09/19	25/09/19
Listado de tareas	26/09/19	26/09/19
Planificación	27/09/19	27/09/19
Recursos necesarios	28/09/19	28/09/19
Análisis de riesgos	29/09/19	29/09/19
Productos obtenidos	30/09/19	30/09/19
Descripción de los capítulos	30/09/19	30/09/19
Entrega 1 - PEC 1 - Plan de trabajo	*1/10/19*	*1/10/19*
Análisis de la herramienta Zeek (Bro IDS)	3/10/19	8/10/19
Descripción	3/10/19	3/10/19
Estructura	4/10/19	4/10/19

NOMBRE DE LA TAREA	INICIO	FIN
Requisitos	5/10/19	5/10/19
Despliegue	6/10/19	7/10/19
Comprobación	8/10/19	8/10/19
Análisis de la herramienta ELK (Elasticsearch, Logstahs, Kibana)	9/10/19	14/10/19
Descripción	9/10/19	9/10/19
Estructura	10/10/19	10/10/19
Requisitos	11/10/19	11/10/19
Despliegue	12/10/19	13/10/19
Comprobación	14/10/19	14/10/19
Métodos de integración de Zeek y ELK	15/10/19	20/10/19
Formato de logs	15/10/19	16/10/19
Comprobación funcional	17/10/19	18/10/19
Descripción	19/10/19	20/10/19
Representación de datos de Zeek en ELK	21/10/19	26/10/19
Investigación	21/10/19	22/10/19
Comprobación	23/10/19	24/10/19
Descripción	25/10/19	26/10/19
Análisis de las listas de reputación	27/10/19	28/10/19
Estudio de listas de reputación	27/10/19	28/10/19
Entrega 2 - PEC2. Seguimiento	*29/10/19*	*29/10/19*
Consulta e integración de listas de reputación	30/10/19	8/11/19
Investigación	30/10/19	1/11/19
Análisis de los mecanismos de consulta	2/11/19	3/11/19
Análisis de los mecanismos de integración	4/11/19	5/11/19
Visualización de información (cuadro de mandos)	6/11/19	8/11/19
Creación de alertas en ELK	9/11/19	17/11/19
Definición de alertas	9/11/19	11/11/19
Modelos de alerta en ELK	12/11/19	14/11/19
Requisitos para el despliegue de alertas	15/11/19	17/11/19
Elección del modelo de alertas	18/11/19	18/11/19
Creación de alerta	19/11/19	20/11/19
Comprobación funcional de la alerta	21/11/19	22/11/19
Representación conjunta de datos Zeek y listas de reputación en Kibana	22/11/19	22/11/19
Descripción	22/11/19	23/11/19
Implementación	24/11/19	25/11/19
Entrega 3 - PEC3. Seguimiento	*26/11/19*	*26/11/19*
Conclusiones	28/11/19	3/12/19
Desarrollo del trabajo	28/11/19	30/11/19
Análisis de riesgos	1/12/19	2/12/19
Identificación de puntos de mejora y trabajos futuros	3/12/19	4/12/19
Glosario	4/12/19	5/12/19
Bibliografía y fuentes consultadas	5/12/19	5/12/19
Bibliografía	6/12/19	6/12/19
Páginas web consultadas	7/12/19	7/12/19
Videos consultados	8/12/19	8/12/19

NOMBRE DE LA TAREA	INICIO	FIN
Redacción de la memoria	9/12/19	20/12/19
Entrega 4 - PEC4 (Memoria final)	*21/12/19*	*2412/19*
Realización del video de presentación	1/01/20	6/01/20
Preparación del video	1/01/20	3/01/20
Grabación del video	4/01/20	6/01/20
Entrega 5 – Presentación en video	*7/01/20*	*7/01/20*
Defensa del Trabajo de Fin de Master	13/01/20	17/01/20

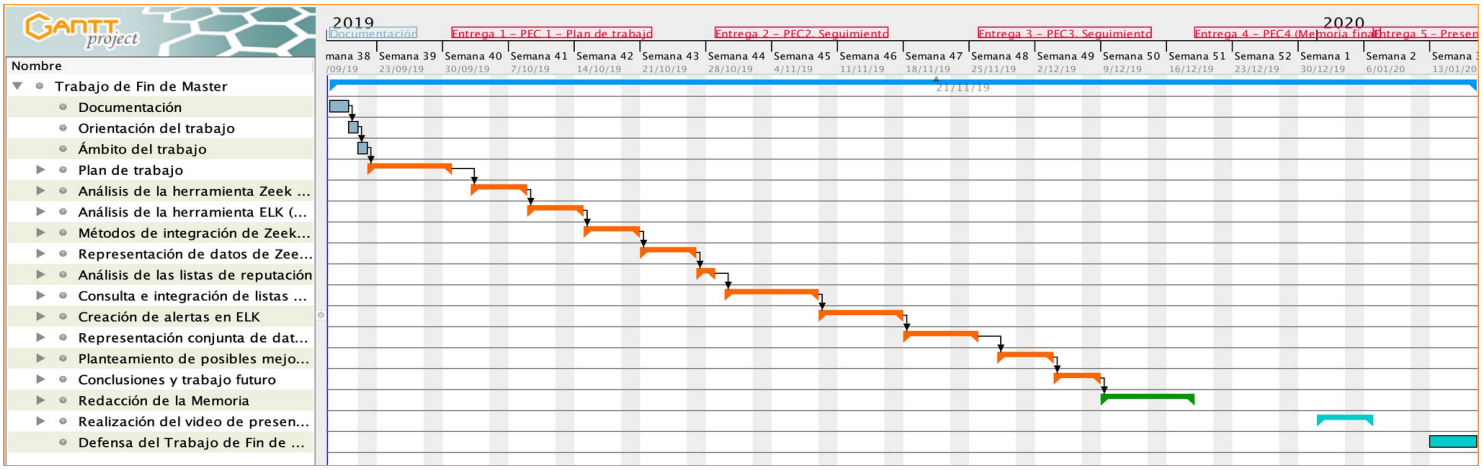

Figura 1: Diagrama de Gantt

1.6 Recursos necesarios para el desarrollo del trabajo

Para el desarrollo del Trabajo de Fin de Master se utiliza un entorno virtual controlado para poder desplegar en una o varias máquinas las herramientas necesarias. En concreto se utiliza el siguiente equipamiento hardware.

- **Equipo para virtualización**: Dell Inc. PowerEdge R710
  - **CPU**: 8 CPUs x Intel(R) Xeon(R) CPU E5620 @ 2.40GHz
  - **Memoria RAM**: 64GB
  - **Tarjetas de red:** 4 gigabit ethernet
- **Switch**: Aruba 2930F JL255A
- **Router**: Firwall Pfsense - 2.4.4-RELEASE-p3 (amd64)
  - Intel(R) Celeron(R) CPU J1800 @ 2.41GHz
  - Current: 2407 MHz, Max: 2408 MHz
  - 2 CPUs: 1 package(s) x 2 core(s)
- **Equipo de trabajo:** MacBook Pro 13” Retina (2014)

El acceso a internet es otro de los recursos necesario para poder realizar la descarga de las herramientas, la consulta de las fuentes de información de reputación (*malware*, *botnets*, etc) y para la búsqueda de información.

1.7 Análisis de riesgos preliminares

En este apartado se realiza un breve análisis de los riesgos que pueden hacer peligrar el seguimiento temporal planteado en el apartado de planificación

generando una desviación de la continuidad del trabajo. Por cada riesgo identificado se realiza una valoración de probabilidad e impacto además de la propuesta de mitigación:

Riesgo 1 – Dimensionamiento temporal

- Definición: Una inadecuada asignación temporal a las tareas identificadas puede causar desviaciones graves debido al desconocimiento en profundidad de las herramientas.
- Probabilidad / Impacto (1/5): 4 / 5
- Mitigación: Tratar de simplificar el desarrollo de la tareas. Aprovechar el tiempo de investigación al máximo para minimizar los posibles retardos.

Riesgo 2 – Fallos de hardware / software

- Definición: Las herramientas se instalan sobre una plataforma de virtualización, que a su vez albergara los hosts necesarios para implementar las herramientas.
- Probabilidad / Impacto: 3 / 5
- Mitigación: Instalación de un equipo UPS (sai) para evitar interrupciones eléctricas inesperadas. Realizar copias de seguridad externas de manera periódicas.

Riesgo 3 – Implementación de la solución

- Definición: Durante el proceso de implementación de los diferentes apartados pueden surgir incompatibilidades o discrepancias con la idea inicial de implementación que haga variar el rumbo del trabajo.
- Probabilidad / Impacto: 3 / 4
- Mitigación: Establecer un contacto constante con los tutores y profesores asociados para tratar de desbloquear las situaciones de riesgo. Por otro lado, se pueden consultar foros especializados para plantear las situaciones de compromiso.

Riesgo 4 – Implementación de la solución

- Definición: Se puede dar el caso en el que el nivel de detalle de la documentación y desarrollo del Trabajo de Fin de Máster sea demasiado extenso que pueda influir en el contenido y el formato final.
- Probabilidad / Impacto: 2 / 3
- Mitigación: Se propone simplificar el contenido. Orientar el contenido a los objetivos definidos y plasmas mediante anexos los detalles más técnicos si fuera necesario.

1.8 Breve sumario del producto obtenido

Con el desarrollo del trabajo se espera poder obtener un documento que pueda servir de guía para el despliegue:

- Sistema IDS (Zeek)
- Sistema de gestión de información (ELK)

- Implementación de una solución de recolección de fuentes de reputación
- Panel de Control de actividad sospechosa de red

Para ello, será necesario cumplir con las metas identificas en apartados anteriores por lo que se entregan los documento relativos a:

- **Entrega 1 - PEC 1 - Plan de trabajo**: Plan de trabajo con la información general y fundamental del inicio del Trabajo de Fin de Master.
- **Entrega 2 - PEC 2 – Seguimiento**: Documento de seguimiento donde se refleja la información del despliegue de las herramientas Zeek y ELK. Además se definen qué son las listas de reputación.
- **Entrega 3 - PEC 3 - Seguimiento**: Documento con los detalles del a puesta en marcha del sistema y la integración de eventos.
- **Entrega 4 - PEC4 (Memoria final)**: Documento completa del Trabajo de Fin de Master con los todos detalles requeridos.
- **Entrega 5 – Presentación en video**: Fichero de video con la presentación y defensa del Trabajo de Fin de Master.

1.9 Breve descripción de los capítulos de la memoria

A continuación, se realiza una breve introducción de los capítulos que posteriores se desarrollan en el Trabajo de Fin de Master.

Introducción

Se realiza la presentación de los objetivos y la motivación del desarrollo de Trabajo de Fin de Master.

Análisis de la herramienta Zeek (Bro IDS)

En este capítulo se facilita la información detalla de herramienta Zeek.

Análisis de la herramienta ELK (Elasticsearch, Logstahs, Kibana)

En este capítulo se facilita la información detalla de las herramientas Elasticsearch, Logstash y Kibana.

Métodos de integración de Zeek y ELK

Este capítulo trata sobre los mecanismos que existen para la comunicación entre las herramientas Zeek y ELK.

Representación de datos de Zeek en ELK

Mediante este capítulo se identifican los pasos a seguir para la creación de panel de panel de control con información de Zeek.

Análisis de las listas de reputación

En este apartado se describe el concepto de lista de reputación y cómo se obtienen.

Consulta e integración de listas de reputación

Complementando el apartado anterior, en este apartado se describen los modos de consulta e integración de las fuentes externas para su utilización en nuestras herramientas.

Creación de alertas en ELK

Este apartado se centra en la generación de alarmas como consecuencia de los eventos creados en base a las necesidades de identificación del tráfico.

Representación conjunta de datos Zeek y listas de reputación en Kibana

Este apartado contempla los pasos seguidos para poder mediante un panel de control información clara pero relevante del comportamiento de la red.

Conclusiones y trabajo futuro

En el último apartado se presentaran las conclusiones del Trabajo de Fin de Master, así como propuestas de trabajo futuros que puedan ser de guía para otros trabajos.

2. Análisis de la herramienta Zeek (Bro IDS)

2.1 Descripción

Zeek es una herramienta open-source para ser utilizada sin restricciones bajo licencia "*BSD license*[6]". Fue desarrollada por Vern Paxson, y posteriormente continuada por Robin Sommer y el equipo de *International Computer Science Institute*[7] y *National Center for Supercomputing Applications*[8].

Zeek, anteriormente conocida como Bro IDS, es una herramienta de análisis, monitorización e inspección de tráfico para la detección de actividad de red sospechosa. La herramienta ofrece una serie de características que permiten a los analistas de seguridad obtener información, ya sea en tiempo real, o fuera de banda (offline) y analizar el comportamiento del tráfico de red. Permite una gran personalización a través de los scripts de la aplicación para adaptarlos a las necesidades de cada entorno y para la detección concreta de anomalías.

Zeek registra la actividad de red gracias a un catálogo de protocolos conocidos para poder agruparlos en archivos de registro.

Permite la integración con herramientas externas para la mejora de funcionalidades.

2.2 Estructura

Zeek se compone por dos elementos principalmente; el motor de eventos (*event engine* o *core*) y el intérprete de scripts (*script interpreter*).

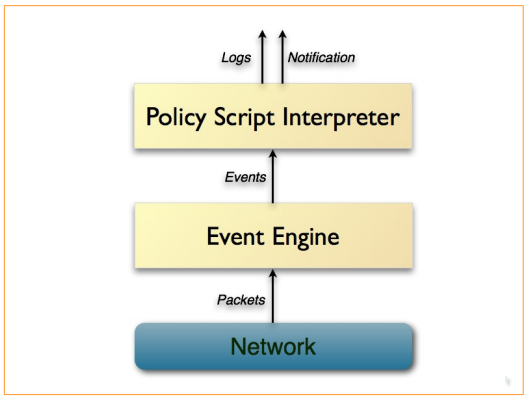

Figura 2: Estructura de Zeek

- Motor de eventos (*event engine*): interpreta los paquetes que recibe y los cataloga a alto nivel en forma de eventos. Estos eventos representan la actividad de la red.
- Interprete de scripts (*Policy Scripts Interpreter*): mediante los "*event handlers*", escritos en lenguaje propio de Zeek se configura la seguridad del entorno donde se despliega. Es decir, se realizan acciones en base a una actividad de red determinada. Estas acciones podrían ser, desde el envío de un correo electrónico, alertas a un sistema de control, etc …

6.- *BSD license*: https://es.wikipedia.org/wiki/Licencia_BSD
7.- *International Computer Science Institute*: *http://www.icsi.berkeley.edu*
8.- *International Computer Science Institute: http://www.ncsa.illinois.edu*

Este módulo tiene una capacidad muy amplia en cuanto a personalización que otros sistemas NIDS no tienen.

Zeek no es una herramienta *multithreaded*, por lo que se debe tener en cuenta para su despliegue, ya que una vez se llega al límite se uso de CPU no hay manera de distribuir la carga si no es mediante un sistema de cluster Zeek.

Se puede desplegar Zeek en modo *standalone* o modo *cluster*. La diferencia es la manera con la que se distribuye la carga y los elementos que aparecen en el modelo de cluster.

Modelo Standalone Zeek; se considera que el mismo host realiza todas las funciones propias de Zeek.

Modelo Cluster Zeek; varias máquinas pueden realizar cada una de las funciones o pueden realizarse de manera combinada. A continuación de describe brevemente cada elemento a nivel informativo.

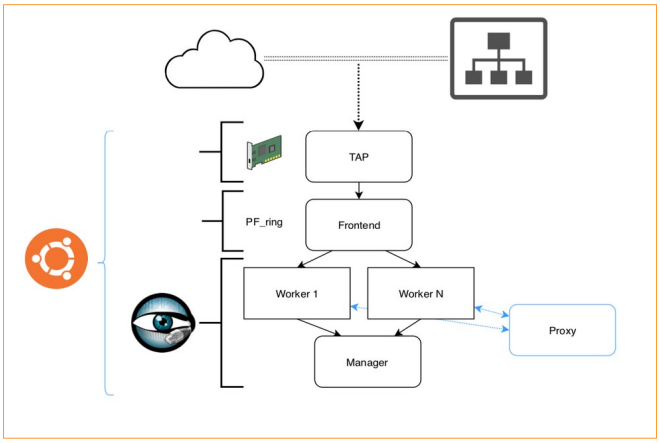

Figura 3: Ejemplo Cluster Zeek

- Tap: se conoce como *Tap*, *Span*, *PortMirroring*, etc … se trata de la manera con la que se recibirá una copia de los paquetes de la red para no interferir en el flujo normal de las comunicaciones. Aunque no todos los modos funcionan de la misma manera, es conveniente analizar cada uno ellos por separado. Es importante identificar el objetivo de análisis para poder ubicar correctamente el host en modo Tap y elegir el mejor modo de implementarlo.
- Frontend: se identifica como la técnica que se usa para capturar los paquetes de información. Zeek no realiza esta función directamente, por lo que en la documentación oficial de Zeek se facilitan algunas técnicas (*cPacket, OpenFlow Switches, PF_RING, Netmap*)[9]
- Worker: es el encargado de *snifar* el tráfico y analizarlo. Se recomienda 1 core por 250Mbps de tráfico. Este modelo de workers permite un fácil crecimiento para la computación.
- Proxy: se delega en este rol las tareas administrativas.

9 Tipo de Frontends : https://docs.zeek.org/en/stable/cluster/index.html

- Manager: es el encargado de recibir los mensajes de logs y las notificaciones de los nodos del cluster. Si el manager no realizase las funciones de loger, sería necesario otro elemento identificado como Logger.

A nivel de aplicación del framework, destacamos *Zeek*, que es la herramienta propiamente dicha y *ZeekControl*, que es la herramienta de administración de Zeek.

En cuanto a la estructura interna de Zeek destacar:

- `/opt/zeek/logs/*` logs
- `/opt/zeek/bin/*` ejecutables (zeek, zeekctl, zeek-cut)
- `/opt/zeek/share/base/*` procesos de inicialización de base de Zeek
- `/opt/zeek/share/policy/*` donde se encuentran los scripts de detección
- `/opt/zeek/share/site/*` donde se almacena la información local de Zeek

2.3 Requisitos

Basándonos en la documentación de Zeek[10-11] y teniendo en cuenta el escenario que ha definido para el desarrollo del trabajo se identifican una serie de requerimientos técnicos como se ha comentado en el apartado ***1.6 Recursos necesarios para el desarrollo del trabajo*** se utiliza un hardware para virtualizar los servidores de aplicaciones necesarios.

En cuanto a los requisitos de la infraestructura software destacamos:

- **Aplicación para virtualización**: VMware versión 6.7.0 Update 2 (Build 13006603) con licencia de demostración de 60 días.
- **Sistema operativo:**
  - Static hostname: vm07pec4zeek
  - Operating System: Ubuntu 19.10 (Eoan Ermine)
  - Kernel: Linux 5.3.0-24-generic
  - Architecture: x86-64

Desplegaremos una máquina virtual Ubuntu 19.10 para poder instalar Zeek con las siguientes características:

- **Sistema Operativo**: Sistema Operativo: Linux 5.3.0-24-generic Ubuntu x86_64 GNU/Linux GNU/Linux
- **Información disco duro**: 60 GB
- **Información RAM**: 8 GB

10 Requisitos de instalación: https://docs.zeek.org/en/stable/install/install.html
11 Installation and Configuration: https://www.zeek.org/documentation/faq.html#installation-and-configuration

2.4 Despliegue

Para el despliegue seguiremos las recomendaciones del manual de instalación de zeek, además de otras fuentes de información externas de usuarios que han realizado despliegues similares.

- **Hostname**: vm07pec4zeek
- **Dirección MAC**: 00:0c:29:47:6d:6c
- **Dirección IP**: 192.168.29.90
- **Núcleos CPU**: 4
- **Memoria RAM**: 8GB

Consideraciones relevantes.

- Configuramos la tarjeta de red del host en modo promiscuo, para capturar el tráfico que pasa por el segmento de red en el que se encuentra la tarjeta de red.

```
# ip link set ens160 promisc on
```

- Instalamos todos los paquetes necesarios para la puesta en marcha de Zeek, Compilación de Zeek y las dependencias correspondientes. Todos estos paquetes han sido necesario, según se indica en la documentación sobre la herramienta y al observar los diversos errores que se desprendían del proceso de instalación.

```
# apt-get install cmake make gcc g++ gdb flex bison libpcap-dev libssl-dev
python-dev swig zlib1g-dev libgeoip-dev build-essential libelf-dev libmagic-
dev libmaxminddb-dev openssl python-pip git iperf libkrb5-dev systemtap
libtool libgdbm-dev libreadline-dev doxygen
```

- Instalamos y configuraremos PF_ring para captura y balanceado de trafico en varios procesos de manera simultánea aprovechando el uso de múltiples cores de un mismo host.

```
/opt/nDPI# cd /opt/
/opt# git clone https://github.com/ntop/PF_RING.git
/opt# cd PF_RING/kernel
/opt/PF_RING/kernel# make
/opt/PF_RING/kernel# insmod ./pf_ring.ko
/opt/PF_RING/kernel# cd ../userland
/opt/PF_RING/userland# make
/opt/PF_RING/userland# cd lib
/opt/PF_RING/userland/lib# ./configure –prefix=/opt/PF_RING
/opt/PF_RING/userland/lib# make install
/opt/PF_RING/userland/lib# cd ../libpcap
/opt/PF_RING/userland/libpcap# ./configure –prefix=/opt/PF_RING/
/opt/PF_RING/userland/libpcap# make install
/opt/PF_RING/userland/libpcap# cd ../tcpdump-*
/opt/PF_RING/userland/tcpdump-4.9.2# ./configure
/opt/PF_RING/userland/tcpdump-4.9.2# make install
/opt/PF_RING/userland/tcpdump-4.9.2# cd ../../kernel
/opt/PF_RING/kernel# make
/opt/PF_RING/kernel# make install
/opt/PF_RING/kernel# echo "pf_ring" >> /etc/modules
```

Verificaremos que se inicia correctamente en el arranque PF_ring si ejecutamos el comando que se indica a continuación y se muestra un mensaje similar al que se indica.

```
/opt/PF_RING/kernel# lsmod | grep pf_ring
pf_ring              1245184  0
```

```
root@serdeb09tmf4:/tmp# ldd /opt/zeek/bin/zeek | grep pcap
        libpcap.so.1 => /opt/PF_RING/lib/libpcap.so.1 (0x00007fbbf1c97000)
```

Al reiniciar nuevamente el sistema, se debe mostrar el mensaje anterior.

Nota: si aparece un error como el que se indica a continuación deberemos aplicar el siguiente comando (*apt-get install linux-headers-$(uname -r)*) para que actualice el sistema.

```
make -C /lib/modules/4.9.0-11-amd64/build SUBDIRS=/opt/PF_RING/kernel
EXTRA_CFLAGS='-I/opt/PF_RING/kernel -
DGIT_REV="\"dev:855871a0fc1b2108578109acd2895fc67ed1c662\"" -no-pie -fno-pie'
modules
make[1]: *** /lib/modules/4.9.0-11-amd64/build: No existe el fichero o el
directorio.  Alto.
Makefile:64: fallo en las instrucciones para el objetivo 'all'
make: *** [all] Error 2
```

- Instalamos Zeek a través de los comandos que se indican a continuación.

```
/home/user# cd /tmp
/tmp# git clone --recursive https://github.com/zeek/zeek
/tmp# cd zeek/
/tmp/zeek# ./configure --with-pcap=/zeek/PF_RING -prefix=/opt/zeek/
/tmp/zeek# make
/tmp/zeek# make install
/tmp/zeek# echo "$PATH:/opt/zeek/bin" >/etc/environment
/tmp/zeek# export PATH=/opt/zeek/bin:$PATH
```

- Además de Zeek también instalamos y verificamos el uso de “GeoLite2-City Database” para geolocalización de ips.

```
/tmp# wget http://geolite.maxmind.com/download/geoip/database/GeoLite2-
City.tar.gz
/tmp# tar zxf GeoLite2-City.tar.gz
/tmp#  mv  GeoLite2-City_20191022/GeoLite2-City.mmdb  /usr/share/GeoIP/GeoLite2-
City.mmdb
```

Verificamos que funciona correctamente, pero para ello, es necesario que Zeek se encuentre iniciado.

```
/tmp# /opt/zeek/bin/zeek -e "print lookup_location(8.8.8.8);"
[country_code=US, region=<uninitialized>, city=<uninitialized>,
latitude=37.751, longitude=-97.822]

/tmp# /opt/zeek/bin/zeek -e "print lookup_location(1.1.1.1);"
[country_code=AU, region=<uninitialized>, city=<uninitialized>, latitude=-
33.494, longitude=143.2104]
```

Es importante indicar que se ha instalado Zeek sobre el directorio */opt/zeek/* y que la instalación se ha realiza con permisos de *root*.

Una vez instalado Zeek realizamos los ajustes de configuración de la herramienta para adatarlo a nuestro escenario.

Editamos la configuración de los ficheros siguientes:

- /opt/zeek/etc/node.cfg

- `/opt/zeek/etc/networks.cfg`

Donde destacamos que se despliega Zeek en modo cluster con para la distribución de la carga de trabajo de Zeek:

- manager
- proxy
- 2 worker,s con 2 procesos *lb_procs* y *pf_ring*

```
GNU  nano 2.7.4  Fichero: /opt/zeek/etc/node.cfg
[manager]
type=manager
host=localhost
#
[proxy-1]
type=proxy
host=localhost
#
[worker-1]
type=worker
host=localhost
interface=ens160
lb_method=pf_ring
lb_procs=2
#
[worker-2]
type=worker
host=localhost
interface=ens160
lb_method=pf_ring
lb_procs=2
```

```
GNU nano 2.7.4 Fichero: /opt/zeek/etc/networks.cfg

# List of local networks in CIDR notation.
10.0.0.0/8          Private IP space
172.16.0.0/12       Private IP space
192.168.0.0/16      Private IP space
```

Para no extender en el apartado, se genera un documento con la ejecución de los comandos y sus resultados: ***ANEXO I. Instalación Zeek.***

2.5 Comprobación

Una vez realiza la instalación de Zeek verificamos su funcionamiento y comprobamos los parámetros de configuración.

- Versión de Zeek instalada: 3.1.0-dev.300
- Versión de Zeek Control instalada: 2.0.0-25

Inicializamos Zeek mediante Zeekctl

```
root@vm07pec4zeek:/tmp/zeek# /opt/zeek/bin/zeekctl install
creating policy directories ...
installing site policies ...
generating cluster-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
root@vm07pec4zeek:/tmp/zeek# /opt/zeek/bin/zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
```

```
generating cluster-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping workers ...
stopping proxy ...
stopping manager ...
starting ...
starting manager ...
starting proxy ...
starting workers ...
```

Comprobación de configuración una vez instalado e iniciado Zeek.

```
root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl check
manager scripts are ok.
proxy-1 scripts are ok.
worker-1-1 scripts are ok.
worker-1-2 scripts are ok.
worker-2-1 scripts are ok.
worker-2-2 scripts are ok.
```

Estado de la ejecución de Zeek.

```
root@vm07pec4zeek:/tmp/zeek# /opt/zeek/bin/zeekctl status
Name          Type    Host             Status    Pid    Started
manager       manager localhost        running   16781  12 Dec 01:10:03
proxy-1       proxy   localhost        running   16829  12 Dec 01:10:04
worker-1-1    worker  localhost        running   16911  12 Dec 01:10:06
worker-1-2    worker  localhost        running   16915  12 Dec 01:10:06
worker-2-1    worker  localhost        running   16920  12 Dec 01:10:06
worker-2-2    worker  localhost        running   16919  12 Dec 01:10:06
```

Verificación del los paquete de red leído por cada worker.

```
root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl netstats
 worker-1-1: 1576685343.214763 recvd=735155 dropped=0 link=735155
 worker-1-2: 1576685343.241476 recvd=598284 dropped=0 link=598284
 worker-2-1: 1576685343.262026 recvd=855378 dropped=0 link=855378
 worker-2-2: 1576685343.314558 recvd=953167 dropped=0 link=953167
```

Verificamos a nivel de proceso de Zeek.

```
root@vm07pec4zeek:/home/usuario# lsof -i -P -n | grep zeek
zeek       3332            root   10u  IPv4 710970      0t0  UDP 127.0.0.1:46363->127.0.0.53:53
zeek       3332            root   18u  IPv6 712747      0t0  TCP *:47761 (LISTEN)
zeek       3332            root   20u  IPv6 710279      0t0  TCP 127.0.0.1:47761->127.0.0.1:33160 (ESTABLISHED)
zeek       3332            root   22u  IPv6 710307      0t0  TCP 127.0.0.1:47761->127.0.0.1:33164 (ESTABLISHED)
zeek       3332            root   25u  IPv6 710313      0t0  TCP 127.0.0.1:47761->127.0.0.1:33168 (ESTABLISHED)
zeek       3332            root   26u  IPv6 710314      0t0  TCP 127.0.0.1:47761->127.0.0.1:33172 (ESTABLISHED)
zeek       3332            root   27u  IPv6 710315      0t0  TCP 127.0.0.1:47761->127.0.0.1:33176 (ESTABLISHED)
zeek       3379            root   10u  IPv4 710987      0t0  UDP 127.0.0.1:34270->127.0.0.53:53
zeek       3379            root   17u  IPv6 712767      0t0  TCP *:47762 (LISTEN)
zeek       3379            root   18u  IPv4 712768      0t0  TCP 127.0.0.1:33160->127.0.0.1:47761 (ESTABLISHED)
zeek       3379            root   19u  IPv6 711919      0t0  TCP 127.0.0.1:47762->127.0.0.1:42278 (ESTABLISHED)
zeek       3379            root   20u  IPv6 711920      0t0  TCP 127.0.0.1:47762->127.0.0.1:42282 (ESTABLISHED)
zeek       3379            root   21u  IPv6 711924      0t0  TCP 127.0.0.1:47762->127.0.0.1:42286 (ESTABLISHED)
zeek       3379            root   22u  IPv6 711930      0t0  TCP 127.0.0.1:47762->127.0.0.1:42290 (ESTABLISHED)
zeek       3464            root   10u  IPv4 712797      0t0  UDP 127.0.0.1:49352->127.0.0.53:53
zeek       3464            root   18u  IPv6 712807      0t0  TCP *:47763 (LISTEN)
zeek       3464            root   19u  IPv4 712808      0t0  TCP 127.0.0.1:42286->127.0.0.1:47762 (ESTABLISHED)
zeek       3464            root   20u  IPv4 712809      0t0  TCP 127.0.0.1:33172->127.0.0.1:47761 (ESTABLISHED)
zeek       3466            root   10u  IPv4 710299      0t0  UDP 127.0.0.1:58764->127.0.0.53:53
zeek       3466            root   18u  IPv6 711916      0t0  TCP *:47764 (LISTEN)
zeek       3466            root   19u  IPv4 711917      0t0  TCP 127.0.0.1:42278->127.0.0.1:47762 (ESTABLISHED)
zeek       3466            root   20u  IPv4 711918      0t0  TCP 127.0.0.1:33164->127.0.0.1:47761 (ESTABLISHED)
zeek       3468            root   10u  IPv4 711051      0t0  UDP 127.0.0.1:42338->127.0.0.53:53
zeek       3468            root   18u  IPv6 711927      0t0  TCP *:47765 (LISTEN)
zeek       3468            root   19u  IPv4 711928      0t0  TCP 127.0.0.1:42290->127.0.0.1:47762 (ESTABLISHED)
zeek       3468            root   20u  IPv4 711929      0t0  TCP 127.0.0.1:33176->127.0.0.1:47761 (ESTABLISHED)
zeek       3470            root   10u  IPv4 711040      0t0  UDP 127.0.0.1:60900->127.0.0.53:53
zeek       3470            root   18u  IPv6 710310      0t0  TCP *:47766 (LISTEN)
zeek       3470            root   19u  IPv4 710311      0t0  TCP 127.0.0.1:42282->127.0.0.1:47762 (ESTABLISHED)
zeek       3470            root   20u  IPv4 710312      0t0  TCP 127.0.0.1:33168->127.0.0.1:47761 (ESTABLISHED)
```

Para realizar el mantenimiento Zeek ejecutaremos alguno de los comandos que se indican a continuación.

Parar un nodo

```
root@vm07pec4zeek:/home/user# /opt/zeek/bin/zeekctl stop worker-1-1
stopping worker ...
```

Comprobar un nodo

```
root@vm07pec4zeek:/home/user# /opt/zeek/bin/zeekctl top worker-1-1
Name         Type    Host             Pid     VSize  Rss  Cpu   Cmd
worker-1-1   worker  localhost        <not running>
```

```
root@vm07pec4zeek:/home/user# /opt/zeek/bin/zeekctl status worker-1-1
Name         Type    Host             Status    Pid    Started
```

Iniciar un nodo

```
root@vm07pec4zeek:/home/user# /opt/zeek/bin/zeekctl start worker-1-1
starting worker ...
```

Reiniciar un nodo

```
root@vm07pec4zeek:/home/user# /opt/zeek/bin/zeekctl restart worker-1-1
stopping ...
stopping worker ...
starting ...
starting worker ...
```

Comprobar el estado de los nodos

```
root@serdeb09tmf4:/home/user# /opt/zeek/bin/zeekctl status
Name         Type    Host             Status    Pid    Started
manager      manager localhost        running   1998   26 Oct 12:28:28
proxy-1      proxy   localhost        running   2045   26 Oct 12:28:30
worker-1-1   worker  localhost        running   5910   26 Oct 17:25:01
worker-1-2   worker  localhost        running   2140   26 Oct 12:28:32
worker-1-3   worker  localhost        running   2145   26 Oct 12:28:32
worker-1-4   worker  localhost        running   2149   26 Oct 12:28:32
worker-1-5   worker  localhost        running   2151   26 Oct 12:28:32
```

Comprobar el estado de un nodo

```
root@serdeb09tmf4:/home/user# /opt/zeek/bin/zeekctl stop worker-1-1
stopping worker ...
```

Para nuestro escenario, no será muy relevante, pero podemos conocer el uso de recursos y *Pid* de cada nodo.

```
root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl top
Name         Type    Host             Pid     VSize  Rss  Cpu   Cmd
manager      manager localhost        3332      2G    95M   0%  zeek
proxy-1      proxy   localhost        3379    535M    92M   6%  zeek
worker-1-1   worker  localhost        3464    578M   137M  18%  zeek
worker-1-2   worker  localhost        3466    579M   137M  18%  zeek
worker-2-1   worker  localhost        3468    579M   137M  31%  zeek
worker-2-2   worker  localhost        3470    578M   136M  25%  zeek
```

Por último, es necesario comprobar que se está almacenando información sobre el tráfico.

```
root@vm07pec4zeek:/home/usuario# ls -la /opt/zeek/logs/current/*.log
-rw-r--r-- 1 root root   1040 dic 19 20:24 /opt/zeek/logs/current/capture_loss.log
-rw-r--r-- 1 root root 129898 dic 19 20:30 /opt/zeek/logs/current/conn.log
-rw-r--r-- 1 root root   2459 dic 19 20:29 /opt/zeek/logs/current/dhcp.log
-rw-r--r-- 1 root root 152571 dic 19 20:30 /opt/zeek/logs/current/dns.log
-rw-r--r-- 1 root root  19304 dic 19 20:29 /opt/zeek/logs/current/files.log
-rw-r--r-- 1 root root   4341 dic 19 20:30 /opt/zeek/logs/current/http.log
-rw-r--r-- 1 root root   2344 dic 19 20:24 /opt/zeek/logs/current/notice.log
-rw-r--r-- 1 root root   6697 dic 19 20:23 /opt/zeek/logs/current/ntp.log
-rw-r--r-- 1 root root    533 dic 19 20:12 /opt/zeek/logs/current/ssh.log
-rw-r--r-- 1 root root  12246 dic 19 20:29 /opt/zeek/logs/current/ssl.log
-rw-r--r-- 1 root root  17215 dic 19 20:29 /opt/zeek/logs/current/stats.log
-rw-r--r-- 1 root root   1080 dic 19 20:00 /opt/zeek/logs/current/stderr.log
-rw-r--r-- 1 root root    188 dic 18 17:54 /opt/zeek/logs/current/stdout.log
-rw-r--r-- 1 root root   1311 dic 19 20:29 /opt/zeek/logs/current/weird.log
-rw-r--r-- 1 root root  18487 dic 19 20:29 /opt/zeek/logs/current/x509.log
```

```
root@vm07pec4zeek:/home/user# cat  /opt/zeek/logs/current/conn.log
#separator \x09
#set_separator    ,
#empty_field      (empty)
#unset_field      -
#path             conn
#open             2019-10-26-17-00-11
#fields           ts              uid             id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto           service         duration        orig_bytes      resp_bytes      conn_state      local_orig      local_resp      missed_bytes    history         orig_pkts       orig_ip_bytes   resp_pkts       resp_ip_bytes   tunnel_parents
#types            time            string          addr            port            addr            port            enum            string          interval        count           count           string          bool            bool            count           string          count           count           count           count           set[string]
1572102001.014664 ChmOxL3HvaekoDiwhe 192.168.29.177   54691           192.168.29.17   53              udp             dns             0.009631        0               90              SHR             T               T               0               Cd              118             -               -               -               -
1572102001.111743 C959Yx3b9hpIifApXc 192.168.29.177   52690           192.168.29.17   53              udp             dns             0.002953        0               42              SHR             T               T               0               Cd              70              -               -               -               -
1572102001.129574 CLEMCpaTFrdNT1U21 192.168.29.177   33678           192.168.29.17   53              udp             dns             0.002966        0               30              SHR             T               T               0               Cd              58              -               -               -               -
1572101957.962506 CJrHjoeXOgOKgctY6 192.168.29.10    57021           239.255.255.250 1900            udp             -               2.001133        525             0               S0              T               F               0               D               609             0               0               -               -
1572102001.115722 C7w7H62Tp30VIwMuC4 192.168.29.177   33815           192.168.29.17   53              udp             dns             0.003357        0               30              SHR             T               T               0               Cd              58              -               -               -               -
1572102001.120221 CyTbbp2qxybrHdPFpg 192.168.29.177   50590           192.168.29.17   53              udp             dns             0.002958        0               42              SHR             T               T               0               Cd              70              -               -               -               -
1572102001.312218 C2Zav12YzrQv1IqUH 192.168.29.177   33355           192.168.29.17   53              udp             dns             0.004697        0               30              SHR             T               T               0               Cd              58              -               -               -               -
1572102001.317895 CxVhQv154gajcMI1fe 192.168.29.177   39445           192.168.29.17   53              udp             dns             0.004202        0               42              SHR             T               T               0               Cd              70              -               -               -               -
1572101961.795536 CmSrI14LO9CroC1p19 192.168.88.1     5678            255.255.255.255 5678            udp             -               -               -               -               S0              T               F               0               D               1               139             0               0               -
1572102051.650685 C0NO3fyJkeYQPbpfg 192.168.29.10    5353            224.0.0.251     5353            udp             dns             -               -               -               S0              T               F               0               D               1               68              0               0               -
```

```
root@vm07pec4zeek:/home/user# cat /opt/zeek/logs/current/conn.log | /opt/zeek/bin/zeek-
cut id.orig_h id.orig_p id.resp_h id.resp_p duration
192.168.29.177    54691 192.168.29.17     53    0.009631
192.168.29.10     57021 239.255.255.250   1900  2.001133
192.168.29.177    33815 192.168.29.17     53    0.003357
192.168.29.10     54526 239.255.255.250   1900  3.003690
192.168.29.15     5353  224.0.0.251 5353  -
192.168.29.10     5353  224.0.0.251 5353  -
192.168.29.15     5353  224.0.0.251 5353  6.552363
192.168.29.34     51497 239.255.255.250   1900  0.500576
192.168.29.15     5353  224.0.0.251 5353  -
192.168.29.10     56029 239.255.255.250   1900  3.001055
192.168.29.34     51497 239.255.255.250   1900  0.497884
192.168.29.15     5353  224.0.0.251 5353  -
192.168.88.1      5678  255.255.255.255   5678  59.993968
192.168.29.15     5353  224.0.0.251 5353  8.561932
192.168.29.32     38536 239.255.255.250   1900  17704.481976
192.168.29.10     56297 239.255.255.250   1900  2.002817
```

3. Análisis de la herramienta ELK (Elasticsearch, Logstahs, Kibana)

3.1 Descripción

ELK son las siglas de Elasticsearch, Logstahs y Kibana. Son tres proyectos *open source.* La compañía (Elastic Inc) que desarrolla este conjunto o *stack* de herramientas nació con la idea de Shay Banon para desarrollar una aplicación para las búsquedas para un libro de recetas, mientras que Jordan Sissel estaba trabajando con Logstash para gestión de logs y Rashid Khan para la creación de Kibana, un entorno gráfico.

Elasticsearch[12] es un motor de análisis de datos tanto estructurados como no estructurados desarrollado en Apache Lucene[13]. Destaca principalmente por el uso sencillo de API REST. Permite una gran escalabilidad y versatilidad. Elasticsearch puede nutrirse de una gran variedad de fuentes de información, donde procesa los datos, los parsea y los indexa.

Logstash[14] es un *pipeline* de procesamiento de datos para la ingesta de datos con el objetivo de parsear, enriquecer, filtrar y posteriormente enviarlos a Elasticsearch o cualquier otra fuente.

Kibana[15] es la herramienta de visualización y gestión de datos que utiliza Elasticsearch para mostrar información. Permite la visualización en tiempo real, mediante grafos, mapas, etc.

3.2 Estructura

En este apartado explicaremos brevemente la arquitectura que compone el conjunto de herramientas ELK. Para el desarrollo del trabajo se utiliza la versión estable de ELK que se encuentra disponible, en concreto la versión 7.5 (release date: Diciembre, 2019)

Elasticsearch es el motor de búsqueda basado en Apache Lucene desarrollado en java. Es una aplicación potente, personalizable y escalable.

Kibana es el componente para la representa de información a modo de panel de control de Elasticsearch.

Logstash es el componente para la gestión de logs, tratamiento y volcado de datos enriquecidos.

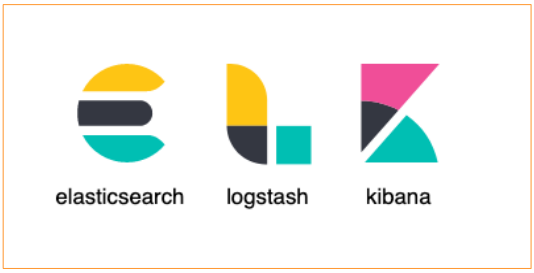

Figura 4: Logotipo de Stack ELK

12 Elasticsearch: https://www.elastic.co/es/products/elasticsearch
13 Apache Lucene: https://lucene.apache.org/
14 Logstash: https://www.elastic.co/es/products/logstash
15 Kibana: https://www.elastic.co/es/products/kibana

Es importante destacar que el flujo de comunicación entre los componentes del stack ELK no tiene por qué ser lineal, y dependerá de la necesidad del proyecto. Para el componente Logstash, se basa en tres módulos para realizar la gestión de los datos; *inputs*, *filters* y *outputs*.

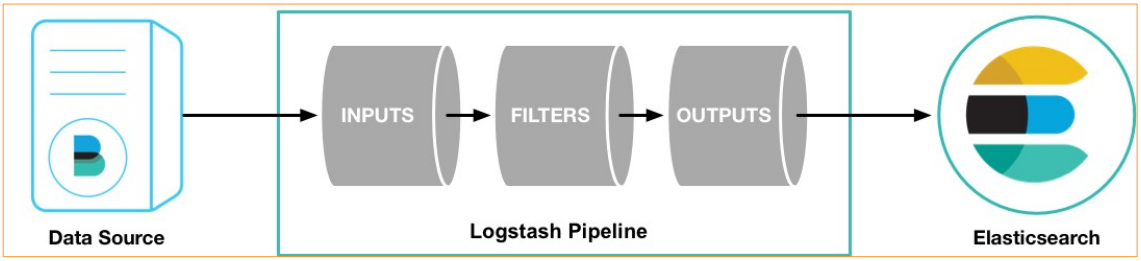

Figura 5: Logstash Pipeline

3.3 Requisitos

El stack de ELK se puede desplegar tanto en entorno windows como en entornos unix. ELK ofrece una matriz de compatibilidades[16], la cual se recomienda consultar para evitar errores de diseño.

Los requisitos mínimos, a nivel de recursos, depende en gran medida del entorno que se quiere desarrollar.

- ELK solo esta soportado para arquitecturas x86_64.
- El cifrado (via *dm-crypt*) solo está soportado en entorno Linux Os
- Requiere Java Oracle/OpenJDK

Para el despliegue utilizaremos la maquina anteriormente desplegada, la cual tenia las siguientes características técnicas.

- **Sistema Operativo**: Linux 5.3.0-24-generic Ubuntu x86_64 GNU/Linux
- **Información disco duro**: 60 GB
- **Información RAM**: 4 GB

- **Hostname**: vm07pec4elk
- **Dirección MAC**: 52:54:00:9a:5d:a8
- **Dirección IP**: 192.168.29.87
- **Núcleos CPU**: 2
- **Memoria RAM**: 4GB

3.4 Despliegue

En este aparatado facilitaremos los pasos seguidos para completar la instalación de los diferentes componentes del stack de ELK. Para ellos nos basaremos en la información oficial que ofrece el fabricante, la cual es muy completa y que se puede consultar en los siguientes enlaces.

- https://www.elastic.co/guide/en/elasticsearch/reference/current/deb.html
- https://www.elastic.co/guide/en/kibana/current/deb.html
- https://www.elastic.co/guide/en/logstash/current/installing-logstash.html

16 Matriz de compatibilidades ELK: https://www.elastic.co/es/support/matrix

Seguimos los pasos de referencia como se indica a continuación, teniendo en cuenta que partimos de la maquina host anterior.

Configuraciones generales para ELK

Añadimos las claves para la descarga de las fuentes oficiales.

```
# wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch |  apt-key add -
```

```
# apt-get install apt-transport-https
```

Añadimos al repositorio de fuentes la entrada para descarga de ELK

```
#  echo  "deb  https://artifacts.elastic.co/packages/7.x/apt  stable  main"  |  tee  -a
/etc/apt/sources.list.d/elastic-7.x.list
```

Instalación de Elasticsearch

A continuación, se indican los comandos de instalación para Elasticsearch.

```
# apt-get update &&  apt-get install elasticsearch
```

Configuración del servicio para que Elasticsearch se inicia automáticamente.

```
# /bin/systemctl daemon-reload
# /bin/systemctl enable elasticsearch.service
```

Tareas de mantenimiento: iniciar, parar, reiniciar y comprobar estado

```
# systemctl start elasticsearch.service
# systemctl stop elasticsearch.service
# systemctl status elasticsearch.service
# systemctl restart elasticsearch.service
```

Para comprobar si efectivamente Elasticsearch esta iniciado y sin fallos ejecutamos el siguiente comando que nos debería mostrar un resultado similar.

```
# curl -X GET "localhost:9200/?pretty"
{
  "name" : "serdeb09tmf4",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "HIVKTHO0RbqrFkFUtaV7PQ",
  "version" : {
    "number" : "7.4.1",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "fc0eeb6e2c25915d63d871d344e3d0b45ea0ea1e",
    "build_date" : "2019-10-22T17:16:35.176724Z",
    "build_snapshot" : false,
    "lucene_version" : "8.2.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

En este apartado configuramos los parámetros adecuados para nuestro entorno de Elasticsearch.

- Ruta: `/etc/elasticsearch/`
- Fichero: `elasticsearch.yml`

•

- Descripción: fichero de configuración de Elasticsearch.

```
# ------------ Network -------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
network.host: localhost
#
# Set a custom port for HTTP:
#
http.port: 9200
#
```

- Ruta: `/etc/elasticsearch/`
- Fichero: `jvm.options`
- Descripción: fichero de configuración de Java.

  Dejaremos las opciones de java por defecto, donde la más relevante es la reserva de memoria de java para Elasticserarch (1GB).

```
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms1g
-Xmx1g
```

- Ruta: `/etc/elasticsearch/`
- Fichero: `log4j2.properties`
- Descripción: fichero de configuración de Elasticsearch.

  No realizaremos ninguna modificación del fichero ya que no es necesario. Para mayor detalle se puede consultar la documentación oficial[17].

Instalación de Kibana

A continuación, se indican los comandos de instalación para Kibana.

```
# apt-get install kibana
```

Configuración del servicio para que Kibana se inicia automáticamente.

```
# /bin/systemctl daemon-reload
# /bin/systemctl enable kibana.service
```

En este apartado configuramos los parámetros adecuados para nuestro entorno de Kibana.

- Fichero: `kibana.yml`
- Ruta: `/etc/kibana/`
- Descripción: fichero de configuración de Elasticsearch.

```
# Kibana is served by a back end server.
#
server.port: 5601
server.host: "localhost"
# The Kibana server's name.  This is used for display purposes.
```

17 Parámetros de *log4j2*: https://www.elastic.co/guide/en/elasticsearch/reference/current/logging.html

```
server.name: "kibana_host"
#
```

Tareas de mantenimiento: iniciar, parar, reiniciar y comprobar estado

```
# systemctl start kibana.service
# systemctl stop kibana.service
# systemctl restart kibana.service
# systemctl status kibana.service
```

Instalación de Nginx

Como Kibana no ofrece seguridad a nivel de acceso, se realiza la instalación del servicio Nginx para asegurar el portal, por lo que a continuación se indican las instrucciones ejecutadas.

```
# apt-get install nginx
```

```
# apt-get install ssl-cert
```

```
# echo "admin:$(openssl passwd -apr1 my-personal-password)" |  tee -a
/etc/nginx/htpasswd.kibana
admin:$apr1$672ZVLKN$35re9SFDiN69m5wnmV57n1
```

```
# rm -f /etc/nginx/sites-enabled/default
```

```
# nano /etc/nginx/sites-available/kibana
```

- Fichero: `kibana`
- Ruta: `/etc/nginx/sites-available/`
- Descripción: fichero de configuración de NGINX para el acceso web a Kibana. Indicar que se utilizan los certificados autofirmados *snakeoil*

```
server {
    listen 80 default_server;
    server_name _;
    return 301 https://$server_name$request_uri;
}
server {
    listen 443 default_server ssl http2;
    server_name _;
    ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;
    ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key;
    ssl_session_cache shared:SSL:10m;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.kibana;
    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Creamos un enlace simbolico

```
# ln -s /etc/nginx/sites-available/kibana /etc/nginx/sites-enabled/kibana
```

Verificamos que no hay errores en Nginx

```
# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
```

Configuración del servicio para que Kibana se inicia automáticamente.

```
# /bin/systemctl daemon-reload
# /bin/systemctl enable nginx.service
```

Tareas de mantenimiento: iniciar, parar, reiniciar y comprobar estado

```
# systemctl start nginx.service
# systemctl stop nginx.service
# systemctl restart nginx.service
# systemctl status nginx.service
```

En este momento ya es posible acceder al portal web de Kibana, pero debemos tener en cuenta que éste se encuentra sin contenido ya que todavía no hemos finalizado el proceso completo de flujo de información para su representación.

Se debe tener en cuenta que es necesario introducir credenciales para el acceso al portal de Kibana, y que esta ha sido configura en un apartado anterior:

- Usuario: `admin`
- Password: `my-personal-password`

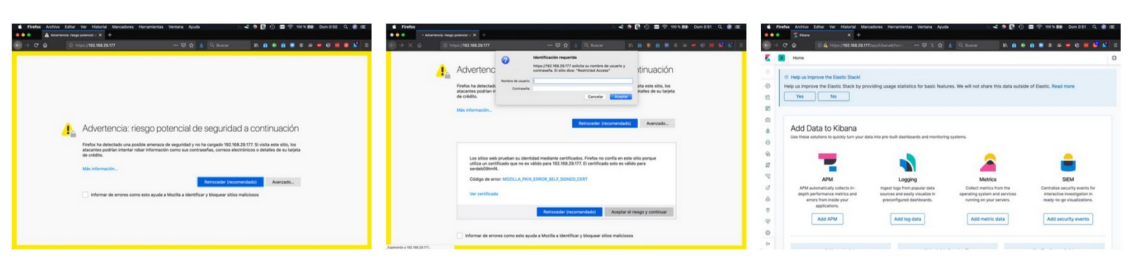

Figura 6: Inicio Panel de Control Kibana

Instalación de Logstash

Para la instalación de Logstash, según el manual, es necesario verificar la versión de java instalada en el host, y como se puede observar cumple los requerimientos de Logstash

```
# java -version
openjdk version "11.0.5" 2019-10-15
OpenJDK Runtime Environment (build 11.0.5+10-post-Ubuntu-0ubuntu1.1)
OpenJDK 64-Bit Server VM (build 11.0.5+10-post-Ubuntu-0ubuntu1.1, mixed mode, sharing)
```

```
# apt-get install logstash
```

Para configurar logstash para que pueda abrir un puerto de escucha y recibir datos de fuentes externas es necesario configurar los datos de conexión mediante los ficheros que a continuación de detallan. Es necesario configurar el

input (puerto de entrada), el *filter* (tratamiento de los datos) y el *output* (salida de los datos) Para nuestro trabajo, utilizamos tres ficheros para segmentar su configuración.

- Ruta: `/etc/logstash/conf.d/`
- Fichero: `01-input-beats.conf`
- Descripción: en este fichero se configura la entrada de datos a Logstash. Las entradas pueden ser de muchas fuentes, como puede ser logs de equipo linux, windows, logs de aplicación, ficheros, etc. Para nuestro entorno indicamos que el puerto que escucha es el 5044.

```
##################################################
# Inputs are used to ingest logs from remote logging clients
##################################################
input {
  # Ingest logs that match the Beat template
  beats {
    # Accept connections on port 5044
    port => 5044

  }
}
```

- Ruta: `/etc/logstash/conf.d/`
- Fichero: `11-filter-beats.conf`
- Descripción: en este fichero se indica que tratamiento se le dan a los datos que van a procesas y que filtro utilizará. Para nuestro trabajo, indicamos que los datos que procedan de una fuente con el tag "`zeek`" exclusivamente serán procesados y transformados en formato JSON.

```
##################################################################
# Filters are used to transform and modify the logs
##################################################################
filter {
# Only apply these transformations to logs that contain the "zeek_filebeat"
tag
    if "zeek" in [tags] {
        # Extract the json into Key value pairs
        json {
            source => "message"
        }
    }
}
```

- Ruta: `/etc/logstash/conf.d/`
- Fichero: `21-elasticsearch-output.conf`
- Descripción: en este fichero se indica que hacer con la información resultante tratada por el filtro. Se puede enviar a otra fuente, enviar a Elasticsearch, etc … Para nuestro trabajo, los datos con el tag "`zeek`" será enviados a Elasticsearch e indexados como "`filebeat-zeek-%{+YYYY.MM.dd}`"

```
##################################################################
# Outputs take the logs and output them to a long term storage
##################################################################
output {
  # Send logs that contain the zeek tag too
  if "zeek" in [tags] {
    # Outputting logs to elasticsearch
    elasticsearch {
      # ES host to send logs too
```

```
      hosts => ["http://localhost:9200"]

      # Index to store data in
      index => "filebeat-zeek-%{+YYYY.MM.dd}"
    }
  }
}
```

Configuración del servicio para que Logstash se inicia automáticamente.

```
# /bin/systemctl daemon-reload
# /bin/systemctl enable logstash.service
```

Tareas de mantenimiento: iniciar, parar, reiniciar y comprobar estado

```
# systemctl start logstash.service
# systemctl stop logstash.service
# systemctl restart logstash.service
# systemctl status logstash.service
```

3.5 Comprobación

En este apartado se pretende listar una serie de comandos que se pueden utilizar para conocer en qué estado se encuentra el stack ELK entre las que se destacas la revisión de logs y servicios para poder identificar posibles errores.

Revisión de logs por herramientas

- Elasticsearch: `/var/elasticsearch/*`
- Kibana: `/var/kibana/*`
- Nginx: `/var/nginx/*`
- Logstash: `/var/logstash/*`

Revisión de estado de servicios

- Elasticsearch: `lsof -i -P -n | grep elasticsearch`
- Kibana: `lsof -i -P -n | grep kibana`
- Nginx: `lsof -i -P -n | grep nginx`
- Logstash: `lsof -i -P -n | grep logstash`

Se debe tener en cuenta los puertos que se ha configurado o se han dejado por defecto. A continuación, se identifican en base a la configuración realizada anteriormente.

Aplicación	*Listener*
Elasticsearch	9200, 9300
Kibana	5601
Nginx	80, 443
Logstash	5044

4. Método de integración de Zeek y ELK

4.1 Formato de logs

En este apartado se revisa el formato de los logs de Zeek y la instalación de Fileeat para enviar los logs a Zeek

Logs de Zeek

Como se ha comentado anteriormente, Zeek guarda una muestra de los datos de red ya que se trata de una aplicación de escucha de red. Zeek almacena la información de red el directorio `/opt/zeek/logs/`, donde crear una carpeta por cada día y una carpeta con los logs activos en ese momento `/opt/zeek/logs/current/`. Realiza la rotación de los logs cada hora. Todas estas opciones son personalizables a través de los ficheros de configuración.

```
root@vm07pec4zeek:/home/user# ls -la /opt/zeek/logs/
total 32
drwxr-xr-x 4 root root  4096 oct 27 02:33 .
drwxr-xr-x 9 root root  4096 oct 26 11:06 ..
drwxr-xr-x 2 root root 12288 oct 27 02:34 2019-10-26
drwxr-xr-x 2 root root 12288 oct 27 08:49 2019-10-27
lrwxrwxrwx 1 root root    23 oct 27 02:33 current -> /opt/zeek/spool/manager
```

Zeek generar por cada tipo de dato un fichero de logs. Estos ficheros son entendibles (datos ASCII) y siguen una estructura definida por Zeek. Los ficheros de datos que podemos encontrar en el directorio de logs hacen referencia los protocolos de red (*conn.log, dhcp.log, dns.log, http.log, smtp.log, ssh.logs*, entre otros), ficheros (*files.log, oscp.log*), de detección (*intel.log, notice.log*), de observación de red (known_certs.log, software.log, known_services.log), de tipo diverso (*weird.log, barnyard2.log)* y de diagnóstico (*broker.log, cluster.log, stats.log, stderr.log*). Para mayores detalles sobre cada uno de los ficheros que es capaz Zeek de genera es conveniente visitar la documentación oficial[18].

A nivel informativo se muestra la estructura de datos del fichero `coon.log` donde destacamos la información de separador (`separator \x09`), los campos (`fields`) y el tipo de campo (`types`). Esta información es útil para su tratamiento posterior.

```
# cat /opt/zeek/logs/current/conn.log
#separator \x09
#set_separator    ,
#empty_field      (empty)
#unset_field      -
#path conn
#open 2019-10-27-09-03-10
#fields     ts    uid   id.orig_h   id.orig_p   id.resp_h   id.resp_p   proto
      service     duration    orig_bytes  resp_bytes  conn_state  local_orig local_resp
      missed_bytes      history     orig_pkts   orig_ip_bytes     resp_pkts
      resp_ip_bytes     tunnel_parents
#types      time  string      addr  port  addr  port  enum  string      interval
      count count string      bool  bool  count string      count count count
      count set[string]
```

18 Ficheros de Logs de Zeek: https://docs.zeek.org/en/stable/script-reference/log-files.html

Con la herramienta de línea de comandos de Zeek (*Zeek-Cut)* se puede trabajar con los logs para extraer información útil pero cuando el volumen de información es tan grande es complejo de gestionar.

Para poder hacer un tratamiento accesible al usuario final se transforman los logs a formato JSON para que Logstash pueda procesarlos. Para convertir a JSON los logs de Zeek creados los scripts de Zeek siguientes:

```
## Configurar Zeek para que escriba los logs en formato JSON
mkdir -p /opt/zeek/share/zeek/site/scripts
tee      /opt/zeek/share/zeek/site/scripts/json-logs.zeek << EOF

@load tuning/json-logs
redef LogAscii::json_timestamps = JSON::TS_ISO8601;
redef LogAscii::use_json = T;
EOF
```

```
tee -a /opt/zeek/share/zeek/site/local.zeek << EOF

# Load policy for JSON output
@load scripts/json-logs
EOF
```

Para que los cambios surjan efecto debemos reiniciar el servicio de Zeek.

```
# /opt/zeek/bin/zeekctl stop
# /opt/zeek/bin/zeekctl deploy
# /opt/zeek/bin/zeekctl status
```

Por lo que ahora el formato de logs que podemos observar en Zeek es el siguiente:

```
# cat /opt/zeek/logs/current/conn.log
{"ts":"2019-10-
27T11:17:04.873857Z","uid":"CKb0Pg2jyQZspx7s8e","id.orig_h":"192.168.29.15","id.orig_p"
:5353,"id.resp_h":"224.0.0.251","id.resp_p":5353,"proto":"udp","service":"dns","duratio
n":2.5987625122070313e-
05,"orig_bytes":556,"resp_bytes":0,"conn_state":"S0","local_orig":true,"local_resp":fal
se,"missed_bytes":0,"history":"D","orig_pkts":2,"orig_ip_bytes":612,"resp_pkts":0,"resp
_ip_bytes":0}
```

Agente de gestión de logs (Beat)

El objetivo es enviar los logs de Zeek a ELK. Para ello se utiliza el agente de ELK llamado Filebeat. Este agente permite enviar los datos de Zeek a Elasticsearch directamente o a Logstash para que los procese previamente.

De este modo al realizar los ajustes de configuración oportunos, es posible transferir los datos de manera regular desde el host donde se ha desplegado Zeek al host donde se ha desplegado ELK.

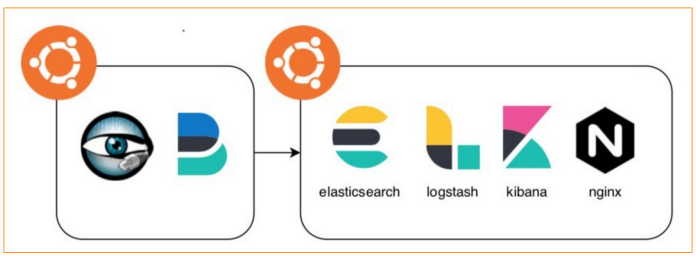

Figura 7: Flujo de logs con Beat (filebeat)

Instalación y configuración de agente Filebeat

Para la instalación de Filebeat se utiliza las referencias de la documentación oficial del producto que se encuentra documentada de Elasticsearch.

```
# apt-get install filebeat
```

A continuación configuramos Filebeat con nuestras necesidades.

- Ruta: `/etc/filebeat/`
- Fichero: `filebeat.yml`
- Descripción: en este fichero se configura la entrada de datos a Logstash. Las entradas pueden ser de muchas fuentes, como puede ser, logs de equipo linux, windows, logs de aplicación, ficheros, etc

```
#=========================== Filebeat inputs =============================
filebeat.inputs:

- type: log
  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /opt/zeek/logs/current/*.log

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

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

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

#==================== Elasticsearch template setting ==========================
setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false
#================================ General =====================================
# The tags of the shipper are included in their own field with each
# transaction published.
tags: ["zeek_filebeat"]

#============================== Kibana =====================================
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana
API.
# This requires a Kibana endpoint configuration.
# setup.kibana:
  #host: "localhost:5601"
#============================= Outputs ======================================
# Configure what output to use when sending the data collected by the beat.
#-------------------------- Elasticsearch output ------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost:9200"]
#----------------------------- Logstash output --------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["localhost:5044"]

#================================ Processors =====================================
# Configure processors to enhance or manipulate events generated by the beat.
processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
```

Configuración del servicio para que Logstash se inicia automáticamente.

```
# /bin/systemctl daemon-reload
# /bin/systemctl enable filebeat.service
```

Tareas de mantenimiento: iniciar, parar, reiniciar y comprobar estado

```
# systemctl start filebeat.service
# systemctl stop filebeat.service
# systemctl restart filebeat.service
# systemctl status filebeat.service
```

Revisión de estado del servicio Filebeat

- Filebeat: `lsof -i -P -n | grep filebeat`

4.2 Comprobación funcional

Es importante cada vez que se realiza un cambio de configuración reiniciar los servicios y analizar los logs.

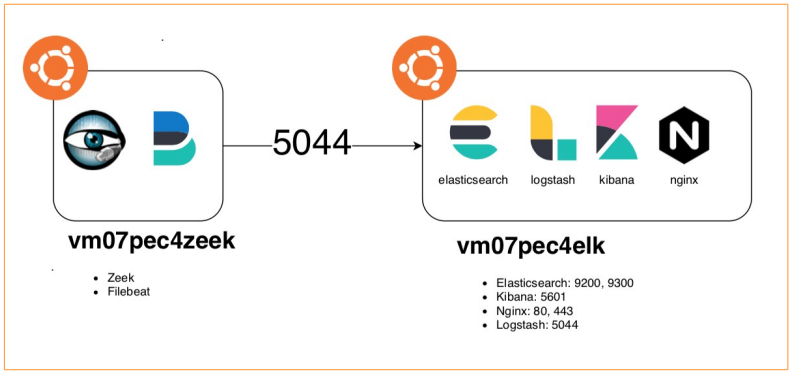

Figura 8: Representación Host Zeek y ELK

Se aplica el comando de parada de los servicios en cada máquina.

```
# /opt/zeek/bin/zeekctl stop
# systemctl stop filebeat.service
# systemctl stop logstash.service
# systemctl stop elasticsearch.service
# systemctl stop kibana.service
```

Comandos para el inicio de los servicios en cada máquina.

```
# /opt/zeek/bin/zeekctl deploy
# systemctl start filebeat.service
# systemctl start logstash.service
# systemctl start kibana.service
# systemctl start elasticsearch.service
```

Verificación del estado de cada servicio para asegurar que no se producen errores en el proceso de inicio que pueda afectar al servicio.

```
root@vm07pec4elk:/home/transferencia# systemctl status elasticsearch
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/lib/systemd/system/elasticsearch.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-12-17 01:18:09 CET; 2 days ago
     Docs: http://www.elastic.co
 Main PID: 13082 (java)
    Tasks: 81 (limit: 4624)
```

```
   Memory: 1.4G
   CGroup: /system.slice/elasticsearch.service
           ├─13082 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.netw
           └─13189 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller
```

```
root@vm07pec4elk:/home/transferencia# systemctl status kibana
● kibana.service - Kibana
   Loaded: loaded (/etc/systemd/system/kibana.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-12-16 08:35:04 CET; 3 days ago
 Main PID: 829 (node)
    Tasks: 11 (limit: 4624)
   Memory: 297.4M
   CGroup: /system.slice/kibana.service
           └─829 /usr/share/kibana/bin/../node/bin/node /usr/share/kibana/bin/../src/cli -c /etc/ki

dic 19 23:12:02 vm07pec4elk kibana[829]: {"type":"response","@timestamp":"2019-12-19T22:12:01Z","ta
dic 19 23:12:30 vm07pec4elk kibana[829]: {"type":"response","@timestamp":"2019-12-19T22:12:30Z","ta
dic 19 23:12:31 vm07pec4elk kibana[829]: {"type":"response","@timestamp":"2019-12-19T22:12:31Z","ta
```

```
root@vm07pec4elk:/home/transferencia# systemctl status logstash
● logstash.service - logstash
   Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-12-17 13:47:45 CET; 2 days ago
 Main PID: 20435 (java)
    Tasks: 55 (limit: 4624)
   Memory: 569.8M
   CGroup: /system.slice/logstash.service
           └─20435 /bin/java -Xms1g -Xmx1g -XX:+UseConcMarkSweepGC -XX:CMSInitiatingOccupancyFracti

dic 19 23:16:26 vm07pec4elk logstash[20435]:        "@version" => "1",
dic 19 23:16:26 vm07pec4elk logstash[20435]:             "ecs" => {
```

```
root@vm07pec4elk:/home/transferencia# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-12-16 08:35:04 CET; 3 days ago
     Docs: man:nginx(8)
  Process: 808 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, st
  Process: 846 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU
 Main PID: 851 (nginx)
    Tasks: 3 (limit: 4624)
   Memory: 4.7M
   CGroup: /system.slice/nginx.service
           ├─851 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
           ├─852 nginx: worker process
           └─853 nginx: worker process
```

```
root@vm07pec4zeek:/home/usuario# systemctl status filebeat
● filebeat.service - Filebeat sends log files to Logstash or directly to Elasticsearch.
   Loaded: loaded (/lib/systemd/system/filebeat.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-12-16 08:19:45 CET; 3 days ago
     Docs: https://www.elastic.co/products/beats/filebeat
 Main PID: 18235 (filebeat)
    Tasks: 17 (limit: 4915)
   Memory: 32.2M
   CGroup: /system.slice/filebeat.service
           └─18235 /usr/share/filebeat/bin/filebeat -e -c /etc/filebeat/filebeat.yml -path.home /usr

dic 19 23:18:09 vm07pec4zeek filebeat[18235]: 2019-12-19T23:18:09.314+0100        INFO        log/ha
dic 19 23:18:14 vm07pec4zeek filebeat[18235]: 2019-12-19T23:18:14.361+0100        INFO        log/ha
dic 19 23:18:15 vm07pec4zeek filebeat[18235]: 2019-12-19T23:18:15.633+0100        INFO        [monit
```

```
root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl status
Name         Type    Host             Status    Pid    Started
manager      manager localhost        running   3332   18 Dec 17:54:42
proxy-1      proxy   localhost        running   3379   18 Dec 17:54:43
worker-1-1   worker  localhost        running   3464   18 Dec 17:54:45
worker-1-2   worker  localhost        running   3466   18 Dec 17:54:45
worker-2-1   worker  localhost        running   3468   18 Dec 17:54:45
worker-2-2   worker  localhost        running   3470   18 Dec 17:54:45
```

Una vez verificados los servicios es necesario comprobar si se está gestionada por parte de Logstash los logs de manera adecuada, donde se verifica que los logs enviados por Filebeat deben pasar los módulos *inputs*, *filter* y *output*, para ser indexados por Elasticsearch.

Por tanto, los pasos que se deben seguir son los siguientes:

Acceso al portal de Kibana

Introduciremos la IP del host donde hemos desplegado Kibana. Al utilizar un certificado auto firmado (*ssl-cert-snakeoil.pem* y *ssl-cert-snakeoil.key*) como hemos indicado anteriormente, deberemos aceptar el mensaje informativo e introducir las credenciales para visualizar el panel de control de Kibana. Las imágenes serán similares a las que se muestran en la figura *Inicio Panel de Control Kibana.*

Indexación de contenido.

Mediante capturas de pantalla se muestra el proceso de indexado de información de datos de los logs de Zeek en Kibana.

Accedemos al panel de Kibana y encontramos en la lado izquierdo inferior la opción "*Management*" donde nos mostrará la ventana para crear un nuevo índice. Haremos clic en "*Create index patterns*" y nos mostrara una nueva ventana de dos pasos.

El primer paso es introducir el patrón del índice que queremos agregar, en nuestro caso "`filebeat-*`" y al introducirlo nos indica mediante un check de confirmación que ha encontrado el índice (*Success! Your index pattern matches 1 index*) por lo que hacemos clic en "*Next step*".

En el segundo paso tenemos que indicar el filtro con el que queremos que se referencie nuestra información a nivel de tiempo. Para nuestro caso utilizamos "*@timestamp*", con lo que creamos nuestro patrón de índice haciendo clic en "*Create index pattern".*

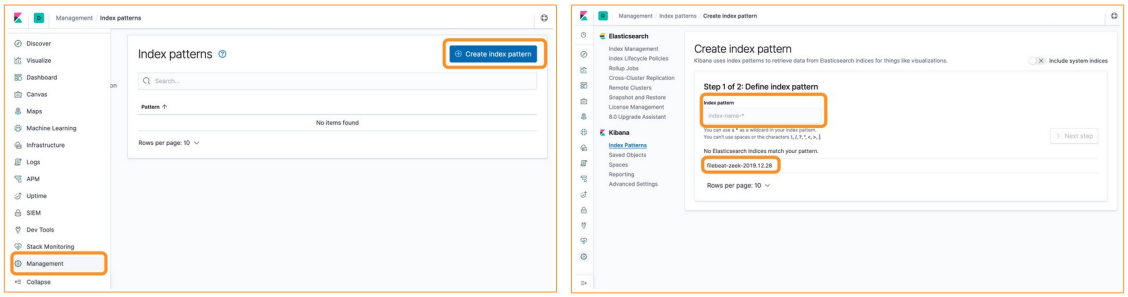

Figura 9: Proceso de Indexado de datos 1/3

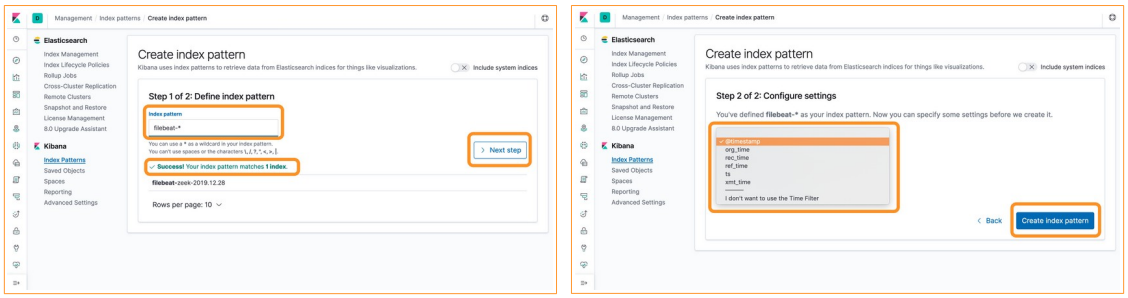

Figura 10: Proceso de Indexado de datos 2/3

Una vez se ha creado el índice nos muestra información de los campos y tipos de campos que se han agregado.

En la imagen siguiente podemos ver, como al poco tiempo del indexado, se muestra información a través del panel boto de "*Discover*" que se encuentra en la lado izquierdo superior. En este panel podemos ver un apartado para la búsqueda, los tiempos de muestra de información, los campos, los mensajes y la tendencia muestreo de mensajes.

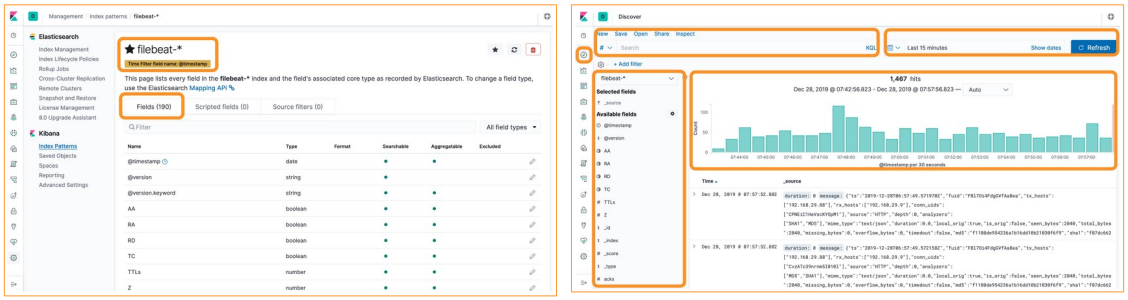

Figura 11: Proceso de Indexado de datos 3/3

4.3 Descripción

Se utiliza las aplicación Filebeat como motor de integración con ELK, de modo que se ha conseguido enviar los datos de los logs de Zeek a ELK y su posterior indexación. De este modo ahora es posible consultar los datos que envía Zeek y analizarlos.

A nivel de configuración ha sido necesario realizar la instalación de Filebeat en el nodo de Zeek, realizar los ajustes propios para el envío de los datos a ELK y crear el script propio en Zeek para la conversión de datos al formato JSON y conversión de fecha.

5. Representación de datos de Zeek en ELK

5.1 Investigación

El objetivo de este apartado es realizar la representación de los datos obtenidos a través de Zeek en un panel de control intuitivo y claro. A continuación, se describen los pasos a seguir y los elementos que componen el panel de Kibana.

Panel Discover

Desde el apartado "*Discover*" se realizan todas las funciones y personalizarlas en base a nuestras necesidades. A continuación se describen los apartados más significativos:

Cuadro de búsqueda: para introducir los datos de búsqueda en base a la información indexada, que debe seguir un modelo concreto, pero que con la ayuda se pueden generar las búsquedas deseadas.

Cuadro de tiempos: donde podremos indicar la ventana temporal que queremos visualizar, así como el refresco de los datos.

Creación de filtro: que nos permite crear un filtro concreto para mejorar las búsquedas.

Campos: se trata de la columna con todos los campos que hemos obtenido de Zeek, más unos campos que son añadidos por ELK para gestión. En esta columna podemos encontrar la información que necesitamos para nuestras búsquedas.

Cuadro de datos: donde se muestran los mensajes con la información en base a nuestra búsqueda.

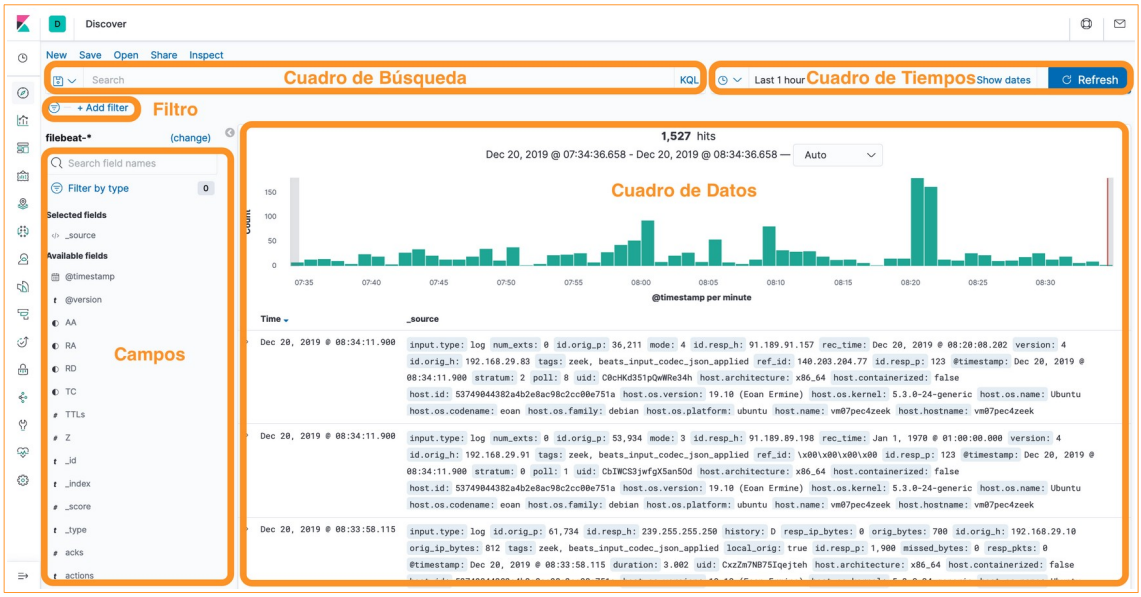

Figura 12: Panel Discover

Búsqueda personalizada

Se crean los filtros necesarios, en concreto los que se indican a continuación y posteriormente se guardaremos para ser utilizados en el panel de control:

- Search – Ip Destino (*id.resp_h :**)
- Search – Ip Origen (*id.orig_h :**)
- Search – Puertos Destino (*id.resp_p :**)
- Search – Servicios (*service : **)
- Search – Protocolos (*proto : **)

Ejemplarizamos una de las búsquedas y como guardarla para su posterior uso.

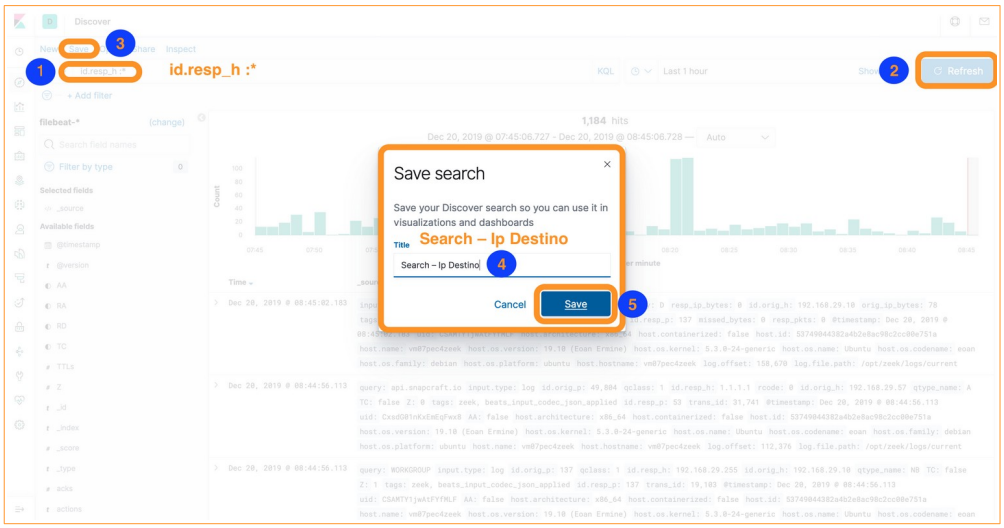

Figura 13: Crear búsqueda

Cuadro de búsqueda: introducimos los valores de campo que queremos buscar, en nuestro caso queremos buscar las IP de destino que estamos analizando de la red, por lo que buscamos por el campo indicado (*id.resp_h: **)

Cuadro de tiempos: introducimos el valor de tiempo que nos interesa, para nuestro ejemplo la última hora (*Last 1 hour*)

Cuadro de datos: nos muestra la información en base a la búsqueda realizada.

Guardar: hacemos click en *save* para guardar la búsqueda personalizada y asignarle el nombre descriptivo que consideremos (Search – Ip Destino).

Todas las búsquedas personalizada que realizamos las encontraremos listadas en el panel de *Management* » *Kibana* » *Saved Objets*

Visualización de datos

Una vez creado la búsqueda es conveniente representar los datos de manera amigable para que puedan ser entendidos por lo que nos dirigimos al apartado de *Visualize.*

En este apartado creamos la vista para cada una de las búsquedas. Estas vistas pueden ser de diferentes tipos por lo que debemos elegir la más adecuada a nuestra necesidad de representación.

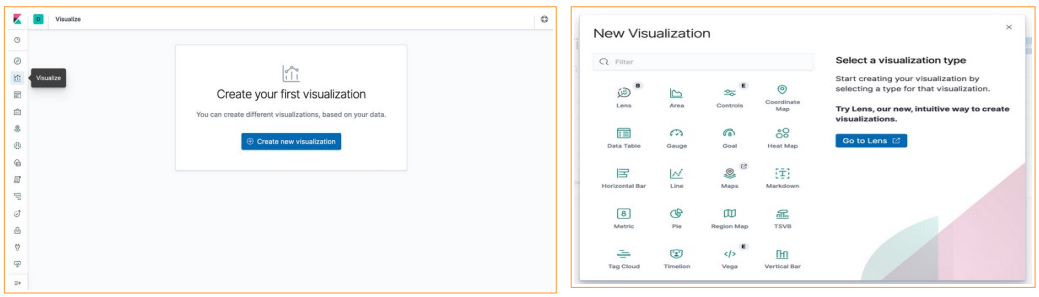

Figura 14: Visualize / Tipos de Visualización

A modo de ejemplo presentamos como realizar la representación de tipo *Pie*, *Horizontal Bar* y *Metric*.

Top 10 - Ip destino (última 1 hora)

Create New Visualization » Seleccionamos el modelo de representación » *Pie* » Elegimos la fuente, en nuestro caso "*Search – IP Destino*". En siguiente paso es elegir adecuadamente la configuraciones:

Data » Metrics » Slice Size Count
Data » Buckets » Add »
- *Split Slices » Aggregation » Significant Terms*
- *Field » id.resp_h.keyword*
- *Size » 10*

Options » Pie settings » Donut = Right y *Labels settings » show*
Guardamos la visualización *Save* e indicamos el nombre.

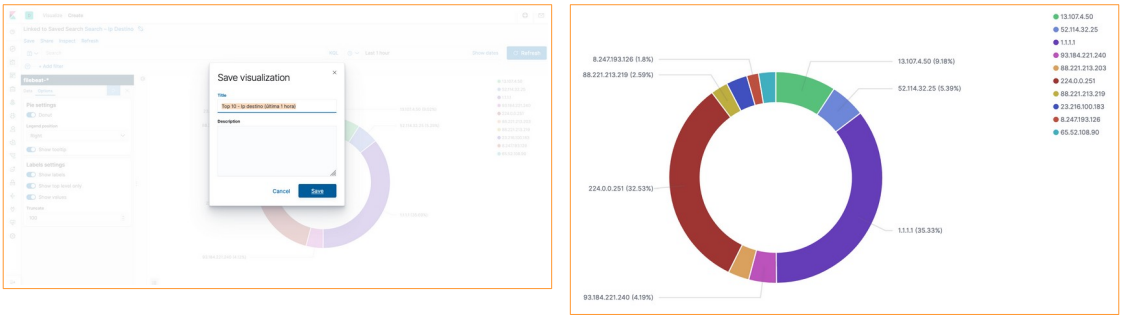

Figura 15: Top 10 - Ip destino (última 1 horas)

Top 10 - Puerto destino (última 1 hora)

Create New Visualization » Seleccionamos el modelo de representación » *Horizontal Bar* » Elegimos la fuente, en nuestro caso "*Search – Puerto Destino*". En siguiente paso es elegir adecuadamente las configuraciones:

Data » Metrics » Y-axis Count
Data » Buckets » Add »
- *Split Slices » Aggregation » Terms*

- *Field » id.resp_p*
- *Order by » Metric: Count*
- *Order » Ascending* y *Size » 10*

Metrics & Axes

- *Metrics » Count » Bar, normal, LeftAxis-1*
- *Y-Axes » LeftAxis-1 » bottom, normal, linear*
- *X-Axes » Possition » left*
- *Panel settings » left*

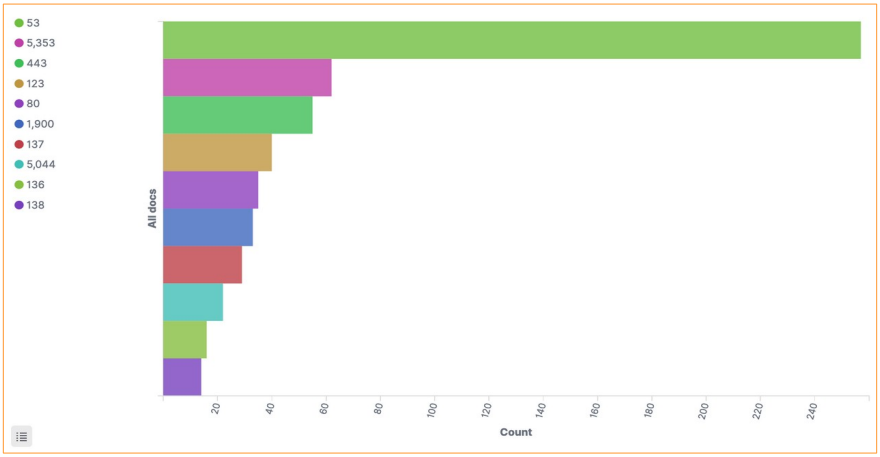

Figura 16: Top 10 - Puerto destino (última 1 hora)

Top 10 - Servicios (última 1 hora 30)

Create New Visualization » Seleccionamos el modelo de representación » *Metric* » Elegimos la fuente, en nuestro caso "*Search – Servicios*". En siguiente paso es elegir adecuadamente las configuraciones:

Data » Metrics » Count
Data » Buckets » Add »

- *Split group » Aggregation » Terms*
- *Field » service.keyword*
- *Order by » Metric: Count*
- *Order » Descending* y *Size » 10*

Figura 17: Top 10 - Servicios (última 1 hora)

Dashboards Zeek

Para la creación del panel de control solo es necesario identificar que bloque de visualización queremos mostrar y colocarlo en la ubicación deseada.

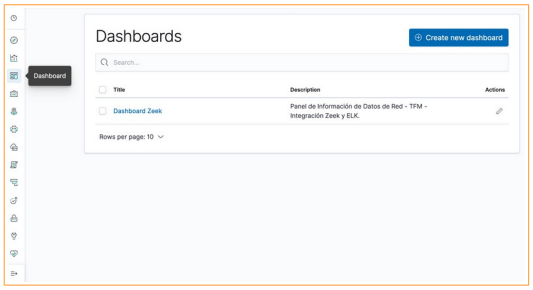

Figura 18: Cuadro de Mandos

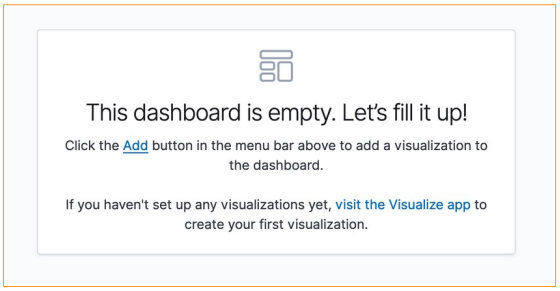

Figura 19: Añadir Visualización

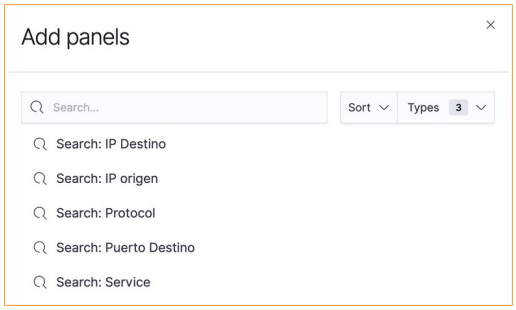

Figura 20: Visualizaciones disponible

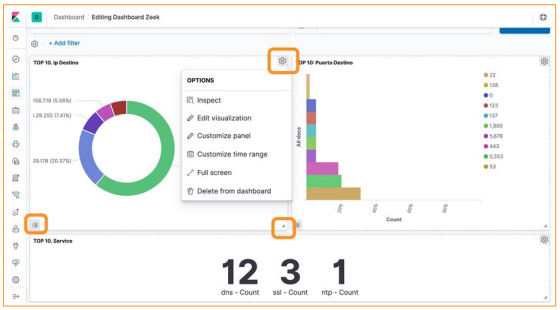

Figura 21: Personalización

A continuación, se muestra el cuadro de mandos con más información sobre el tráfico que transcurre por la red que se está analizando. Estos solo han sido algunas representaciones para el panel de visualización de datos de Zeek y será necesario valorar los datos que son relevantes para la organización para poder representarlos.

Figura 22: Cuadro de Mandos

5.2 Comprobación

Con los pasos realizados anteriormente se obtiene un panel de control sencillo y claro. Este panel es altamente personalizable a nivel de representación de datos. Como es posible ajustar los datos de tiempo se puede obtener datos en una franja temporal determinada, con lo que pueda ayudar a la identificación de actividades en la red.

5.3 Descripción

En este capítulo se ha descrito los pasos seguidos para crear un panel de control sencillo para la representación de los datos en Kibana. Al tener de Zeek en formato JSON es posible mostrar con mayor granularidad todos los campos que proporciona Zeek. Por tanto, es posible representar los datos que se consideran más apropiados siguiente los mismos pasos que se describen.

6. Análisis de las listas de reputación

6.1 Estudio de listas de reputación

Las listas de reputación son listas que hacen referencia a la calidad de una ip o nombre de dominio, es decir, son listas de direcciones ip las cuales han sido utilizadas en alguna actividad sospechosas.

En internet hay una gran cantidad de bases de datos de ip y nombres de dominio que están vinculadas a actividades sospechosas. Entendemos por actividad sospecho a cualquier actividad que implique una actividad maliciosa como podría ser paginas web de suplantación (*phising*), de distribución de softwate malicioso, de *bootnets*, *spam*, servidores de Comando y Control (*C&C*), etc …

Muchos de los productos comerciales de seguridad implementan estos servicios en base a las listas de reputación, ya sea para advertir o bloquear las ips a las que acceden sus clientes / usuarios.

A continuación, se muestran una serie de urls donde podemos encontrar listas abiertas para la consulta de reputación de una ip determina. Estas son solo algunas de las listas. Es importante indicar que también existes listas por categorías que pueden servir para identificar ciertos servicios: concretos:

- **Talos**: https://talosintelligence.com/documents/ip-blacklist
- **FireHOL IP Lists**: https://iplists.firehol.org/
- **Apility.io**: https://apility.io/
- **malc0de**: https://malc0de.com/bl/IP_Blacklist.txt
- **ForoAbuse:** http://www.abuses.es/eswl/
- **CriticalStack Intel Marketplace**: https://intelstack.com/
- **Symantec**: https://ipremoval.sms.symantec.com/
- **Mxtoolbox:** https://mxtoolbox.com/blacklists.aspx
- **Apivoid**: https://www.apivoid.com/api/ip-reputation/

Por tanto, en internet podremos encontrar tanto, listados de ips para tener nuestra propia lista negra de ip sospechosas, como una lista blanca de ip validadas, como servicios web de consulta la conocer la integridad de determinadas ip.

Es importante aclarar que estas listas están en continuo cambio. Algunas de estas listas son administradas por usuarios preocupados por la seguridad y otras están incluidas en sus paquetes de software de detección de amenazas. Se recomienda que antes de utilizar cualquier lista publicada en internet se realice un análisis de la integridad de los datos y la reputación de la fuente consultada.

7. Consulta e integración de listas de reputación

7.1 Investigación

Para este apartado nos vamos a centrar en las listas de reputación que se obtienen a través de fuentes libres, es decir, sin restricción de uso o licencia para obtener la información como suele ser frecuente con la adquisición de productos comerciales.

Para este Trabajo se focaliza el esfuerzo de desplegar una herramienta de automatización que permita realizar la consulta de varias fuentes externas para posteriormente integrarla en el sistema que se está desplegando. En concreto nos referimos a la herramienta IntelMQ

IntelMQ

Es una aplicación de automatización para la recolección y procesado de fuentes de información externa. Se trata de una iniciativa de la Agencia de la Unión Europea de Ciberseguridad (ENISA)[19]. Esta herramienta ha sido desarrollado a través de varias agencias como son, CERT.BE, CERT-EU, CERT.AT, CNCS y ENISA y puesta a disposición de la comunidad de usuario de internet. Esta pensada para facilitar el trabajo de los equipos de respuesta ante incidentes (CERTs[20], CSIRTs[21], SOCs[22], etc). Esta herramienta recolecta información de múltiples fuentes, como pueden ser logs, mensajes, url, textos, etc para que mediante varios mecanismos de tratamientos puedan enriquecer la información para posteriormente gestionarla.

Figura 23: Logotipo IntelMQ

La herramienta se puede descargar de la url de repositorio de Github:

- *https://github.com/certtools/intelmq*

Esta herramienta tiene una serie de ventajas notables como son:

- Reduce la complejidad de administración para la captura de información.
- Reduce la complejidad de desarrollo de procesos de automatización para la captura de datos de información de fuentes externas.
- Utiliza formato JSON para los mensajes, por lo que facilita su gestión.
- Facilita la integración con herramienta de tratamiento de logs como pueden ser ELK, Splunk, Graylog, bases de datos, como pude ser postgreSQL, etc.

19 European Union Agency for Cybersecurity: https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/incident-handling-automation
20 CERT: Equipo de Respuesta a Emergencias Informáticas (Computer Emergency Response Team)
21 CSIRT: Equipo de Respuesta de Seguridad Informática (Computer Security Incident Response Team,)
22 SOC: Centro Operaciones de Seguridad (Security Operation Center)

- Permite la personalización de módulos para la gestión de listas negras.

La herramienta está compuesta por varios módulos de los cuales solo mencionaremos los más relevantes como son:

- IntelMQ Core: Es el core de la herramienta. A través de un *bot*[23] o red de *bots* se configuran los diferentes componentes internos que realizan las tareas de ingesta (*collector*), procesado de datos (*parser*), transformación de datos (*expert*) y extracción del dato (*output*).

  Documentación sobre los bots:

  - *https://github.com/certtools/intelmq/blob/develop/docs/Bots.md*

  Documentación sobre las fuentes de información disponibles

  - *https://github.com/certtools/intelmq/blob/develop/docs/Feeds.md*
- IntelMQ Manager: Se trata de la herramienta para la gestión del core, la cual mediante una interface gráfica se realizan las tareas de administración de IntelMQ

7.2 Análisis de los mecanismos de consulta

Una vez configurada la herramienta intelMQ nos ofrece un fichero con información de ip de reputación. Estos datos son de vital importancia para tratar de identificar actividad sospechosa de la red.

Intelmq - "*Incident Handling Automation*"

Instalación Intelmq

A continuación, se identifican los pasos seguidos para la puesta en marcha de la herramienta intelMQ

Se utiliza como guía para la instalación de la herramienta IntelMQ la información que se proporciona desde el repositorio oficial de IntelMQ (url: *https://github.com/certtools/intelmq*) para desplegar en el entorno de virtualización:

- **Sistema Operativo**: Linux 5.3.0-24-generic Ubuntu x86_64 GNU/Linux
- **Información disco duro**: 60 GB
- **Información RAM**: 4 GB

- **Hostname**: vm07pec4intel
- **Dirección MAC**: 00:0c:29:31:e7:c9
- **Dirección IP**: 192.168.29.88
- **Núcleos CPU**: 2
- **Memoria RAM**: 4GB

Se siguen las instrucciones indicadas en la url que se indica a continuación donde los comandos que aplicamos son los siguientes:

- *https://github.com/certtools/intelmq/blob/develop/docs/INSTALL.md*

23 Bot: herramienta software para la automatización de tareas repetitivas.

```
usuario@vm07pec4intel:~$ sudo apt install python3-pip python3-dnspython python3-psutil
python3-redis python3-requests python3-termstyle python3-tz python3-dateutil
usuario@vm07pec4intel:~$ sudo apt install redis-server
usuario@vm07pec4intel:~$ sudo apt install bash-completion jq
usuario@vm07pec4intel:~$  sudo  apt  install  python3-sleekxmpp  python3-pymongo  python3-
psycopg2
root@vm07pec4intel:~# pip3 install intelmq
root@vm07pec4intel:~# useradd -d /opt/intelmq -U -s /bin/bash intelmq
root@vm07pec4intel:~# sudo pip3 install beautifulsoup4
root@vm07pec4intel:~# sudo pip3 install bs4
root@vm07pec4intel:~# sudo intelmqsetup
```

Se prosigue con las instrucciones indicas del manual de instalación:

- *https://github.com/certtools/intelmq/blob/develop/docs/User-Guide.md*

```
root@vm07pec4intel:~# systemctl enable redis.service
root@vm07pec4intel:~# systemctl start redis.service
```

Se verifica que se ha completado la instalación correctamente en el directorio esperado `/opt/intelmq/` y se comprueba el estado de los bots, donde observamos que la *botnets* se encuentran en estado `stop`. Se comprueba de igual manera los logs de IntelMQ.

```
root@vm07pec4intel:~# ls -la /opt/intelmq/etc/
root@vm07pec4intel:~#  intelmqctl status
root@vm07pec4intel:~# tail -f /opt/intelmq/var/log/*.log
```

Instalación Intelmq Manager

El siguiente paso es instalar **IntelMQ Manager** (V2.1.0). Se trata de la interfaz gráfica de **IntelMQ** (V2.1.1)

Se siguen las instrucciones del manual de instalación indicado en la url:

- *https://github.com/certtools/intelmq-manager/blob/master/docs/INSTALL.md*

```
root@vm07pec4intel:~# apt-get install git apache2 php libapache2-mod-php7.2
root@vm07pec4intel:~# git clone https://github.com/certtools/intelmq-manager.git
/tmp/intelmq-manager
root@vm07pec4intel:~# cp -R /tmp/intelmq-manager/intelmq-manager/* /var/www/html/
root@vm07pec4intel:~# chown -R www-data.www-data /var/www/html/
root@vm07pec4intel:~# usermod -a -G intelmq www-data
root@vm07pec4intel:~# mkdir /opt/intelmq/etc/manager/
root@vm07pec4intel:~# touch /opt/intelmq/etc/manager/positions.conf
root@vm07pec4intel:~# chgrp www-data /opt/intelmq/etc/*.conf
/opt/intelmq/etc/manager/positions.conf
root@vm07pec4intel:~# chmod g+w /opt/intelmq/etc/*.conf
/opt/intelmq/etc/manager/positions.conf
root@vm07pec4intel:~# sudo -u intelmq /usr/local/bin/intelmqctl
root@vm07pec4intel:~# apt install python-pip
root@vm07pec4intel:~# pip install beautifulsoup4
```

El próximo paso es asignar los permisos añadiendo la siguiente entrada en el directorio `/etc/sudoers`

`www-data ALL=(intelmq) NOPASSWD: /usr/local/bin/intelmqctl`

```
root@vm07pec4intel:~# nano /etc/sudoers
```

```
root@vm07pec4intel:~# sudo chown intelmq.intelmq /opt/intelmq -R && sudo chmod u+rw
/opt/intelmq -R
```

En el anexo "*ANEXO II Instalación IntelMQ*" se representan todos los comandos y las salidas de los mismos para tener una orientación de los pasos seguidos en el proceso de instalación del conjunto de aplicaciones **IntelMQ** y **IntelMQ Mananger**, por lo que se recomienda su consulta.

Llegado a ese punto ya se dispone de la herramienta **IntelMQ** y **IntelMQ Manager** desplegadas. Se puede acceder al portal de administración accediendo mediante *http://ip/index.php* donde se muestra el panel de control de inicio de la aplicación IntelMQ Manager

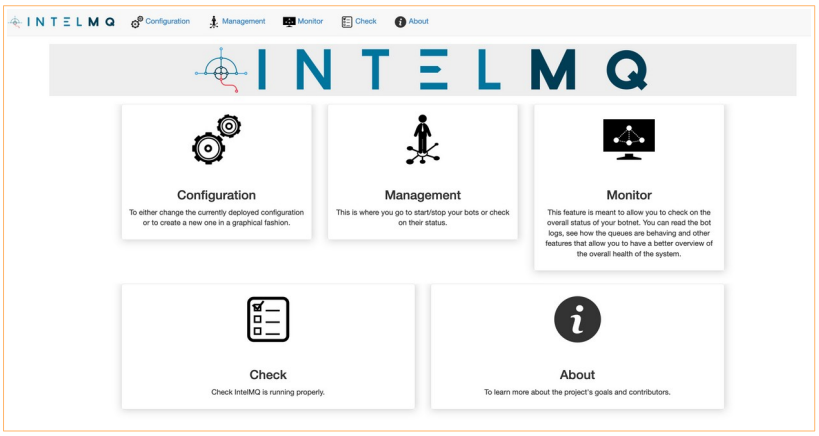

Figura 24: Portal de Inicio IntelMQ Manager

Verificación funcional

Para conocer el estado de la instalación se puede realizar una comprobación mediante comandos (*intelmqctl*) o mediante el portal de administración.

Algunos de los comandos más relevantes son:

```
root@vm07pec4intel:~# intelmqctl status
root@vm07pec4intel:~# intelmqctl start
root@vm07pec4intel:~# intelmqctl stop
root@vm07pec4intel:~# intelmqctl restart
root@vm07pec4intel:~# intelmqctl reload
root@vm07pec4intel:~# intelmqctl list queues-and-status
```

También es importante revisar los logs para verificar que no se producen errores funcionales.

```
root@vm07pec4intel:~# tail -f /opt/intelmq/var/log/*.log
```

Además, es conveniente verificar los procesos que se encuentran activos, como son los procesos *redis* y *intelmq*.

- Redis: *$ sudo lsof -i -P -n | grep redis*
- Intelmq: *$ sudo lsof -i -P -n | grep intelmq*

Nota: `redis` se utiliza como proceso interno para la gestión de la información que trata intelMQ.

Mediante el comando `intelmqctl -h` se muestra la información de ayuda para el manejo de la herramienta por si se requiere mayor detalle de los comandos indicados.

Otro método para revisar el funcionamiento de la aplicación y crear desarrollos nuevos de manera gráfica es mediante el acceso web a **IntelMQ Manager**, donde podemos encontrar varios apartados: configuración, administración, monitorizan y comprobación. Los detalles de cómo utilizar la herramienta a través del portal web se pueden consultar en el "*ANEXO II Instalación IntelMQ*".

7.3 Análisis de los mecanismos de integración

Para añadir los datos que nos proporciona la herramienta intelMQ utilizamos `filebeat`.

A continuación se representa el despliegue realizado:

- Nodo 01: Zeek Framework
- Nodo 02: Stack ELK
- Nodo 03: IntelMQ

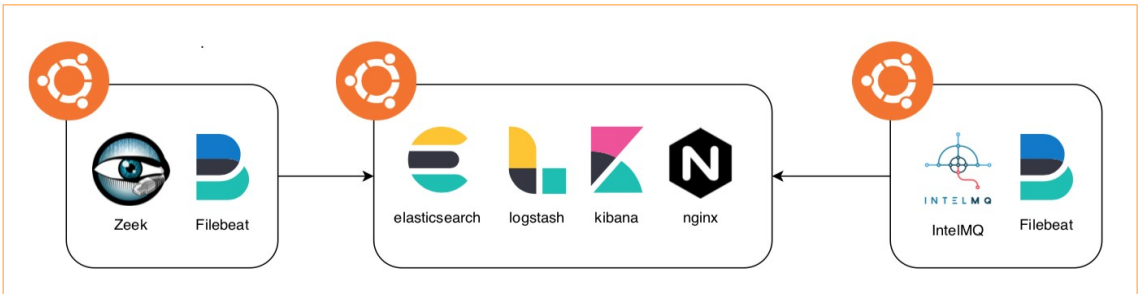

Figura 25: Diagrama general del Sistema

Para esta nueva instalación de filebeat se utiliza la fuente *.DEB* para mostrar otro método de instalación diferente la mostrado en apartados anteriores.

```
usuario@vm07pec4intel:~$ curl -L -O
https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.5.0-
amd64.deb

usuario@vm07pec4intel:~$ sudo dpkg -i filebeat-7.5.0-amd64.deb
```

Una vez instalado se debe realizar los cambios de configuración propios para cada integración, en nuestro caso para IntelMQ, Minemeld y para los Scripts.

A continuación, configuramos Filebeat con nuestras necesidades.

- Ruta: `/etc/filebeat/`
- Fichero: `filebeat.yml`
- Descripción: en este fichero se configura la ruta donde se encuentran los datos que se quieren enviar a nuestra instancia de ELK, y en concreto a Logstash.

```
#========================== Filebeat inputs =============================
filebeat.inputs:
```

```
- type: log

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

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /opt/intelmq/var/lib/bots/file-output/events.txt
    #- /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*


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

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

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

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

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

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

#================================ General =====================================
tags: ["intelmq"]
#================================ Kibana ======================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
#setup.kibana:
  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
  #host: "192.168.29.87:5601"

#================================ Outputs =====================================
# Configure what output to use when sending the data collected by the beat.
#-------------------------- Elasticsearch output ------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
  #hosts: ["localhost:9200"]
#------------------------------ Logstash output -------------------------------
output.logstash:
  # The Logstash hosts
  hosts: ["192.168.29.87:5045"]

#================================ Processors =====================================
# Configure processors to enhance or manipulate events generated by the beat.
processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~
```

Una vez instalado y configurado es necesario validar que se encuentra en marcha mediante los comandos `systemctl status | start | stop | reload | enable filebeat`

```
root@pec3serverintelmq:/home/usuario# lsof -i -P -n | grep filebeat
```

NOTA 1: El proceso de volcado de datos, si todo ha ido bien es muy rápido, por lo que es posible que no se observe ningún proceso en marcha, pero si realizamos la ejecución en un breve espacio de tiempo desde que se inicia por primera vez, observaremos una salida como la que se muestra a continuación.

```
root@vm07pec4intel:/home/usuario# lsof -i -P -n | grep filebeat
filebeat  23635           root    5u  IPv4 538422      0t0  TCP 192.168.29.88:43946->192.168.29.87:5045 (ESTABLISHED)
```

7.4 Visualización de información (cuadro de mandos)

Una vez realiza la configuración anterior es necesario indexar la nueva información relacionada con intelMQ, por lo que deberemos realizar los pasos similares a los ya realizados en el apartado de Zeek. En términos generales los pasos son los siguientes:

- Indexar un nuevo índice
- Crear búsquedas personalizadas
- Crear un panel de control amigable

A continuación, se especifican los pasos a seguir para el indexado de los datos.

Accedemos al panel de "*Kibana* » *Management* » *Index patterns* » *Create index pattern"* para añadir una nuevo indice.

Introducimos el tag que hemos indicado para indexar la información de IntelMQ, en nuestro caso "`filebeat-intelmq-*`" y al introducir el valor nos indica mediante un check de confirmación que ha encontrado el índice (*Success! Your index pattern matches 1 index*) por lo que hacemos clic en "*Next step*".

En el segundo paso tenemos que indicar el filtro con el que queremos que se referencie nuestra información a nivel de tiempo. Para nuestro caso utilizamos "*I don't want to use the Time Filter*", ya que al ser datos estáticos no es necesario conocer la evolución temporal, por lo que creamos nuestro patrón de índice haciendo click en "*Create index pattern*".

Si todos los pasos anteriores han sido correctos, al igual que en los apartados anteriores se procede a crear una serie de búsquedas de datos para posteriormente representarlos en ELK. A continuación, se muestra el panel de control creado.

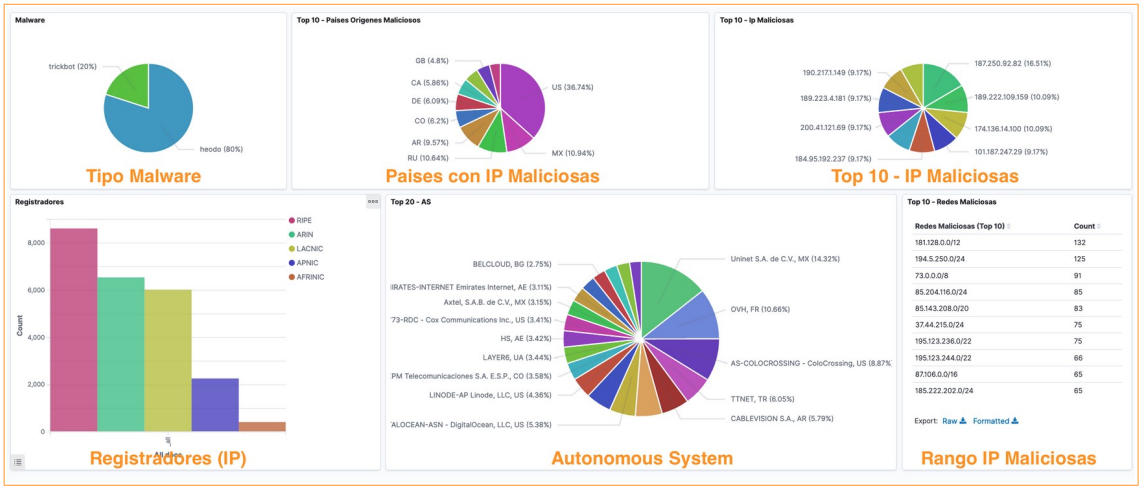

Figura 26: Panel de Información intelMQ

8. Creación de alertas en ELK

ELK es un sistema muy completo a nivel de explotación de datos, pero un sistema como éste, en el que es necesario identificar comportamientos anómalos no es suficiente con representar los datos, por lo que generar alertas en base a eventos determinados ayuda a estar informado en todo momento. Es por este motivo por el cual una correcta implementación es muy importante para la identificación temprana de amenazas identificadas por red.

8.1 Definición de alertas

Se define una alarma como el resultado de la ejecución de eventos en un espacio de tiempo. El objetivo de una alerta en el sistema desplegado (Zeek, IntelMQ y ELK) es notificar cuando se produzca una coincidencia determinada, es decir:

- Analizar el tráfico de red en tiempo real
- Comparar el tráfico de red con una lista de ip determinada
- Realizar una o varias acciones en caso de coincidencias

Por tanto, sería deseable que nuestro sistema de alerta y notificación pudiera ser capaz avisar cuando se produzca una coincidencia previamente programada. Con ello se pretende:

1. Enviar un correo electrónico
2. Crear una entrada en el log de nuestro elasticsearch

El escenario, a grandes rasgos, el algo similar a lo que se muestra en la siguiente imagen que se observa a continuación.

Figura 27: Diagrama para la detección de amenazas

8.2 Modelo de alerta en ELK

El Stack ELK utiliza un plugin llamado "*Watcher*" para generar las alertas a través de eventos. Este plugin requiere licencia como se indica en la documentación del producto.

- https://www.elastic.co/guide/en/watcher/current/installing-watcher.html
- https://www.elastic.co/es/downloads/watcher

Se utiliza la licencia de demostración la cual tiene una duración de 30 días.

> **Nota**: durante la realización del Trabajo se ha solicitado a través del portal de soporte (https://www.elastic.co/) información sobre el precio de licenciamiento, pero no he obtenido respuesta.

La activación se realiza desde el portal de ELK. A continuación, se muestran los pasos seguidos desde el portal de administración mediante 3 imágenes.

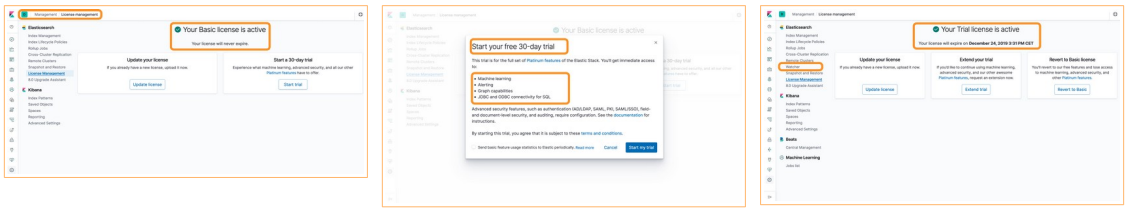

Figura 28: Panel de activación licencias demostración

Al activar la licencias se activan nuevas funcionalidades entre las que se encuentra "*watcher*". Watcher se encuentra en el apartado de Elasticsearch.

Watcher permite la creación de alarmas simples mediante su wizard o avanzadas en formato JSON mediante una consola web.

Para entrar en mayor detalle sobre el funcionamiento de las alertas en ELK debemos conocer que elementos fundamentales son necesarios para la creación de una alerta. Una alerta se compone de *inputs*, *triggers*, *conditions* i *acctions* además de *transformers*.

- *Inputs*: se trata del tipo de entrada que queremos ejecutar. Admite tres tipo; simple (*simple*), búsqueda (*search*), http (*http*) y cadena (*chain*)
- *Triggers*: se trata del disparados, es decir, cuando queremos que se ejecute nuestro tarea.
- *Conditions*: se trata de la condición por la cual queremos que se produzca un evento o alerta determinado.
- *Actions*: se trata de la acción que se realiza si se cumple la condición.

Para más detalles sobre su funcionamiento y sus características es recomendable realizar una revisión de la documentación oficial donde se muestran varios ejemplos de apoyo que ayudan a la comprensión de cada uno de los elementos.

- https://www.elastic.co/guide/en/watcher/current/reference.html

La idea general para la creación de una alerta básicamente sigue el modelo que se representa a continuación en una imagen, donde cada cierto tiempo (*trigger*) se ejecuta el código de comprobación (comparación de ip destino de una comunicación proporcionada por Zeek y la ip de la base de datos de IntelMQ). En el caso de no producirse una coincidencia no se realiza ninguna otra acción hasta que se vuelva a ejecutar. Por el contrario, si al producirse una coincidencia de ambas ips, el sistema generará una o varias acciones definidas por el usuario. Estas acciones pueden ser el envío de correo electrónico, indexado de datos, insertar datos en log, notificación en slack, webhock, jira, etc ...

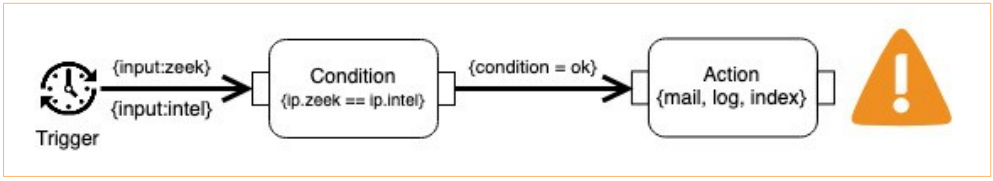

Figura 29: Alerta en ELK

Para la puesta en servicio de watcher se debe tener en cuenta el siguiente flujo, el cual es el proceso que siguen los datos generados por Zeek hasta llegar a Elasticsearch.

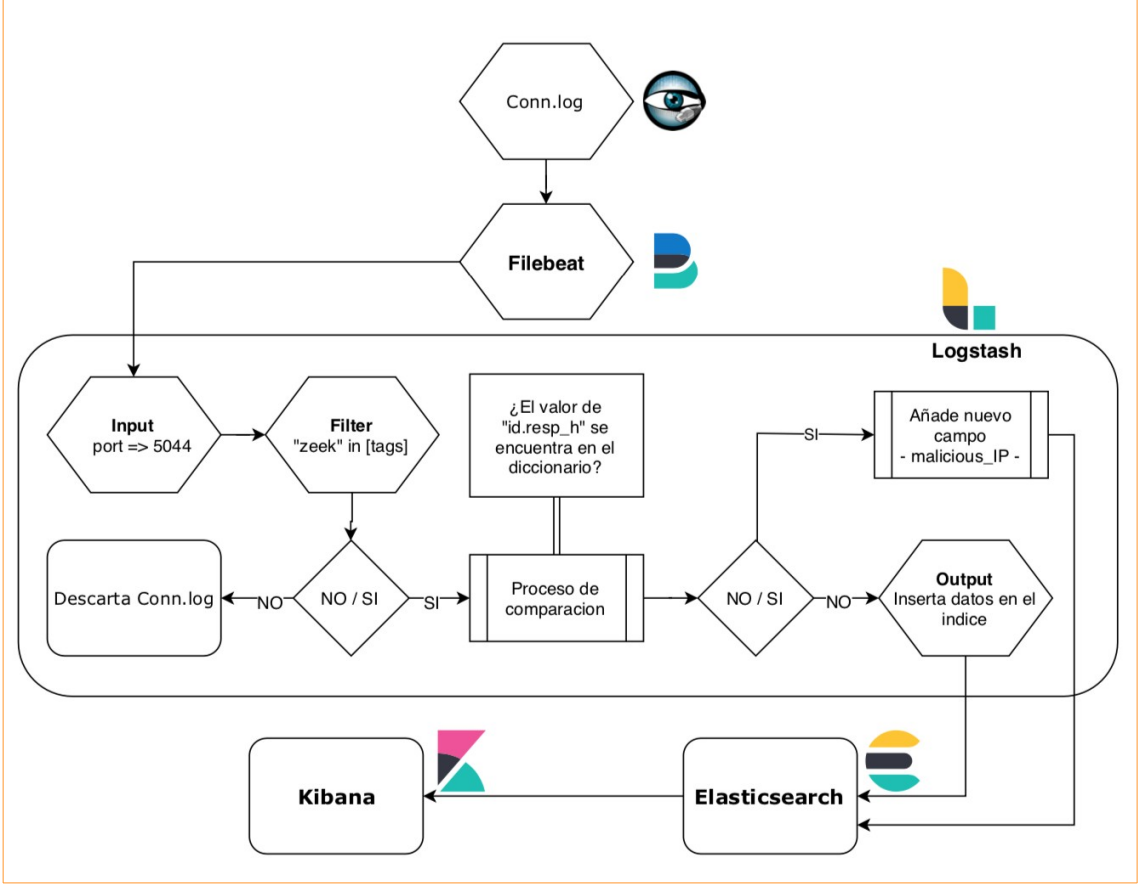

Figura 30: Flujo Inserción de Datos

8.3 Requisitos para el despliegue de alertas

Para la puesta en servicio de alertas en watcher, previamente es necesario realizar algunas nuevas configuraciones.

Envío de correo electrónico: Para el envío de correo electrónicos a través de ELK se necesario realizar ajustes en nuestro sistema. En concreto los parámetros de seguridad y de correo que se indican en la documentación del fabricante:

- Ruta: `/etc/elasticsearch/`
- Fichero: `elasticsearch.yml`

```
action.destructive_requires_name: true
xpack.security.enabled : false
```

```
xpack.watcher.enabled : true

xpack.notification.email.account:
    gmail_account:
        profile: gmail
        smtp:
            auth: true
            starttls.enable: true
            host: smtp.gmail.com
            port: 587
            user: not.watcher.mail@gmail.com
```

Utilizamos una cuenta de correo del servicio de Google por ser un recurso gratuito y fácil de implementar.

- Cuenta de correo: `not.watcher.mail@gmail.com`

Se debe aplicar el siguiente comando para poder utilizar el cifrado de la contraseña para el servicio de correo para gmail.

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

Una vez aplicado, es necesario realizar el reinicio de los servicios de elasticsearch y revisar los logs por si hubiera algún error para analizar.

Envío de log: No es necesario realizar ninguna configuración para este apartado, ya que mediante la acción configurada se crea una entrada en el log de elasticsearch.

Array de ip maliciosas: Para poder implementar la comparación del tráfico de red y la lista de ip maliciosas procedentes de la herramienta IntelMQ es necesario realizar una transformación previa de la información. Esta transformación consiste en extraer la ip y crear un fichero en formato CSV. A continuación, se detallan las configuraciones para implementar esta necesidad:

- **Origen de datos**: La herramienta IntelMQ genera un fichero en formato JSON con información enriquecida. Esta información es indexada por parte de ELK a través de filbeat.
  - Ruta: `/opt/intelmq/var/lib/bots/file-output/`
  - Fichero: `events.txt`
- **Transformación de datos**: Es necesaria la información IP de cada una de las entradas existente en el fichero interior, en concreto del valor `source.ip` del fichero `events.txt`. Para obtener esta información se crea el siguiente código en python para que automáticamente se realice la conversión y genere un nuevo fichero añadiendo un texto descriptivo.
  - Host: `vm07pec4intel`
  - Ruta ejecutable: `/home/usuario/json2csvrun`
  - Ejecutable: `json2csv.py`
  - Ruta fichero: `/home/usuario/exchangefiles/`
  - Fichero: `events.csv`

  A continuación, se muestra el código del ejecutable.

```
import json
file_json = '/opt/intelmq/var/lib/bots/file-output/events.txt'
file_csv = '/home/usuario/exchangefiles/events.csv'

with open(file_json, "r") as f:
    lines = f.readlines()

data_list = (json.loads(line) for line in lines)
with open(file_csv, "w") as f:
    for data in data_list:
        if 'source.ip' in data:
            f.write(f'{data["source.ip"]},malicious_IP\n')
```

- **Transferencia de datos**: Una vez tenemos el fichero `events.csv` en la máquina donde tenemos desplegado la herramienta IntelMQ es momento de transferir el fichero a la máquina del Stak de ELK. Para ello, utilizaremos la aplicación SCP y el cifrado de clave pública y privada para que pueda realizarse de manera automática mediante certificado.

- **Rutina de ejecución**: para que la información de las listas de reputación estes actualizadas, creamos un cron para que cada hora se actualice el fichero CSV con la información que pueda aportar IntelMQ. A continuación, se muestra la información sobre los job generados.

```
# m h  dom mon dow   command
11 * * * * python3 /home/usuario/json2csvrun/json2csv.py
12 * * * * scp /home/usuario/exchangefiles/* intelip@192.168.29.87:/home/transferencia/
```

  - Ejecución del script de conversión (JSON a CSV): se ejecuta todos los días del año en el minuto 11, creando un nuevo fichero (`events.csv`)
  - Ejecución del script de transferencia entre máquina (origen `vm07pec4intel`, destino `vm07pec4elk`): se ejecuta todos los días del año a todas horas en el minuto 12. Realiza la transferencia desde el nodo de IntelMQ al nodo de ELK

A continuación, se muestra mediante una imagen el proceso y los elementos que participan y que representa claramente como se ha desplegado el sistema.

- Host Zeek: Zeek genera log y Filebeat los envía al host ELK
- Host IntelMQ: Obtiene datos de reputación que almacena en un fichero. Los datos son enviados al host ELK a través de Filebeat. Al mismo tiempo, en este host se configura un script para convertir el fichero obtenido de IntelMQ en otro fichero son la información de Ips. Este nuevo fichero es enviado al host ELK y almacenado en un directorio.
- Host ELK: ELK recibe los datos de Zeek por el puerto (5044) y los procesa. Por otro lado, recibe los datos de IntelMQ por el puerto (5055). Además, recibe un en fichero un listado de ip maliciosos mediante una tarea programada.

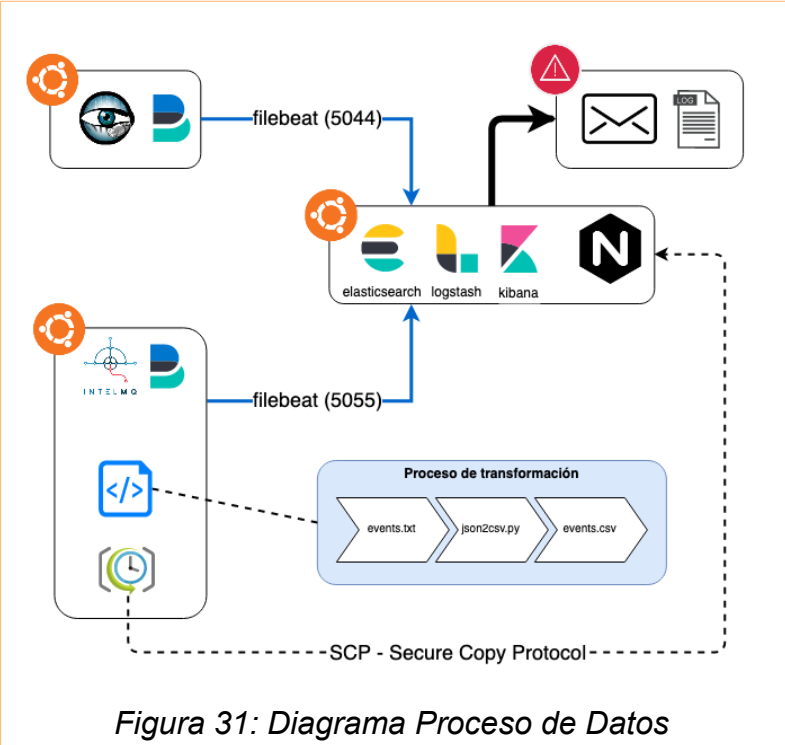

Figura 31: Diagrama Proceso de Datos

En la siguiente imagen se puede ver como se generan los ficheros que intervienen en paso de transferencia entre el host IntelMQ y el host ELK.

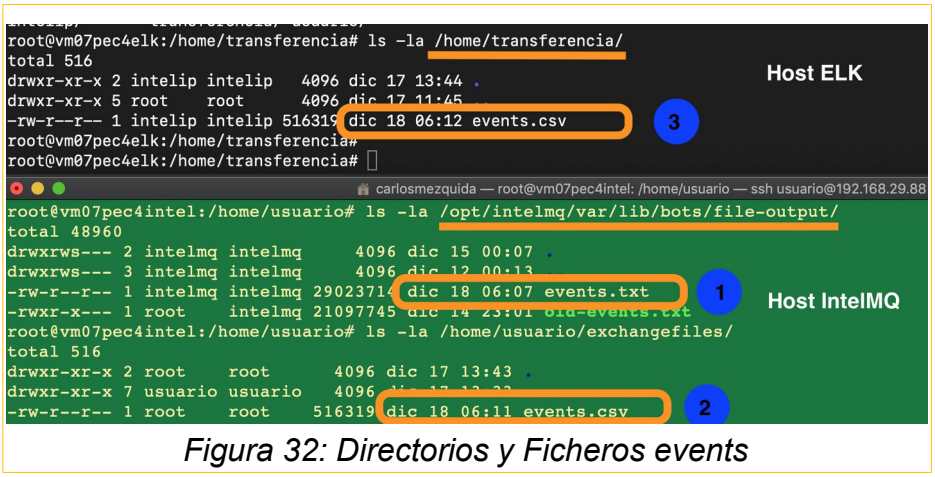

Figura 32: Directorios y Ficheros events

Este es el resumen de directorios a tener en cuenta para la transformación y transferencia de ficheros:

- Host: `vm07pec4intel` - IntelMQ
- Ruta: `/opt/intelmq/var/lib/bots/file-output/`
- Fichero: `events.txt`

- Host: `vm07pec4intel` - IntelMQ
- Ruta: `/home/usuario/exchangefiles/`
- Fichero: `events.csv`

- Host: `vm07pec4elk` - ELK

- Ruta: `/home/transferencia/`
- Fichero: `events.csv`

A continuación se adjunta una muestra del contenido del fichero `events.csv`

```
root@vm07pec4elk:/home/usuario# tail /home/transferencia/events.csv
91.242.138.5,malicious_IP
148.244.114.49,malicious_IP
148.69.94.166,malicious_IP
88.80.195.221,malicious_IP
```

8.4 Elección del modelo de alerta

Al realizar el análisis de varios modelos para el proceso de comparación y gestión de eventos se identifican varios modelos:

- Comparación de datos en Zeek
- Comparación de datos en ELK
- Comparación de datos en Watcher

Para el Trabajo se utiliza el modelo de comparación de datos en ELK, por lo que se realizar una serie de cambios sobre los ficheros de Logstash para poder implementar esta nueva funcionalidad de comparación.

8.5 Creación de Alerta

A continuación explican las modificaciones necesarias para la creación de una alerta.

Configuraciones de Logstash

- Configuración de Logstash para el tratamiento de los datos:
  - Ruta del fichero: `/etc/logstash/conf.d/`
  - Fichero: `11-filter-beats.conf`

```
filter {
    if "zeek" in [tags] {
         translate {
          field => "[id.resp_h]"
          destination => "malicious_IP"
          dictionary_path => "/home/transferencia/events.csv"
          override => true
         }
    }
 }
```

Como se observa en la configuración de fichero `11-filter-beats.conf` el campo relevante elegido para la comparación es `id.resp_h`, el cual representa la ip de destino en una comunicación donde el origen es cualquier equipo de la red interna.

Creando un nuevo bloque `translate` para comparar el valor de `id.resp_h` procedente `conn.log` con el diccionario de ip del fichero events.csv ubicado en el directorio `home/transferencia/`

- Configuración de Logstash para el indexado de los datos.

- Ruta del fichero: `/etc/logstash/conf.d/`
- Fichero: `21-elasticsearch-output.conf`

```
output {
  if "zeek" in [tags] {
    elasticsearch {
      hosts => ["http://localhost:9200"]
      index => "filebeat-zeek-%{+YYYY.MM.dd}"
      }
      stdout {
      codec => rubydebug
      }
   }
 }
```

- Es necesario actualizar el índices en Kibana para que muestre el nuevo campo cuando se produce una coincidencia.

Configuraciones en Watcher

En este apartado se explica el código fuente que se ha utilizado para generar la alarma:

- **Trigger**: Cada 30 segundos, para nuestro modelo de red es suficiente, pero debe tenerse en cuenta en entorno de mucho tráfico.

```
  "trigger": {
    "schedule": {
      "interval": "30s"
    }
  },
```

- **Input**: se realiza una búsqueda sobre el índice `filebeat*` donde se espera encontrar una coincidencia con el string `malicious_IP`, el cual es el texto que hemos añadido a nuestra entrada cuando se produce una coincidencia con la lista de ips maliciosas. El rango que tiempo que analiza en nuestro caso son los 5 minutos desde que se detecta la primera coincidencia.

```
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "filebeat*"
        ],
        "rest_total_hits_as_int": true,
        "body": {
          "query": {
            "bool": {
              "must": [
                {
                  "query_string": {
                    "query": "malicious_IP"
                  }
                },
                {
                  "range": {
                    "@timestamp": {
                      "gte": "now-5m"
                    }
                  }
```

```
                }
              ]
            }
          },
          "_source": [
            "message"
          ],
          "sort": [
            {
              "@timestamp": {
                "order": "desc"
              }
            }
          ]
        }
      }
    }
  },
```

- **Condition**: En caso de producirse una coincidencia con el texto malicious_IP y éste coincida con el número indicar por nosotros (0) pasará al proceso de `acction`.

```
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
```

Para el apartado de acciones el stack de ELK permite varias opciones como se puede ver en propia documentación. Nos permite una entrada de log, envío de correo electrónico, entrada en un índice, integración con aplicaciones de terceros (jira, webhok, etc …). Para nuestro Trabajo utilizamos Log y Email como medidas de acción, donde resaltaremos mediante una texto el evento.

- **Acción**: Log

```
  "actions": {
    "log": {
      "logging": {
        "level": "info",
               "text": "ALERTA. Identificada Acceso IP Maliciosa - Watch
[{{ctx.metadata.name}}] has exceeded the threshold {{ctx.payload.hits.total}}"
      }
    },
```

- **Acción**: Notificación Correo electrónico a la cuenta de correo creada para este trabajo `not.watcher.mail@gmail.com`

```
   "email_1": {
      "email": {
        "profile": "standard",
        "to": [
          "not.watcher.mail@gmail.com"
        ],
                  "subject": "ALERTA - Acceso a IP Maliciosa - Watch
[{{ctx.metadata.name}}]       has       exceeded       the       threshold
{{ctx.payload.hits.total}}",
        "body": {
            "text": "Hola, \nSe ha detectado el acceso a una IP identificada
como maliciosa desde la Red de la Organización. There are >>
{{ctx.payload.hits.total}} << documents in your index. Threshold is 0.\
nSaludos."
        }}
```

Se recomienda la consulta del anexo referente a los ficheros de configuración para un mejor entendimiento de los datos configurados: *ANEXO III. Ficheros de configuración*

Para realizar las comprobaciones y optimización de nuestro watcher utilizo la opción que proporciona ELK llamada (Dev Tool). Esta herramienta es muy útil para ir ajustando nuestros scripts.

- **Crear watcher**
  `PUT _watcher/watch/nombre-del-watcher`
- **Ejecutar watcher**
  `POST _watcher/watch/nombre-del-watcher/_execute`
- **Borrar watcher**
  `DELETE _watcher/watch/nombre-del-watcher`

8.6 Comprobación funcional de la alerta

Una vez realiza la configuración necesaria es momento de comprobar que se obtienen los resultados esperados, es decir:

- Añadir un nuevo campo si la ip de destino coincide con una ip de la lista de reputación.
- Enviar un correo electrónico de notificación
- Añadir una entrada en el log de elasticsearch

Para realizar la comprobación generamos tráfico sospechoso de manera intencionada con el objetivo de que sea almacenado por Zeek en los logs, transferido a ELK para que Logstash lo trate y añada un nuevo campo (`malicious_IP`), lo cual desencadenará dos acciones; el envío de un correo electrónico y la entrada en los logs de Elasticsearch.

Comprobación de nuevo campo en el índice

Figura 33: Panel Kibana y Búsqueda Malicious_IP

Comprobación de entradas en el log:

```
root@vm07pec4elk:/home/transferencia# tail -n 100 /var/log/elasticsearch/ClusterElkZeekService.log | grep ALERTA
[2019-12-18T00:11:42,957][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 6
[2019-12-18T00:12:12,966][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:12:42,971][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:13:12,990][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:13:42,994][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:14:13,005][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:14:43,018][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:15:13,023][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:15:43,031][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 8
[2019-12-18T00:16:13,046][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 7
[2019-12-18T00:16:43,047][INFO ][o.e.x.w.a.l.ExecutableLoggingAction] [vm07pec4elk] ALERTA. [Alarma IP MALICIOSA detectada ] Identificada Acceso IP Maliciosa. Se ha excedido el umbral: 2
```

Figura 34: Entrada LOG de Alarma

Comprobación de entradas de correos electrónicos:

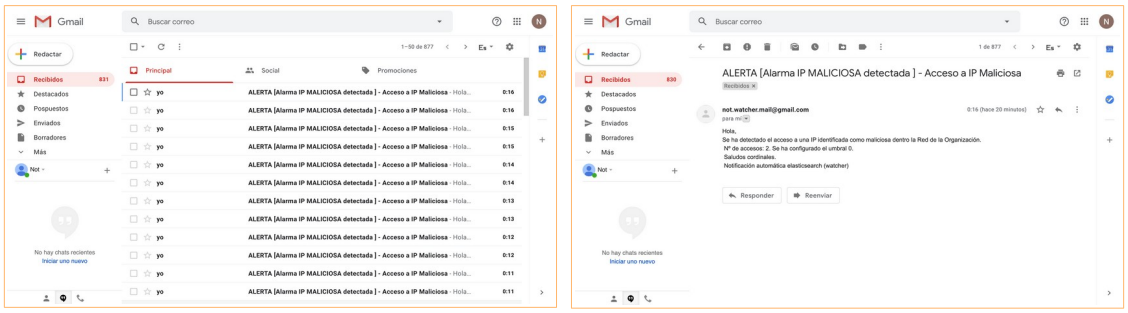

Figura 35: Notificación Correos Electrónicos

La configuración del watcher que se ha representado en los apartados anteriores es simple, pero es totalmente personalizable a las necesidades de los administradores que vayan a implementar este servicio. Realizando ajustes en la programación del watcher o generando unas nuevas, es posible generar alarmas de más complejas y específicas. Las posibilidades son muy grandes gracias al uso del formato JSON que implementa watcher.

9. Representación conjunta de datos Zeek y listas de reputación en Kibana

9.1 Descripción

El objetivo de este apartado es crear un panel donde se representa información procedente de Zeek y de IntelMQ para tener una visión conjunta, de modo se sea posible tener información en tiempo real de la red de la organización y al mismo tiempo saber si el tráfico que se está generando en la red es sospecho o tiene alguna relación con actividades maliciosa.

9.2 Implementación

Teniendo en cuenta que es el tercer panel que se configura no es necesario explicar como configurar el panel.

La información que nos puede interesar observar es la que se indica a continuación por lo que hemos creado varias búsquedas y posteriormente las hemos representado en un panel especifico. La información que se considera relevante para mostrar es la siguiente:

- Las coincidencias que se producen de ips maliciosas en la red (`malicious_IP`)
- Las ips de destino maliciosas que se consultan (`id.orig_h.keyword` y `malicious_IP`)
- Las ips de la red interna que contactan con ips maliciosas (`id.res_h.keyword` y `malicious_IP`)

A continuación, se muestra el panel de control con información de la actividad sospechosa identificada en la red. Este es un ejemplo sencillo y claro que pretende ayudar a los administradores de este sistema a estar informados sobre actividades sospechosas.

Figura 36: Panel de Control Tráfico Malicioso

10. Conclusiones

En este apartado se identifican las conclusiones del desarrollo del Trabajo, como es el cumplimiento de los hitos marcados, los problemas encontrados y trabajos futuros.

10.1 Desarrollo del trabajo

El objetivo general del presente trabajo ha consistido en la integración de la herramienta Zeek y ELK con listas de reputación procedentes de una herramienta de gestión centralizada (intelMQ) de modo que mediante una serie de alarmas los administradores de seguridad puedan tener información en tiempo real sobre actividades sospechosas en las redes de una organización. Esta información real se plasma, por un lado, mediante notificaciones de correos electrónicos, entrada en el log y con la creación de unos paneles de control, donde adquiere especial relevancia el panel especifico donde se integra la información procedente de la red obtenida por Zeek y la información de inteligencia proporcionada por intelMQ en un panel de control de Kibana.

En cuanto los objetivos específicos se han completado los objetivos de investigación, de implantación y de entrega indicados en la Plan de Trabajo.

En resumen, se puede afirmar que se ha completado con éxito el objetivo de investigación, ya que se ha logrado entender el funcionamiento de las herramientas necesarias para para el uso y explotación de Zeek y ELK junto con otr herramienta muy interesante que ha sido intelMQ.

En cuanto a los objetivos de implantación se puede decir que se ha conseguido mediante el despliegue de tres máquinas virtuales donde en cada una ellas se ha instalado las herramientas indicadas anteriormente. Prueba de ello son los anexos de instalación y las diferentes pantallas mostradas durante el desarrollo del trabajo.

Por otro lado, en cuanto al objetivo se entrega, a pesar de que cumplirse en tiempo la entrega de cada documento, los hitos/tiempos establecidos no fueron dimensionados correctamente para la PEC3, pero se ha compensado con la entrega de la PEC4.

10.2 Análisis de riesgos

Durante la realización del trabajo se han identificados varios riesgos que han hecho peligrar el evolución y cumplimiento de los hitos donde se destaca:

- El tiempo invertido en el proyecto ha sido superior al esperado, por lo que organizar y optimizar el tiempo ha sido un reto complejo de superar. Este riesgo se había contemplado en la definición del proyecto, y por tanto se ha superado con éxito.
- Infraestructura de virtualización dimensionada inadecuadamente. Inicialmente se planteó la utilización de hardware QNAP TS473, pero al tratar de desplegar varias máquinas virtuales al mismo tiempo, el sistema se ralentizaba por el aumento de RAM y CPU. La solución para este caso ha sido utilizar otro hardware con mayores prestaciones. Este riesgo se había contemplado en la definición del proyecto, y por tanto se

ha superado con éxito. Es conveniente indicar que se han desplegado para el proyecto en torno a 45 máquinas virtuales durante el transcurso del Trabajo de las cuales solo se han utilizado 3 de ellas.

- El desconocimiento de las herramientas ha sido un reto importante, ya que ha requerido la lectura de documentación, realizar consultas en los foros especializados para orientar adecuadamente el camino a seguir para obtener el resultado esperado. Este riesgo se había contemplado en la definición del proyecto, y por tanto se ha superado con éxito.

10.3 Identificación de puntos de mejora y trabajos futuros

Durante el desarrollo del trabajo se han identificado una serie de mejoras que en un entorno productivo, ya sea una pequeña o gran empresa debe tener en consideración a la hora de valorar su puesta en servicio.

- A nivel de servidores de aplicaciones:
  - Es aconsejable crear usuarios de servicios para las aplicaciones que se desplieguen con el mínimo de privilegios para su correcta ejecución y evitar comprometer otros sistemas.
  - La información almacenada en los servidores debe ser cifrada para dificultar su interpretación en caso de ser comprometida o interceptada.
  - Se debe restringir el acceso a los servidores de aplicación al grupo de usuarios que sea necesario teniendo en cuenta las reglas de seguridad de red y la aplicación.
- A nivel de aplicaciones:
  - Se debe segurizar los accesos de las aplicaciones mediante los protocolos adecuados y certificados válidos.
  - La transmisión de información entre aplicaciones debe realizarse de manera segura y cifrada, para evitar en caso de capturar los datos de red, no poder interpretarlos.
- Respaldo:
  - Se debe aplicar las medidas de respaldo de logs y su rotación para poder tener una histórico de información que pueda ayudar en las investigaciones de incidentes de seguridad. También es importante realizar un respaldo de las configuraciones.

En cuanto al Trabajo y en concreto a las aplicaciones con las que se ha trabajado se plantean las siguientes propuestas de mejoras.

- Aplicación Zeek:
  - Zeek es una herramienta muy versátil, la cual mediante una programación de script adecuada (políticas) permite obtener grandes resultados en cuando al análisis del tráfico. El proyecto actual se ha centrado en la integración, pero es posible utilizar y personalizar Zeek para poder explotar más aun su potencial.

- Sería interesante poder realizar una comparación real del producto comercial de Coreligth[24] para tener una opinión objetiva sobre los dos productos, donde Zeek es el core de ambos.
- Mejoraría mucho la gestión y administración de Zeek mediante un entorno gráfico, a pesar de ser una herramienta sencilla a nivel general.
- La creación de script personalizados para Zeek es un campo que no se ha abordado, ya que daría para un proyecto como tal, por lo que considera importante este apartado como proyecto de mejora.

- Aplicación ELK:
  - ELK es un conjunto de herramienta con un potencial abrumador, el margen de mejora y optimización es amplio. En este trabajo solo se ha podido un conocimiento base, pero sería muy interesante un proyecto que abarcara todos los servicios que se pueden implementar y como implementarlo desde un punto básico hasta un punto experto.
  - La correlación de datos es un punto que debido al tiempo no ha sido posible investigar en profundidad.
- Aplicación IntelMQ:
  - Esta herramienta, a pesar de estar pensada para la recolección de información, es decir, herramienta de inteligencia, es multipropósito, por lo que es conveniente asegurar la integración con cada uno de los tipos de feed que puedan ir apareciendo de los diferentes proveedores que ofrecen, ya sea de manera gratuito o no información reputacional.

24 Coreligthn: https://www.corelight.com/

11. Glosario

- **Activo:** servidor, ordenador, equipo de red o similar de especial importancia para la organización.
- **Agente**: software que realiza las funciones de supervisión de una activo
- **ASCII**: código de caracteres basado en alfabeto latino
- Botnet: Red de robots (ordenadores) que realizan procesos automáticos.
- **BSD**: (*Berkeley Software Distribution)* Sistema operativo Unix.
- **C&C:** (*Command and Control*) Sistema de equipos para el control automatizado de equipos remotos.
- **Cluster**: conjunto de ordenadores que trabajan de manera simultanea para ofrecer alta disponibilidad del servicio que ofrecen
- **CPU**: (*Central Processing Unit*) es el hardware de un ordenador que realiza el procesado de datos
- **IP**: Es un conjunto de números que identifica de manera lógica a una tarjeta de red de un ordenador.
- **JSON**: (*JavaScript Object Notation*) Es un formato de texto para el intercambio de dato.
- **Linux**: es un conjunto de sistemas operativos libres multiplataforma, multiusuario y multitarea basados en Unix.
- **Logs**: hace referencia a la grabación de datos secuenciales en fichero de un proceso particular.
- **Malware**: es un software dañino para los ordenadores.
- **Match**: proceso de coincidencia de diferentes campos.
- **Memoria**: componente de un ordenador donde se almacena información.
- **Monitorización:** servicio de supervisión de un proceso concreto.
- **NIDS**: (*Network Intrusion Detection System*) Sistema de detección de intrusión de red.
- **Parsear**: Se trata del proceso de dividir datos para su tratamiento posterior.
- **Pharmin**: se trata de un tipo de estafa relaciona con el uso de DNS que se realiza vía telemática.
- **Phishing**: se trata de estafas realizadas mediante medios telemáticos suplantando la identidad de webs.
- **Pipeline**: proceso de transferencia de datos, hilos de comunicación.
- **Protocolo de red**: reglas por las que se rigen las comunicaciones telemáticas en los sistemas de información.
- **Proxy**: aplicación que realiza las veces de intermediario entre cliente y servidor en un sistema de información.

- **Ransomware**: es un software que cifra el contenido del ordenador y el malhechor solicita un rescate.
- **Router**: equipo electrónico para las conexiones de ordenadores para encaminar las comunicaciones.
- **Script:** se trata de un programa por lo general simple
- **Spoofing**: hace referencia a la suplantación de identidad con objetivo fraudulentos
- **Spyware**: es un malware que recopila información del ordenador infectado
- **Switch**: equipo electrónico para las conexiones de múltiples ordenadores para crear una red de ordenadores.
- **Tarjeta de red:** es el hardware de un ordenador que se utiliza para establecer comunicaciones de red.
- **Unix**: es un sistema operativo multitarea y multiusuario.
- **URL**: (*Uniform Resource Locator*) es un identificador de recursos locales basados en caracteres.
- **Virtualización**: es la creación a través de software de una versión virtual de algún recurso tecnológico.

nota: fuentes de información extasiadas de https://es.wikipedia.org

12. Bibliografía y fuentes consultadas

En este apartado se presenta la bibliografía y fuentes consultas para el desarrollo del Trabajo.

12.1 Bibliografía

- Descripción: Comparison of IDS Suitability for Covert ChannelsDetection
  URL:https://www.it.murdoch.edu.au/nsrg/cc_detection_ids/reports/Murdoch_University_IT_NSRG_TR20170818A.pdf

12.2 Páginas web consultadas

Zeek

- Descripción: The Zeek Network Security Monitor
  URL: https://www.zeek.org/

- Descripción: Zeek Installation
  URL: https://docs.zeek.org/en/stable/install/index.html

- Descripción: Github - Zeek Network Monitoring Project
  URL: https://github.com/zeek/

- Descripción: Configures bro to write logs out in JSON
  URL: https://gist.github.com/dcode/97039239c74a1c1e420c

- Descripción: Part 1: Install/Setup Zeek + pf_ring on Ubuntu 18.04 on Proxmox 5.3 + openVswitch
  URL: https://holdmybeersecurity.com/2019/04/03/part-1-install-setup-zeek-pf_ring-on-ubuntu-18-04-on-proxmox-5-3-openvswitch/

ELK

- Descripción: Kibana Guide
  URL: https://www.elastic.co/guide/en/kibana/current/index.html

- Descripción: Elasticsearch Reference
  URL:
  https://www.elastic.co/guide/en/elasticsearch/reference/7.5/index.html

- Descripción: Logstash Reference
  URL: https://www.elastic.co/guide/en/logstash/7.5/index.html

- Descripción: Filebeat Reference
  URL: https://www.elastic.co/guide/en/beats/filebeat/7.5/index.html

- Descripción: Foro de discusión Elastic
  URL: https://discuss.elastic.co/

- Descripción: Discuss Elastic - Compare two datasets (failed)
  URL: https://discuss.elastic.co/t/compare-two-datasets-failed/211968

Intelmq

- Descripción: IntelMQ - Incident Handling Automation
  URL: https://github.com/certtools/intelmq

Otros

- Descripción: How To Remotely Copy Files Over SSH Without Entering Your Password
  URL: https://www.howtogeek.com/66776/HOW-TO-REMOTELY-COPY-FILES-OVER-SSH-WITHOUT-ENTERING-YOUR-PASSWORD/

- Descripción: Backups remotos con rsync a través de ssh sin contraseña
  URL: https://trasteandoarduino.com/2016/03/24/backups-remotos-con-rsync-a-traves-de-ssh-sin-contrasena/

- Descripción: Crontab in Linux with 20 Useful Examples to Schedule Jobs
  URL: https://tecadmin.net/crontab-in-linux-with-20-examples-of-cron-schedule/

- Descripción: Draw.io
  URL: https://www.draw.io/

- Descripción: Ubuntu
  URL: https://ubuntu.com/

12.3 Videos consultados

- Descripción: Watcher Lab — Creating Your First Alert (Video 1)
  URL: https://www.elastic.co/es/videos/watcher-lab-creating-your-first-alert?blade=video&hulk=youtube

- Descripción: Watcher Lab — Using Elasticsearch Aggregations in Your Watch (Video 2)
  URL: https://www.youtube.com/watch?v=-UVUzGMpyoo

- Descripción: Watcher Lab — Creating Alerts with Dynamic Threshold (Video 3)
  URL: https://www.elastic.co/es/videos/watcher-lab-creating-alerts-with-dynamic-threshold?blade=video&hulk=youtube

- Descripción: Time Series Visual Builder in Kibana

- URL: https://www.youtube.com/watch?v=CNR-4kZ6v_E&list=PLcyy31IBGJ4yzhdvkLM9Yt68FHcuoxN8L

13. Anexos

13.1 ANEXO I. Instalación Zeek

El objetivo de este apartad es representar los pasos seguidos para la instalación de Zeek.

Estructura

Se identifican en color las líneas de entrada para resaltar donde se encuentran los comandos

Durante la instalación de utilizan comentarios para explicar algunos de los comandos aplicados.

```
usuario@vm07pec4zeek: usuario / host
### Comentarios
Comandos
```

Proceso de instalación inicial

A continuación, se inicia el procedimiento seguido para el despliegue de Zeek.

```
usuario@vm07pec4zeek:~$ cat /etc/*-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=19.10
DISTRIB_CODENAME=eoan
DISTRIB_DESCRIPTION="Ubuntu 19.10"
NAME="Ubuntu"
VERSION="19.10 (Eoan Ermine)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 19.10"
VERSION_ID="19.10"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=eoan
UBUNTU_CODENAME=eoan
usuario@vm07pec4zeek:~$ uname -a
Linux vm07pec4zeek 5.3.0-24-generic #26-Ubuntu SMP Thu Nov 14 01:33:18 UTC 2019 x86_64 x86_64
x86_64 GNU/Linux
usuario@vm07pec4zeek:~$ uname -m
x86_64
usuario@vm07pec4zeek:~$ uname -r
5.3.0-24-generic


### Configuración de la tarjeta de red en modo PROMISUO.

root@vm07pec4zeek:/home/usuario# ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:47:6d:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.29.90/24 brd 192.168.29.255 scope global dynamic ens160
       valid_lft 7074sec preferred_lft 7074sec
    inet6 fe80::20c:29ff:fe47:6d6c/64 scope link
       valid_lft forever preferred_lft forever

### Habilitamos el modo promiscuo en la tarjeta de red (ens160)

root@vm07pec4zeek:/home/usuario# ip link set ens160 promisc on
root@vm07pec4zeek:/home/usuario# ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

```
2: ens160: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
1000
    link/ether 00:0c:29:47:6d:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.29.90/24 brd 192.168.29.255 scope global dynamic ens160
       valid_lft 7053sec preferred_lft 7053sec
    inet6 fe80::20c:29ff:fe47:6d6c/64 scope link
       valid_lft forever preferred_lft forever

### Instalación de paquetes y dependencias para la puesta en servicio de Zeek.

root@vm07pec4zeek:/home/usuario# apt-get install cmake make gcc g++ gdb flex bison libpcap-dev
libssl-dev python-dev swig zlib1g-dev libgeoip-dev build-essential libelf-dev libmagic-dev
libmaxminddb-dev openssl python-pip git iperf libkrb5-dev systemtap libtool libgdbm-dev
libreadline-dev doxygen
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssl ya está en su versión más reciente (1.1.1c-1ubuntu4).
fijado openssl como instalado manualmente.
git ya está en su versión más reciente (1:2.20.1-2ubuntu1.19.10.1).
fijado git como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
  autoconf automake autotools-dev binutils binutils-common binutils-x86-64-linux-gnu cmake-data
comerr-dev cpp cpp-9 dpkg-dev fakeroot g++-9 gcc-9 gdbserver geoip-bin
  javascript-common krb5-multidev libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-
merge-perl libasan5 libatomic1 libavahi-client3 libavahi-common-data
  libavahi-common3 libbabeltrace1 libbinutils libc-dev-bin libc6-dbg libc6-dev libcc1-0 libclang1-9
libdpkg-perl libdw1 libexpat1-dev libfakeroot libfile-fcntllock-perl libfl-dev
  libgcc-9-dev libgomp1 libgssrpc4 libisl21 libitm1 libjs-jquery libjs-sphinxdoc libjs-underscore
libjsoncpp1 libkadm5clnt-mit11 libkadm5srv-mit11 libkdb5-9 libllvm9 liblsan0
  libltdl-dev libltdl7 libmaxminddb0 libmpc3 libncurses-dev libnetaddr-ip-perl libnspr4 libnss3
libpcap0.8-dev libpython-all-dev libpython-dev libpython-stdlib libpython2-dev
  libpython2-stdlib libpython2.7 libpython2.7-dev libpython2.7-minimal libpython2.7-stdlib
libquadmath0 librhash0 libsocket6-perl libstdc++-9-dev libtsan0 libubsan1 libuv1
  libxapian30 linux-libc-dev m4 manpages-dev python python-all python-all-dev python-asn1crypto
python-cffi-backend python-configparser python-crypto python-cryptography
  python-dbus python-entrypoints python-enum34 python-gi python-ipaddress python-keyring python-
keyrings.alt python-minimal python-pip-whl python-pkg-resources
  python-secretstorage python-setuptools python-six python-wheel python-xdg python2 python2-dev
python2-minimal python2.7 python2.7-dev python2.7-minimal swig3.0 systemtap-common
  systemtap-runtime
Paquetes sugeridos:
  autoconf-archive gnu-standards autoconf-doc gettext binutils-doc bison-doc cmake-doc ninja-build
doc-base cpp-doc gcc-9-locales doxygen-latex doxygen-doc doxygen-gui graphviz
  debian-keyring flex-doc g++-multilib g++-9-multilib gcc-9-doc libstdc++6-9-dbg gcc-multilib gcc-
doc gcc-9-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg
  libasan5-dbg liblsan0-dbg libtsan0-dbg libubsan1-dbg libquadmath0-dbg gdb-doc apache2 | lighttpd
| httpd krb5-doc glibc-doc bzr krb5-user libtool-doc mmdb-bin ncurses-doc
  readline-doc libssl-doc libstdc++-9-doc gfortran | fortran95-compiler gcj-jdk xapian-tools m4-doc
make-doc python-doc python-tk python-cryptography-doc
  python-cryptography-vectors python-dbus-dbg python-dbus-doc python-enum34-doc python-gi-cairo
gnome-keyring libkf5wallet-bin gir1.2-gnomekeyring-1.0 python-gdata python-keyczar
  python-secretstorage-doc python-setuptools-doc python2-doc python2.7-doc binfmt-support swig-doc
swig-examples swig3.0-examples swig3.0-doc systemtap-doc vim-addon-manager
Se instalarán los siguientes paquetes NUEVOS:
  autoconf automake autotools-dev binutils binutils-common binutils-x86-64-linux-gnu bison build-
essential cmake cmake-data comerr-dev cpp cpp-9 doxygen dpkg-dev fakeroot flex
  g++ g++-9 gcc gcc-9 gdb gdbserver geoip-bin iperf javascript-common krb5-multidev libalgorithm-
diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1
  libavahi-client3 libavahi-common-data libavahi-common3 libbabeltrace1 libbinutils libc-dev-bin
libc6-dbg libc6-dev libcc1-0 libclang1-9 libdpkg-perl libdw1 libelf-dev
  libexpat1-dev libfakeroot libfile-fcntllock-perl libfl-dev libgcc-9-dev libgdbm-dev libgeoip-dev
libgomp1 libgssrpc4 libisl21 libitm1 libjs-jquery libjs-sphinxdoc
  libjs-underscore libjsoncpp1 libkadm5clnt-mit11 libkadm5srv-mit11 libkdb5-9 libkrb5-dev libllvm9
liblsan0 libltdl-dev libltdl7 libmagic-dev libmaxminddb-dev libmaxminddb0
  libmpc3 libncurses-dev libnetaddr-ip-perl libnspr4 libnss3 libpcap-dev libpcap0.8-dev libpython-
all-dev libpython-dev libpython-stdlib libpython2-dev libpython2-stdlib
  libpython2.7 libpython2.7-dev libpython2.7-minimal libpython2.7-stdlib libquadmath0 libreadline-
dev librhash0 libsocket6-perl libssl-dev libstdc++-9-dev libtool libtsan0
  libubsan1 libuv1 libxapian30 linux-libc-dev m4 make manpages-dev python python-all python-all-dev
python-asn1crypto python-cffi-backend python-configparser python-crypto
  python-cryptography python-dbus python-dev python-entrypoints python-enum34 python-gi python-
ipaddress python-keyring python-keyrings.alt python-minimal python-pip
  python-pip-whl python-pkg-resources python-secretstorage python-setuptools python-six python-
wheel python-xdg python2 python2-dev python2-minimal python2.7 python2.7-dev
  python2.7-minimal swig swig3.0 systemtap systemtap-common systemtap-runtime zlib1g-dev
0 actualizados, 139 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 108 MB de archivos.
Se utilizarán 475 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Continuamos con la preparación del entorno. Especial interés en la instalación de **PF_RING**.

```
root@vm07pec4zeek:/home/usuario# cd /opt
root@vm07pec4zeek:/opt# git clone https://github.com/ntop/PF_RING.git
Cloning into 'PF_RING'...
remote: Enumerating objects: 106, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (76/76), done.
remote: Total 24626 (delta 57), reused 62 (delta 28), pack-reused 24520
Receiving objects: 100% (24626/24626), 68.50 MiB | 10.26 MiB/s, done.
Resolving deltas: 100% (18101/18101), done.
root@vm07pec4zeek:/opt# cd PF_RING/kernel
root@vm07pec4zeek:/opt/PF_RING/kernel# make
#make -C /lib/modules/5.3.0-24-generic/build SUBDIRS=/opt/PF_RING/kernel
EXTRA_CFLAGS='-I/opt/PF_RING/kernel -DGIT_REV="\"dev:b623603281f0ace3a8554f9a329863e9ee6e1626\"" -
no-pie -fno-pie' modules
make -C /lib/modules/5.3.0-24-generic/build M=/opt/PF_RING/kernel
EXTRA_CFLAGS='-I/opt/PF_RING/kernel -DGIT_REV="\"dev:b623603281f0ace3a8554f9a329863e9ee6e1626\"" -
no-pie -fno-pie' modules
make[1]: Entering directory '/usr/src/linux-headers-5.3.0-24-generic'
  CC [M]  /opt/PF_RING/kernel/pf_ring.o
/opt/PF_RING/kernel/pf_ring.c: In function ‘hash_pkt_cluster’:
/opt/PF_RING/kernel/pf_ring.c:3806:7: warning: this statement may fall through [-Wimplicit-
fallthrough=]
 3806 |     if(l3_proto == IPPROTO_TCP)
      |       ^
/opt/PF_RING/kernel/pf_ring.c:3813:3: note: here
 3813 |   case cluster_per_flow_2_tuple:
      |   ^~~~
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /opt/PF_RING/kernel/pf_ring.mod.o
  LD [M]  /opt/PF_RING/kernel/pf_ring.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.3.0-24-generic'
root@vm07pec4zeek:/opt/PF_RING/kernel# insmod ./pf_ring.ko

root@vm07pec4zeek:/opt/PF_RING/kernel# cd ../userland

root@vm07pec4zeek:/opt/PF_RING/userland# make
./configure
checking for gcc... gcc
  [ proceso make ][ proceso make ][ proceso make ][ proceso make ][ proceso make ][ proceso make ]
root@vm07pec4zeek:/opt/PF_RING/userland# cd lib

root@vm07pec4zeek:/opt/PF_RING/userland/lib# ./configure --prefix=/opt/PF_RING/

root@vm07pec4zeek:/opt/PF_RING/userland/lib# make install

root@vm07pec4zeek:/opt/PF_RING/userland/lib# cd ../libpcap

root@vm07pec4zeek:/opt/PF_RING/userland/libpcap# ./configure --prefix=/opt/PF_RING/

root@vm07pec4zeek:/opt/PF_RING/userland/libpcap# make install

root@vm07pec4zeek:/opt/PF_RING/userland/libpcap# cd ../tcpdump-*

root@vm07pec4zeek:/opt/PF_RING/userland/tcpdump-4.9.2# ./configure --prefix=/opt/PF_RING/

root@vm07pec4zeek:/opt/PF_RING/userland/tcpdump-4.9.2# make install

root@vm07pec4zeek:/opt/PF_RING/userland/tcpdump-4.9.2# cd ../../kernel

root@vm07pec4zeek:/opt/PF_RING/kernel# make
#make -C /lib/modules/5.3.0-24-generic/build SUBDIRS=/opt/PF_RING/kernel
EXTRA_CFLAGS='-I/opt/PF_RING/kernel -DGIT_REV="\"dev:b623603281f0ace3a8554f9a329863e9ee6e1626\"" -
no-pie -fno-pie' modules
make -C /lib/modules/5.3.0-24-generic/build M=/opt/PF_RING/kernel
EXTRA_CFLAGS='-I/opt/PF_RING/kernel -DGIT_REV="\"dev:b623603281f0ace3a8554f9a329863e9ee6e1626\"" -
no-pie -fno-pie' modules
make[1]: Entering directory '/usr/src/linux-headers-5.3.0-24-generic'
  Building modules, stage 2.
  MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-5.3.0-24-generic'
root@vm07pec4zeek:/opt/PF_RING/kernel# make install
mkdir -p /lib/modules/5.3.0-24-generic/kernel/net/pf_ring
cp *.ko /lib/modules/5.3.0-24-generic/kernel/net/pf_ring
mkdir -p /usr/include/linux
cp linux/pf_ring.h /usr/include/linux
/sbin/depmod 5.3.0-24-generic
root@vm07pec4zeek:/opt/PF_RING/kernel# echo "pf_ring" >> /etc/modules
root@vm07pec4zeek:/opt/PF_RING/kernel# lsmod | grep pf_ring
pf_ring               724992  0

### Reiniciamos el host para comprobar que se ha guardado la configuración pf_ring
root@vm07pec4zeek:/opt/PF_RING/kernel# reboot
```

```
root@vm07pec4zeek:/opt/PF_RING/kernel# Connection to 192.168.29.90 closed by remote host.
Connection to 192.168.29.90 closed.
HOST02321:~ userlocale$ ssh usuario@192.168.29.90
usuario@192.168.29.90's password:
Welcome to Ubuntu 19.10 (GNU/Linux 5.3.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of mié dic 11 23:33:30 CET 2019

  System load:  0.36               Processes:             261
  Usage of /:   14.3% of 48.96GB   Users logged in:       0
  Memory usage: 3%                 IP address for ens160: 192.168.29.90
  Swap usage:   0%


0 updates can be installed immediately.
0 of these updates are security updates.


Last login: Wed Dec 11 23:09:58 2019 from 192.168.29.10
usuario@vm07pec4zeek:~$ sudo su
[sudo] password for usuario:
root@vm07pec4zeek:/home/usuario# lsmod | grep pf_ring
pf_ring               724992  0
root@vm07pec4zeek:/home/usuario# ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:47:6d:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.29.90/24 brd 192.168.29.255 scope global dynamic ens160
       valid_lft 7099sec preferred_lft 7099sec
    inet6 fe80::20c:29ff:fe47:6d6c/64 scope link
       valid_lft forever preferred_lft forever
root@vm07pec4zeek:/home/usuario# ip link set ens160 promisc on
root@vm07pec4zeek:/home/usuario# ip ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen
1000
    link/ether 00:0c:29:47:6d:6c brd ff:ff:ff:ff:ff:ff
    inet 192.168.29.90/24 brd 192.168.29.255 scope global dynamic ens160
       valid_lft 7080sec preferred_lft 7080sec
    inet6 fe80::20c:29ff:fe47:6d6c/64 scope link
       valid_lft forever preferred_lft forever
```

A continuación se inicia el proceso de instalación de Zeek.

```
root@vm07pec4zeek:/home/usuario# cd /tmp

root@vm07pec4zeek:/tmp# git clone --recursive https://github.com/zeek/zeek
Cloning into 'zeek'...
remote: Enumerating objects: 54, done.
remote: Counting objects: 100% (54/54), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 119524 (delta 13), reused 29 (delta 10), pack-reused 119470

root@vm07pec4zeek:/tmp# cd zeek
root@vm07pec4zeek:/tmp/zeek# ./configure --with-pcap=/opt/PF_RING --prefix=/opt/zeek/
Build Directory : build
```

En la máquina que se ha desplegado ha tadado 90 minutos aproximadamente en completar el proceso "*make*".

```
root@vm07pec4zeek:/tmp/zeek# make
make -C build all

root@vm07pec4zeek:/tmp/zeek# make install
make -C build all

root@vm07pec4zeek:/tmp/zeek# echo "$PATH:/opt/zeek/bin" >/etc/environment
```

```
root@vm07pec4zeek:/tmp/zeek# export PATH=/opt/zeek/bin:$PATH
```

Ajustes de la instalación de Zeek y revisión fucional

Una vez se ha realizado la instalación, debemos realizar los ajustes propios de nuestro sistema.

```
root@vm07pec4zeek:/tmp/zeek# nano /opt/zeek/etc/node.cfg
```

```
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# 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

## Below is an example clustered configuration. If you use this,
## remove the [zeek] node above.

#[logger]
#type=logger
#host=localhost
#
[manager]
type=manager
host=localhost
#
[proxy-1]
type=proxy
host=localhost
#
[worker-1]
type=worker
host=localhost
interface=ens160
lb_method=pf_ring
lb_procs=2
#
[worker-2]
type=worker
host=localhost
interface=ens160
lb_method=pf_ring
lb_procs=2
```

```
root@vm07pec4zeek:/tmp/zeek# nano /opt/zeek/etc/networks.cfg
```

```
# List of local networks in CIDR notation, optionally followed by a
# descriptive tag.
# For example, "10.0.0.0/8" or "fe80::/64" are valid prefixes.

10.0.0.0/8          Private IP space
172.16.0.0/12       Private IP space
192.168.0.0/16      Private IP space
```

```
### Se ha completado el proceso de instalación. Se procede con el inicio de Zeek.

root@vm07pec4zeek:/tmp/zeek# /opt/zeek/bin/zeekctl install
creating policy directories ...
installing site policies ...
generating cluster-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
root@vm07pec4zeek:/tmp/zeek# /opt/zeek/bin/zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating cluster-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
```

```
stopping ...
stopping workers ...
stopping proxy ...
stopping manager ...
starting ...
starting manager ...
starting proxy ...
starting workers ...
root@vm07pec4zeek:/tmp/zeek# /opt/zeek/bin/zeekctl status
Name          Type    Host             Status    Pid    Started
manager       manager localhost        running   16781  12 Dec 01:10:03
proxy-1       proxy   localhost        running   16829  12 Dec 01:10:04
worker-1-1    worker  localhost        running   16911  12 Dec 01:10:06
worker-1-2    worker  localhost        running   16915  12 Dec 01:10:06
worker-2-1    worker  localhost        running   16920  12 Dec 01:10:06
worker-2-2    worker  localhost        running   16919  12 Dec 01:10:06

root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl check
manager scripts are ok.
proxy-1 scripts are ok.
worker-1-1 scripts are ok.
worker-1-2 scripts are ok.
worker-2-1 scripts are ok.
worker-2-2 scripts are ok.

root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl df
                                          total  avail  capacity
           manager/localhost        /dev/sda2   49G    35G   24.5 %


root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl diag
[manager]

No core file found.

Zeek 3.1.0-dev.300
Linux 5.3.0-24-generic

Zeek plugins: (none found)

==== No reporter.log

==== stderr.log
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found
send-mail: /usr/sbin/sendmail not found

==== stdout.log
max memory size         (kbytes, -m) unlimited
data seg size           (kbytes, -d) unlimited
virtual memory          (kbytes, -v) unlimited
core file size          (blocks, -c) unlimited

==== .cmdline
-U .status -p zeekctl -p zeekctl-live -p local -p manager local.zeek zeekctl
base/frameworks/cluster zeekctl/auto

==== .env_vars
PATH=/opt/zeek/bin:/opt/zeek/share/zeekctl/scripts:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/snap/bin
ZEEKPATH=/opt/zeek/spool/installed-scripts-do-not-touch/site::/opt/zeek/spool/installed-scripts-do-
not-touch/auto:/opt/zeek/share/zeek:/opt/zeek/share/zeek/policy:/opt/zeek/share/zeek/site
CLUSTER_NODE=manager

==== .status
RUNNING [net_run]

==== No prof.log

==== No packet_filter.log

==== No loaded_scripts.log

[proxy-1]

No core file found.

Zeek 3.1.0-dev.300
Linux 5.3.0-24-generic
```

```
Zeek plugins: (none found)

==== No reporter.log

==== stderr.log

==== stdout.log
max memory size         (kbytes, -m) unlimited
data seg size           (kbytes, -d) unlimited
virtual memory          (kbytes, -v) unlimited
core file size          (blocks, -c) unlimited

==== .cmdline
-U .status -p zeekctl -p zeekctl-live -p local -p proxy-1 local.zeek zeekctl
base/frameworks/cluster zeekctl/auto

==== .env_vars
PATH=/opt/zeek/bin:/opt/zeek/share/zeekctl/scripts:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/snap/bin
ZEEKPATH=/opt/zeek/spool/installed-scripts-do-not-touch/site::/opt/zeek/spool/installed-scripts-do-
not-touch/auto:/opt/zeek/share/zeek:/opt/zeek/share/zeek/policy:/opt/zeek/share/zeek/site
CLUSTER_NODE=proxy-1

==== .status
RUNNING [net_run]

==== No prof.log

==== No packet_filter.log

==== No loaded_scripts.log

[worker-1-1]

No core file found.

Zeek 3.1.0-dev.300
Linux 5.3.0-24-generic

Zeek plugins: (none found)

==== No reporter.log

==== stderr.log
listening on ens160


==== stdout.log
max memory size         (kbytes, -m) unlimited
data seg size           (kbytes, -d) unlimited
virtual memory          (kbytes, -v) unlimited
core file size          (blocks, -c) unlimited

==== .cmdline
-i ens160 -U .status -p zeekctl -p zeekctl-live -p local -p worker-1-1 local.zeek zeekctl
base/frameworks/cluster zeekctl/auto

==== .env_vars
PATH=/opt/zeek/bin:/opt/zeek/share/zeekctl/scripts:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/snap/bin
ZEEKPATH=/opt/zeek/spool/installed-scripts-do-not-touch/site::/opt/zeek/spool/installed-scripts-do-
not-touch/auto:/opt/zeek/share/zeek:/opt/zeek/share/zeek/policy:/opt/zeek/share/zeek/site
CLUSTER_NODE=worker-1-1

==== .status
RUNNING [net_run]

==== No prof.log

==== No packet_filter.log

==== No loaded_scripts.log

[worker-1-2]

No core file found.

Zeek 3.1.0-dev.300
Linux 5.3.0-24-generic

Zeek plugins: (none found)

==== No reporter.log

==== stderr.log
listening on ens160
```

```
==== stdout.log
max memory size         (kbytes, -m) unlimited
data seg size           (kbytes, -d) unlimited
virtual memory          (kbytes, -v) unlimited
core file size          (blocks, -c) unlimited

==== .cmdline
-i ens160 -U .status -p zeekctl -p zeekctl-live -p local -p worker-1-2 local.zeek zeekctl
base/frameworks/cluster zeekctl/auto

==== .env_vars
PATH=/opt/zeek/bin:/opt/zeek/share/zeekctl/scripts:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/snap/bin
ZEEKPATH=/opt/zeek/spool/installed-scripts-do-not-touch/site::/opt/zeek/spool/installed-scripts-do-
not-touch/auto:/opt/zeek/share/zeek:/opt/zeek/share/zeek/policy:/opt/zeek/share/zeek/site
CLUSTER_NODE=worker-1-2

==== .status
RUNNING [net_run]

==== No prof.log

==== No packet_filter.log

==== No loaded_scripts.log

[worker-2-1]

No core file found.

Zeek 3.1.0-dev.300
Linux 5.3.0-24-generic

Zeek plugins: (none found)

==== No reporter.log

==== stderr.log
listening on ens160


==== stdout.log
max memory size         (kbytes, -m) unlimited
data seg size           (kbytes, -d) unlimited
virtual memory          (kbytes, -v) unlimited
core file size          (blocks, -c) unlimited

==== .cmdline
-i ens160 -U .status -p zeekctl -p zeekctl-live -p local -p worker-2-1 local.zeek zeekctl
base/frameworks/cluster zeekctl/auto

==== .env_vars
PATH=/opt/zeek/bin:/opt/zeek/share/zeekctl/scripts:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/snap/bin
ZEEKPATH=/opt/zeek/spool/installed-scripts-do-not-touch/site::/opt/zeek/spool/installed-scripts-do-
not-touch/auto:/opt/zeek/share/zeek:/opt/zeek/share/zeek/policy:/opt/zeek/share/zeek/site
CLUSTER_NODE=worker-2-1

==== .status
RUNNING [net_run]

==== No prof.log

==== No packet_filter.log

==== No loaded_scripts.log

[worker-2-2]

No core file found.

Zeek 3.1.0-dev.300
Linux 5.3.0-24-generic

Zeek plugins: (none found)

==== No reporter.log

==== stderr.log
listening on ens160


==== stdout.log
max memory size         (kbytes, -m) unlimited
data seg size           (kbytes, -d) unlimited
virtual memory          (kbytes, -v) unlimited
```

```
core file size          (blocks, -c) unlimited

==== .cmdline
-i ens160 -U .status -p zeekctl -p zeekctl-live -p local -p worker-2-2 local.zeek zeekctl
base/frameworks/cluster zeekctl/auto

==== .env_vars
PATH=/opt/zeek/bin:/opt/zeek/share/zeekctl/scripts:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin:/snap/bin
ZEEKPATH=/opt/zeek/spool/installed-scripts-do-not-touch/site::/opt/zeek/spool/installed-scripts-do-
not-touch/auto:/opt/zeek/share/zeek:/opt/zeek/share/zeek/policy:/opt/zeek/share/zeek/site
CLUSTER_NODE=worker-2-2

==== .status
RUNNING [net_run]

==== No prof.log

==== No packet_filter.log

==== No loaded_scripts.log


root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl netstats
 worker-1-1: 1576685343.214763 recvd=735155 dropped=0 link=735155
 worker-1-2: 1576685343.241476 recvd=598284 dropped=0 link=598284
 worker-2-1: 1576685343.262026 recvd=855378 dropped=0 link=855378
 worker-2-2: 1576685343.314558 recvd=953167 dropped=0 link=953167


root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl netstats
 worker-1-1: 1576685343.214763 recvd=735155 dropped=0 link=735155
 worker-1-2: 1576685343.241476 recvd=598284 dropped=0 link=598284
 worker-2-1: 1576685343.262026 recvd=855378 dropped=0 link=855378
 worker-2-2: 1576685343.314558 recvd=953167 dropped=0 link=953167
root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl nodes
         manager - addr=127.0.0.1 aux_scripts= count=1 env_vars= ether= host=localhost interface=
lb_interfaces= lb_method= lb_procs= name=manager pin_cpus= test_mykey= type=manager zeekbase=
zone_id=
         proxy-1 - addr=127.0.0.1 aux_scripts= count=1 env_vars= ether= host=localhost interface=
lb_interfaces= lb_method= lb_procs= name=proxy-1 pin_cpus= test_mykey= type=proxy zeekbase=
zone_id=
      worker-1-1 - addr=127.0.0.1 aux_scripts= count=1 env_vars=PCAP_PF_RING_APPNAME=zeek-
ens160,PCAP_PF_RING_CLUSTER_ID=21,PCAP_PF_RING_USE_CLUSTER_PER_FLOW_4_TUPLE=1 ether= host=localhost
interface=ens160 lb_interfaces= lb_method=pf_ring lb_procs=2 name=worker-1-1 pin_cpus= test_mykey=
type=worker zeekbase= zone_id=
      worker-1-2 - addr=127.0.0.1 aux_scripts= count=2 env_vars=PCAP_PF_RING_APPNAME=zeek-
ens160,PCAP_PF_RING_CLUSTER_ID=21,PCAP_PF_RING_USE_CLUSTER_PER_FLOW_4_TUPLE=1 ether= host=localhost
interface=ens160 lb_interfaces= lb_method=pf_ring lb_procs=2 name=worker-1-2 pin_cpus= test_mykey=
type=worker zeekbase= zone_id=
      worker-2-1 - addr=127.0.0.1 aux_scripts= count=3 env_vars=PCAP_PF_RING_APPNAME=zeek-
ens160,PCAP_PF_RING_CLUSTER_ID=21,PCAP_PF_RING_USE_CLUSTER_PER_FLOW_4_TUPLE=1 ether= host=localhost
interface=ens160 lb_interfaces= lb_method=pf_ring lb_procs=2 name=worker-2-1 pin_cpus= test_mykey=
type=worker zeekbase= zone_id=
      worker-2-2 - addr=127.0.0.1 aux_scripts= count=4 env_vars=PCAP_PF_RING_APPNAME=zeek-
ens160,PCAP_PF_RING_CLUSTER_ID=21,PCAP_PF_RING_USE_CLUSTER_PER_FLOW_4_TUPLE=1 ether= host=localhost
interface=ens160 lb_interfaces= lb_method=pf_ring lb_procs=2 name=worker-2-2 pin_cpus= test_mykey=
type=worker zeekbase= zone_id=
root@vm07pec4zeek:/home/usuario# /opt/zeek/bin/zeekctl peerstatus
    manager
1576685368.053619 peer=9C87851D091971823B66F57928002534BF166D00#21192 host=127.0.0.1
status=Broker::PEERED
1576685368.053619 peer=9C87851D091971823B66F57928002534BF166D00#21276 host=127.0.0.1
status=Broker::PEERED
1576685368.053619 peer=9C87851D091971823B66F57928002534BF166D00#21280 host=127.0.0.1
status=Broker::PEERED
1576685368.053619 peer=9C87851D091971823B66F57928002534BF166D00#21281 host=127.0.0.1
status=Broker::PEERED
1576685368.053619 peer=9C87851D091971823B66F57928002534BF166D00#21283 host=127.0.0.1
status=Broker::PEERED
1576685368.053619 peer=9C87851D091971823B66F57928002534BF166D00#31417 host=127.0.0.1
status=Broker::PEERED

    proxy-1
1576685368.086091 peer=9C87851D091971823B66F57928002534BF166D00#21144 host=127.0.0.1
status=Broker::PEERED
1576685368.086091 peer=9C87851D091971823B66F57928002534BF166D00#21276 host=127.0.0.1
status=Broker::PEERED
1576685368.086091 peer=9C87851D091971823B66F57928002534BF166D00#21280 host=127.0.0.1
status=Broker::PEERED
1576685368.086091 peer=9C87851D091971823B66F57928002534BF166D00#21281 host=127.0.0.1
status=Broker::PEERED
1576685368.086091 peer=9C87851D091971823B66F57928002534BF166D00#21283 host=127.0.0.1
status=Broker::PEERED
1576685368.086091 peer=9C87851D091971823B66F57928002534BF166D01#31417 host=127.0.0.1
status=Broker::PEERED
```

```
 worker-1-1
1576685368.114594 peer=9C87851D091971823B66F57928002534BF166D00#21144 host=127.0.0.1
status=Broker::PEERED
1576685368.114594 peer=9C87851D091971823B66F57928002534BF166D00#21192 host=127.0.0.1
status=Broker::PEERED
1576685368.114594 peer=9C87851D091971823B66F57928002534BF166D02#31417 host=127.0.0.1
status=Broker::PEERED

 worker-1-2
1576685368.153098 peer=9C87851D091971823B66F57928002534BF166D00#21144 host=127.0.0.1
status=Broker::PEERED
1576685368.153098 peer=9C87851D091971823B66F57928002534BF166D00#21192 host=127.0.0.1
status=Broker::PEERED
1576685368.153098 peer=9C87851D091971823B66F57928002534BF166D03#31417 host=127.0.0.1
status=Broker::PEERED

 worker-2-1
1576685368.185113 peer=9C87851D091971823B66F57928002534BF166D00#21144 host=127.0.0.1
status=Broker::PEERED
1576685368.185113 peer=9C87851D091971823B66F57928002534BF166D00#21192 host=127.0.0.1
status=Broker::PEERED
1576685368.185113 peer=9C87851D091971823B66F57928002534BF166D04#31417 host=127.0.0.1
status=Broker::PEERED

 worker-2-2
1576685368.219635 peer=9C87851D091971823B66F57928002534BF166D00#21144 host=127.0.0.1
status=Broker::PEERED
1576685368.219635 peer=9C87851D091971823B66F57928002534BF166D00#21192 host=127.0.0.1
status=Broker::PEERED
1576685368.219635 peer=9C87851D091971823B66F57928002534BF166D05#31417 host=127.0.0.1
status=Broker::PEERED

root@serdeb09tmf4:/tmp# ldd /opt/zeek/bin/zeek | grep pcap
        libpcap.so.1 => /opt/PF_RING/lib/libpcap.so.1 (0x00007fbbf1c97000)
```

Instalación de GeoLite2-City

Realizamos la instalación del MMDB de GeoLite2-City

```
root@vm07pec4zeek:/tmp/zeek# ###  Realizamos extra de instalación
root@vm07pec4zeek:/tmp/zeek# ###  GeoLite2-City Database Installation
root@vm07pec4zeek:/tmp# wget http://geolite.maxmind.com/download/geoip/database/GeoLite2-
City.tar.gz
--2019-12-12 01:14:21--  http://geolite.maxmind.com/download/geoip/database/GeoLite2-City.tar.gz
Resolving geolite.maxmind.com (geolite.maxmind.com)... 104.17.200.89, 104.17.201.89,
2606:4700::6811:c959, ...
Connecting to geolite.maxmind.com (geolite.maxmind.com)|104.17.200.89|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30711694 (29M) [application/gzip]
Saving to: 'GeoLite2-City.tar.gz'

GeoLite2-City.tar.gz                                              100%
[=================================================================================================
==================================================>]  29,29M  11,0MB/s    in 2,7s

2019-12-12 01:14:24 (11,0 MB/s) - 'GeoLite2-City.tar.gz' saved [30711694/30711694]

root@vm07pec4zeek:/tmp# tar zxf GeoLite2-City.tar.gz
root@vm07pec4zeek:/tmp# ls -la
total 30048
drwxrwxrwt 14 root root     4096 dic 12 01:14 .
drwxr-xr-x 20 root root     4096 dic 10 00:11 ..
drwxrwxrwt  2 root root     4096 dic 11 23:32 .font-unix
drwxr-xr-x  2 root root     4096 dic 10 16:38 GeoLite2-City_20191210
-rw-r--r--  1 root root 30711694 dic 10 16:38 GeoLite2-City.tar.gz
drwxrwxrwt  2 root root     4096 dic 11 23:32 .ICE-unix
drwx------  3 root root     4096 dic 11 23:32 snap.lxd
drwx------  3 root root     4096 dic 11 23:32 systemd-private-dcebf827d22f4a6f98f7fe734218c20f-
systemd-logind.service-3isAdi
drwx------  3 root root     4096 dic 11 23:32 systemd-private-dcebf827d22f4a6f98f7fe734218c20f-
systemd-resolved.service-HHCdQh
drwx------  3 root root     4096 dic 11 23:32 systemd-private-dcebf827d22f4a6f98f7fe734218c20f-
systemd-timesyncd.service-CeaCTf
drwxrwxrwt  2 root root     4096 dic 11 23:32 .Test-unix
drwx------  2 root root     4096 dic 11 23:32 vmware-root_822-2965448144
drwxrwxrwt  2 root root     4096 dic 11 23:32 .X11-unix
drwxrwxrwt  2 root root     4096 dic 11 23:32 .XIM-unix
drwxr-xr-x 11 root root     4096 dic 11 23:39 zeek
root@vm07pec4zeek:/tmp# cd GeoLite2-City_20191210/
root@vm07pec4zeek:/tmp/GeoLite2-City_20191210# ls
COPYRIGHT.txt  GeoLite2-City.mmdb  LICENSE.txt  README.txt
root@vm07pec4zeek:/tmp/GeoLite2-City_20191210# mv GeoLite2-City.mmdb /usr/share/GeoIP/GeoLite2-
City.mmdb
```

```
root@vm07pec4zeek:/tmp/GeoLite2-City_20191210# ls -la /usr/share/GeoIP/
total 72416
drwxr-xr-x   2 root root     4096 dic 12 01:15 .
drwxr-xr-x 125 root root     4096 dic 11 23:14 ..
-rw-r--r--   1 root root  2039607 sep 20 09:45 GeoIP.dat
-rw-r--r--   1 root root  7898259 sep 20 09:45 GeoIPv6.dat
-rw-r--r--   1 root root 64204493 dic 10 16:38 GeoLite2-City.mmdb
root@vm07pec4zeek:/tmp/GeoLite2-City_20191210# /opt/zeek/bin/zeek -e "print
lookup_location(8.8.8.8);"
[country_code=US, region=<uninitialized>, city=<uninitialized>, latitude=37.751, longitude=-97.822]
root@vm07pec4zeek:/tmp/GeoLite2-City_20191210# /opt/zeek/bin/zeek -e "print
lookup_location(1.1.1.1);"
[country_code=AU, region=<uninitialized>, city=<uninitialized>, latitude=-33.494,
longitude=143.2104]
```

13.2 ANEXO II. Instalación IntelMQ

En este apartado se identifican las conclusiones del desarrollo del Trabajo, como es el cumplimiento de lo hitos marcados, los problemas encontrados y trabajos futuros.

Descripción

El objetivo de este manual es representar los pasos seguidos para la instalación de IntelMQ en un host y mostrar los resultados de cada uno de los comandos. Esta información debe servir como referencia de los pasos para la instalación. Las fuentes de información para el desarrollo de este manual proceden de los repositorios públicos de IntelMQ que se pueden consultar en la siguiente dirección web:

- https://github.com/certtools/intelmq

Estructura

Se identifican en color las lineas de entrada para resaltar donde se encuentran los comandos. Durante la instalación de utilizan comentarios para explicar algunos de los comandos aplicados.

```
usuario@vm07pec4zeek: usuario / host
### Comentarios
Comandos
```

Proceso de instalación inicial

A continuación se documenta el proceso seguido para la instalación de de IntelMQ.

```
### Acceso al host e información del sistema operativo y actualización base del sistema operativo

PC0231:~ usuario$ ssh usuario@192.168.29.88
The authenticity of host '192.168.29.88 (192.168.29.88)' can't be established.
ECDSA key fingerprint is SHA256:2eeaNuzBimNv7NdpA6eOiNz877vGeGfB26XWovtKagQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.29.88' (ECDSA) to the list of known hosts.
usuario@192.168.29.88's password:
Welcome to Ubuntu 19.10 (GNU/Linux 5.3.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of jue dic 12 00:04:37 CET 2019

  System load:  0.0               Processes:             209
  Usage of /:   12.6% of 48.96GB  Users logged in:       0
  Memory usage: 5%                IP address for ens160: 192.168.29.88
  Swap usage:   0%


0 updates can be installed immediately.
0 of these updates are security updates.
```

```
Last login: Wed Dec 11 21:03:28 2019
usuario@vm07pec4intel:~$ sudo apt-get update

usuario@vm07pec4intel:~$ sudo apt-get upgrade

usuario@vm07pec4intel:~$ sudo apt-get dist-upgrade

usuario@vm07pec4intel:~$ date
jue dic 12 00:05:31 CET 2019
usuario@vm07pec4intel:~$ cat /etc/issue
Ubuntu 19.10 \n \l
usuario@vm07pec4intel:~$ cat /etc/*release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=19.10
DISTRIB_CODENAME=eoan
DISTRIB_DESCRIPTION="Ubuntu 19.10"
NAME="Ubuntu"
VERSION="19.10 (Eoan Ermine)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 19.10"
VERSION_ID="19.10"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=eoan
UBUNTU_CODENAME=eoan

### Instalación de dependencias necesarias

usuario@vm07pec4intel:~$ sudo apt install python3-pip python3-dnspython python3-psutil python3-
redis python3-requests python3-termstyle python3-tz python3-dateutil
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
python3-requests ya está en su versión más reciente (2.21.0-1).
fijado python3-requests como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dh-python dpkg-dev
fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0 libdpkg-
perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-9-dev libgomp1
  libisl21 libitm1 liblsan0 libmpc3 libpython3-dev libpython3.7-dev libquadmath0 libstdc++-9-dev
libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-crypto
  python3-dev python3-distutils python3-keyrings.alt python3-lib2to3 python3-setuptools python3-
wheel python3-xdg python3.7-dev zlib1g-dev
Paquetes sugeridos:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-multilib g++-9-multilib gcc-9-doc libstdc+
+6-9-dbg gcc-multilib autoconf automake libtool flex bison gdb gcc-doc
  gcc-9-multilib libgcc1-dbg libgomp1-dbg libitm1-dbg libatomic1-dbg libasan5-dbg liblsan0-dbg
libtsan0-dbg libubsan1-dbg libquadmath0-dbg glibc-doc bzr libstdc++-9-doc make-doc
  gir1.2-gnomekeyring-1.0 python-psutil-doc python3-hiredis python-setuptools-doc
Se instalarán los siguientes paquetes NUEVOS:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dh-python dpkg-dev
fakeroot g++ g++-9 gcc gcc-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcc1-0 libdpkg-
perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-9-dev libgomp1
  libisl21 libitm1 liblsan0 libmpc3 libpython3-dev libpython3.7-dev libquadmath0 libstdc++-9-dev
libtsan0 libubsan1 linux-libc-dev make manpages-dev python-pip-whl python3-crypto
  python3-dateutil python3-dev python3-distutils python3-dnspython python3-keyrings.alt python3-
lib2to3 python3-pip python3-psutil python3-redis python3-setuptools
  python3-termstyle python3-tz python3-wheel python3-xdg python3.7-dev zlib1g-dev
0 actualizados, 59 nuevos se instalarán, 0 para eliminar y 0 no actualizados.

usuario@vm07pec4intel:~$ sudo apt install redis-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libhiredis0.14 libjemalloc2 liblua5.1-0 lua-bitop lua-cjson redis-tools
Paquetes sugeridos:
  ruby-redis
Se instalarán los siguientes paquetes NUEVOS:
  libhiredis0.14 libjemalloc2 liblua5.1-0 lua-bitop lua-cjson redis-server redis-tools
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 883 kB de archivos.
Se utilizarán 3.939 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S

usuario@vm07pec4intel:~$ sudo apt install bash-completion jq
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
bash-completion ya está en su versión más reciente (1:2.9-1ubuntu1).
```

```
fijado bash-completion como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
  libjq1 libonig5
Se instalarán los siguientes paquetes NUEVOS:
  jq libjq1 libonig5
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 296 kB de archivos.
Se utilizarán 1.019 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S

usuario@vm07pec4intel:~$ sudo apt install python3-sleekxmpp python3-pymongo python3-psycopg2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  javascript-common libjs-jquery libjs-sphinxdoc libjs-underscore libpq5 python3-bson python3-bson-
ext python3-gnupg python3-gridfs python3-pymongo-ext python3-socks
Paquetes sugeridos:
  apache2 | lighttpd | httpd python-psycopg2-doc python-pymongo-doc
Se instalarán los siguientes paquetes NUEVOS:
  javascript-common libjs-jquery libjs-sphinxdoc libjs-underscore libpq5 python3-bson python3-bson-
ext python3-gnupg python3-gridfs python3-psycopg2 python3-pymongo
  python3-pymongo-ext python3-sleekxmpp python3-socks
0 actualizados, 14 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 1.930 kB de archivos.
Se utilizarán 8.286 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S

usuario@vm07pec4intel:~$ sudo -i
root@vm07pec4intel:~# pip3 install intelmq
Collecting intelmq
  Downloading
https://files.pythonhosted.org/packages/67/3f/48118af0f049e31568aa21cd6f3bf0c2b33dce6ba753059741559
bf62a42/intelmq-2.1.1-py2.py3-none-any.whl (1.0MB)
    100% |████████████████████████████████| 1.0MB 1.0MB/s
Requirement already satisfied: psutil>=1.2.1 in /usr/lib/python3/dist-packages (from intelmq)
(5.5.1)
Requirement already satisfied: redis>=2.10 in /usr/lib/python3/dist-packages (from intelmq) (3.2.1)
Requirement already satisfied: dnspython>=1.11.1 in /usr/lib/python3/dist-packages (from intelmq)
(1.16.0)
Requirement already satisfied: python-termstyle>=0.1.10 in /usr/lib/python3/dist-packages (from
intelmq) (0.1.10)
Requirement already satisfied: python-dateutil>=2.5 in /usr/lib/python3/dist-packages (from
intelmq) (2.7.3)
Requirement already satisfied: pytz>=2012c in /usr/lib/python3/dist-packages (from intelmq)
(2019.2)
Installing collected packages: intelmq
Successfully installed intelmq-2.1.1
root@vm07pec4intel:~# useradd -d /opt/intelmq -U -s /bin/bash intelmq
root@vm07pec4intel:~# sudo intelmqsetup
Moved '/usr/local/lib/python3.7/dist-packages/opt/intelmq/' to '/opt/'.
Created directory '/opt/intelmq/var/lib/bots/file-output'.
Created directory '/opt/intelmq/var/run'.
Created directory '/opt/intelmq/var/log'.
Use example 'pipeline.conf'.
Use example 'defaults.conf'.
Use example 'harmonization.conf'.
Use example 'runtime.conf'.
Setting intelmq as owner for it's directories.
root@vm07pec4intel:~# systemctl status redis.service
● redis-server.service - Advanced key-value store
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-12-12 00:10:38 CET; 3min 52s ago
     Docs: http://redis.io/documentation,
           man:redis-server(1)
 Main PID: 7588 (redis-server)
    Tasks: 4 (limit: 4624)
   Memory: 1.8M
   CGroup: /system.slice/redis-server.service
           └─7588 /usr/bin/redis-server 127.0.0.1:6379

dic 12 00:10:38 vm07pec4intel systemd[1]: Starting Advanced key-value store...
dic 12 00:10:38 vm07pec4intel systemd[1]: Started Advanced key-value store.
root@vm07pec4intel:~# sudo pip3 install beautifulsoup4
Collecting beautifulsoup4
  Downloading
https://files.pythonhosted.org/packages/3b/c8/a55eb6ea11cd7e5ac4bacdf92bac4693b90d3ba79268be1652755
5e186f0/beautifulsoup4-4.8.1-py3-none-any.whl (101kB)
    100% |████████████████████████████████| 102kB 2.2MB/s
Collecting soupsieve>=1.2 (from beautifulsoup4)
  Downloading
https://files.pythonhosted.org/packages/81/94/03c0f04471fc245d08d0a99f7946ac228ca98da4fa75796c507f6
1e688c2/soupsieve-1.9.5-py2.py3-none-any.whl
Installing collected packages: soupsieve, beautifulsoup4
Successfully installed beautifulsoup4-4.8.1 soupsieve-1.9.5
root@vm07pec4intel:~# sudo pip3 install bs4
Collecting bs4
```

```
  Downloading
https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f456174139ec089c769f89a94a1a4025fe967691de
971f314/bs4-0.0.1.tar.gz
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from bs4)
(4.8.1)
Requirement already satisfied: soupsieve>=1.2 in /usr/local/lib/python3.7/dist-packages (from
beautifulsoup4->bs4) (1.9.5)
Building wheels for collected packages: bs4
  Running setup.py bdist_wheel for bs4 ... done
  Stored in directory:
/root/.cache/pip/wheels/a0/b0/b2/4f80b9456b87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: bs4
Successfully installed bs4-0.0.1
```

Comprobación funcional IntelMQ

Comandos aplicados para verificar el correcto funcionamiento de IntelMQ

```
root@vm07pec4intel:~# intelmqctl -h
usage: intelmqctl [-h] [-v] [--type {text,json}] [--quiet]

{list,clear,log,run,check,help,start,stop,restart,reload,status,enable,disable,upgrade-config}
                  ...

        description: intelmqctl is the tool to control intelmq system.

        Outputs are logged to /opt/intelmq/var/log/intelmqctl

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
  --type {text,json}, -t {text,json}
                        choose if it should return regular text or other
                        machine-readable
  --quiet, -q           Quiet mode, useful for reloads initiated scripts like
                        logrotate

subcommands:
  {list,clear,log,run,check,help,start,stop,restart,reload,status,enable,disable,upgrade-config}
    list                Listing bots or queues
    clear               Clear a queue
    log                 Get last log lines of a bot
    run                 Run a bot interactively
    check               Check installation and configuration
    help                Show the help
    start               Start a bot or botnet
    stop                Stop a bot or botnet
    restart             Restart a bot or botnet
    reload              Reload a bot or botnet
    status              Status of a bot or botnet
    enable              Enable a bot
    disable             Disable a bot
    upgrade-config      Upgrade IntelMQ configuration to a newer version.

        intelmqctl [start|stop|restart|status|reload] --group [collectors|parsers|experts|outputs]
        intelmqctl [start|stop|restart|status|reload] bot-id
        intelmqctl [start|stop|restart|status|reload]
        intelmqctl list [bots|queues|queues-and-status]
        intelmqctl log bot-id [number-of-lines [log-level]]
        intelmqctl run bot-id message [get|pop|send]
        intelmqctl run bot-id process [--msg|--dryrun]
        intelmqctl run bot-id console
        intelmqctl clear queue-id
        intelmqctl check
        intelmqctl upgrade-config

Starting a bot:
    intelmqctl start bot-id
Stopping a bot:
    intelmqctl stop bot-id
Reloading a bot:
    intelmqctl reload bot-id
Restarting a bot:
    intelmqctl restart bot-id
Get status of a bot:
    intelmqctl status bot-id

Run a bot directly for debugging purpose and temporarily leverage the logging level to DEBUG:
    intelmqctl run bot-id
Get a pdb (or ipdb if installed) live console.
    intelmqctl run bot-id console
See the message that waits in the input queue.
    intelmqctl run bot-id message get
```

```
See additional help for further explanation.
    intelmqctl run bot-id --help

Starting the botnet (all bots):
    intelmqctl start
    etc.

Starting a group of bots:
    intelmqctl start --group experts
    etc.

Get a list of all configured bots:
    intelmqctl list bots
If -q is given, only the IDs of enabled bots are listed line by line.

Get a list of all queues:
    intelmqctl list queues
If -q is given, only queues with more than one item are listed.

Get a list of all queues and status of the bots:
    intelmqctl list queues-and-status

Clear a queue:
    intelmqctl clear queue-id

Get logs of a bot:
    intelmqctl log bot-id number-of-lines log-level
Reads the last lines from bot log.
Log level should be one of DEBUG, INFO, ERROR or CRITICAL.
Default is INFO. Number of lines defaults to 10, -1 gives all. Result
can be longer due to our logging format!

Upgrade from a previous version:
    intelmqctl upgrade-config
Make a backup of your configuration first, also including bot's configuration files.

Outputs are additionally logged to /opt/intelmq/var/log/intelmqctl

root@vm07pec4intel:~# intelmqctl start
Starting Botnet...
Starting cymru-whois-expert...
Starting deduplicator-expert...
Starting feodo-tracker-browse-collector...
Starting feodo-tracker-browse-parser...
Starting file-output...
Starting gethostbyname-1-expert...
Starting gethostbyname-2-expert...
Starting malc0de-parser...
Starting malc0de-windows-format-collector...
Starting malware-domain-list-collector...
Starting malware-domain-list-parser...
Starting spamhaus-drop-collector...
Starting spamhaus-drop-parser...
Starting taxonomy-expert...
Starting url2fqdn-expert...
Bot cymru-whois-expert is running.
Bot deduplicator-expert is running.
Bot feodo-tracker-browse-collector is running.
Bot feodo-tracker-browse-parser is running.
Bot file-output is running.
Bot gethostbyname-1-expert is running.
Bot gethostbyname-2-expert is running.
Bot malc0de-parser is running.
Bot malc0de-windows-format-collector is running.
Bot malware-domain-list-collector is running.
Bot malware-domain-list-parser is running.
Bot spamhaus-drop-collector is running.
Bot spamhaus-drop-parser is running.
Bot taxonomy-expert is running.
Bot url2fqdn-expert is running.
Bot Botnet is running.
root@vm07pec4intel:~# intelmqctl status
Bot cymru-whois-expert is running.
Bot deduplicator-expert is running.
Bot feodo-tracker-browse-collector is running.
Bot feodo-tracker-browse-parser is running.
Bot file-output is running.
Bot gethostbyname-1-expert is running.
Bot gethostbyname-2-expert is running.
Bot malc0de-parser is running.
Bot malc0de-windows-format-collector is running.
Bot malware-domain-list-collector is running.
Bot malware-domain-list-parser is running.
Bot spamhaus-drop-collector is running.
Bot spamhaus-drop-parser is running.
Bot taxonomy-expert is running.
Bot url2fqdn-expert is running.
```

```
root@vm07pec4intel:~# intelmqctl list bots
Bot ID: cymru-whois-expert
Description: Cymry Whois (IP to ASN) is the bot responsible to add network information to the events
(BGP, ASN, AS Name, Country, etc..).
Bot ID: deduplicator-expert
Description: Deduplicator is the bot responsible for detection and removal of duplicate messages.
Messages get cached for <redis_cache_ttl> seconds. If found in the cache, it is assumed to be a
duplicate.
Bot ID: feodo-tracker-browse-collector
Description: Generic URL Fetcher is the bot responsible to get the report from an URL.
Bot ID: feodo-tracker-browse-parser
Description: HTML Table Parser is a bot configurable to parse different html table data.
Bot ID: file-output
Description: File is the bot responsible to send events to a file.
Bot ID: gethostbyname-1-expert
Description: fqdn2ip is the bot responsible to parsing the ip from the fqdn.
Bot ID: gethostbyname-2-expert
Description: fqdn2ip is the bot responsible to parsing the ip from the fqdn.
Bot ID: malc0de-parser
Description: Malc0de Parser is the bot responsible to parse the IP Blacklist and either Windows
Format or Bind Format reports and sanitize the information.
Bot ID: malc0de-windows-format-collector
Description:
Bot ID: malware-domain-list-collector
Description: Malware Domain List Collector is the bot responsible to get the report from source of
information.
Bot ID: malware-domain-list-parser
Description: Malware Domain List Parser is the bot responsible to parse the report and sanitize the
information.
Bot ID: spamhaus-drop-collector
Description:
Bot ID: spamhaus-drop-parser
Description: Spamhaus Drop Parser is the bot responsible to parse the DROP, EDROP, DROPv6, and ASN-
DROP reports and sanitize the information.
Bot ID: taxonomy-expert
Description: Taxonomy is the bot responsible to apply the eCSIRT Taxonomy to all events.
Bot ID: url2fqdn-expert
Description: url2fqdn is the bot responsible to parsing the fqdn from the url.

### Se obtiene el fichero event.txt una vez recolectada la información por parte de IntelMQ de
diferentes fuentes y posteriormente enriquecida

root@vm07pec4intel:~# ls -la /opt/intelmq/var/lib/bots/file-output/
total 552
drwxrws--- 2 intelmq intelmq   4096 dic 12 00:21 .
drwxrws--- 3 intelmq intelmq   4096 dic 12 00:13 ..
-rwxrwx--- 1 intelmq intelmq 554615 dic 12 00:42 events.txt

root@vm07pec4intel:~# ls -la /opt/intelmq/var/lib/bots/file-output/
total 6924
drwxrws--- 2 intelmq intelmq    4096 dic 12 00:21 .
drwxrws--- 3 intelmq intelmq    4096 dic 12 00:13 ..
-rwxrwx--- 1 intelmq intelmq 7075321 dic 12 01:04 events.txt

root@vm07pec4intel:~# ls -la /opt/intelmq/var/log/*.log
-rwxrwx--- 1 intelmq intelmq  1889 dic 12 00:41 /opt/intelmq/var/log/cymru-whois-expert.log
-rwxrwx--- 1 intelmq intelmq  2438 dic 12 00:42 /opt/intelmq/var/log/deduplicator-expert.log
-rwxrwx--- 1 intelmq intelmq  2668 dic 12 00:41 /opt/intelmq/var/log/feodo-tracker-browse-
collector.log
-rwxrwx--- 1 intelmq intelmq  4961 dic 12 00:42 /opt/intelmq/var/log/feodo-tracker-browse-parser.log
-rwxrwx--- 1 intelmq intelmq  1544 dic 12 00:41 /opt/intelmq/var/log/file-output.log
-rwxrwx--- 1 intelmq intelmq  1916 dic 12 00:42 /opt/intelmq/var/log/gethostbyname-1-expert.log
-rwxrwx--- 1 intelmq intelmq  1916 dic 12 00:42 /opt/intelmq/var/log/gethostbyname-2-expert.log
-rwxrwx--- 1 intelmq intelmq 23229 dic 12 00:41 /opt/intelmq/var/log/intelmqctl.log
-rwxrwx--- 1 intelmq intelmq  1716 dic 12 00:41 /opt/intelmq/var/log/malc0de-parser.log
-rwxrwx--- 1 intelmq intelmq  2695 dic 12 00:41 /opt/intelmq/var/log/malc0de-windows-format-
collector.log
-rwxrwx--- 1 intelmq intelmq  2671 dic 12 00:41 /opt/intelmq/var/log/malware-domain-list-
collector.log
-rwxrwx--- 1 intelmq intelmq  1993 dic 12 00:41 /opt/intelmq/var/log/malware-domain-list-parser.log
-rwxrwx--- 1 intelmq intelmq  2482 dic 12 00:41 /opt/intelmq/var/log/spamhaus-drop-collector.log
-rwxrwx--- 1 intelmq intelmq  2159 dic 12 00:41 /opt/intelmq/var/log/spamhaus-drop-parser.log
-rwxrwx--- 1 intelmq intelmq  2318 dic 12 00:42 /opt/intelmq/var/log/taxonomy-expert.log
-rwxrwx--- 1 intelmq intelmq  2318 dic 12 00:42 /opt/intelmq/var/log/url2fqdn-expert.log
```

Instalación de IntelMQ Manager

A continuación se inicia el proceso de instalación de IntelMQ Manager, portal web para la gestión de IntelMQ.

```
root@vm07pec4intel:~# apt-get install git apache2 php libapache2-mod-php
Leyendo lista de paquetes... Hecho
```

```
Creando árbol de dependencias
Leyendo la información de estado... Hecho
git ya está en su versión más reciente (1:2.20.1-2ubuntu1.19.10.1).
fijado git como instalado manualmente.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapache2-mod-php7.3 libapr1 libaprutil1 libaprutil1-dbd-
sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0 php-common php7.3 php7.3-cli
  php7.3-common php7.3-json php7.3-opcache php7.3-readline ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser php-pear openssl-
blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php libapache2-mod-php7.3 libapr1
libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0 php
  php-common php7.3 php7.3-cli php7.3-common php7.3-json php7.3-opcache php7.3-readline ssl-cert
0 actualizados, 21 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 5.792 kB de archivos.
Se utilizarán 25,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S

root@vm07pec4intel:~# git clone https://github.com/certtools/intelmq-manager.git /tmp/intelmq-
manager
Cloning into '/tmp/intelmq-manager'...
remote: Enumerating objects: 53, done.
remote: Counting objects: 100% (53/53), done.
remote: Compressing objects: 100% (40/40), done.
remote: Total 2948 (delta 15), reused 33 (delta 13), pack-reused 2895
Receiving objects: 100% (2948/2948), 4.99 MiB | 5.87 MiB/s, done.
Resolving deltas: 100% (1854/1854), done.
root@vm07pec4intel:~# cp -R /tmp/intelmq-manager/intelmq-manager/* /var/www/html/
root@vm07pec4intel:~# chown -R www-data.www-data /var/www/html/
root@vm07pec4intel:~# usermod -a -G intelmq www-data
root@vm07pec4intel:~# mkdir /opt/intelmq/etc/manager/
root@vm07pec4intel:~# touch /opt/intelmq/etc/manager/positions.conf
root@vm07pec4intel:~# chgrp www-data /opt/intelmq/etc/*.conf
/opt/intelmq/etc/manager/positions.conf
root@vm07pec4intel:~# chmod g+w /opt/intelmq/etc/*.conf /opt/intelmq/etc/manager/positions.conf
root@vm07pec4intel:~# nano /etc/sudoers
```

```
root@vm07pec4intel:/home/usuario# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults env_reset
Defaults mail_badpass
Defaults secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/
snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
www-data ALL=(intelmq) NOPASSWD: /usr/local/bin/intelmqctl
```

```
root@vm07pec4intel:~# sudo -u intelmq /usr/local/bin/intelmqctl
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/intelmq/bin/intelmqctl.py", line 668, in __init__
    logging_level_stream=logging_level_stream)
  File "/usr/local/lib/python3.7/dist-packages/intelmq/lib/utils.py", line 337, in log
    handler = FileHandler("%s/%s.log" % (log_path, name))
  File "/usr/lib/python3.7/logging/__init__.py", line 1087, in __init__
    StreamHandler.__init__(self, self._open())
  File "/usr/lib/python3.7/logging/__init__.py", line 1116, in _open
    return open(self.baseFilename, self.mode, encoding=self.encoding)
PermissionError: [Errno 13] Permission denied: '/opt/intelmq/var/log/intelmqctl.log'

During handling of the above exception, another exception occurred:
```

```
Traceback (most recent call last):
  File "/usr/local/bin/intelmqctl", line 10, in <module>
    sys.exit(main())
  File "/usr/local/lib/python3.7/dist-packages/intelmq/bin/intelmqctl.py", line 1754, in main
    x = IntelMQController(interactive=True)
  File "/usr/local/lib/python3.7/dist-packages/intelmq/bin/intelmqctl.py", line 672, in __init__
    logging_level_stream=logging_level_stream)
  File "/usr/local/lib/python3.7/dist-packages/intelmq/lib/utils.py", line 348, in log
    raise ValueError("Invalid configuration, neither log_path is given nor syslog is used.")
ValueError: Invalid configuration, neither log_path is given nor syslog is used.
root@vm07pec4intel:~# su - intelmq
intelmq@vm07pec4intel:~$ intelmqctl start
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/intelmq/bin/intelmqctl.py", line 668, in __init__
    logging_level_stream=logging_level_stream)
  File "/usr/local/lib/python3.7/dist-packages/intelmq/lib/utils.py", line 337, in log
    handler = FileHandler("%s/%s.log" % (log_path, name))
  File "/usr/lib/python3.7/logging/__init__.py", line 1087, in __init__
    StreamHandler.__init__(self, self._open())
  File "/usr/lib/python3.7/logging/__init__.py", line 1116, in _open
    return open(self.baseFilename, self.mode, encoding=self.encoding)
PermissionError: [Errno 13] Permission denied: '/opt/intelmq/var/log/intelmqctl.log'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/usr/local/bin/intelmqctl", line 10, in <module>
    sys.exit(main())
  File "/usr/local/lib/python3.7/dist-packages/intelmq/bin/intelmqctl.py", line 1754, in main
    x = IntelMQController(interactive=True)
  File "/usr/local/lib/python3.7/dist-packages/intelmq/bin/intelmqctl.py", line 672, in __init__
    logging_level_stream=logging_level_stream)
  File "/usr/local/lib/python3.7/dist-packages/intelmq/lib/utils.py", line 348, in log
    raise ValueError("Invalid configuration, neither log_path is given nor syslog is used.")
ValueError: Invalid configuration, neither log_path is given nor syslog is used.
intelmq@vm07pec4intel:~$ exit
logout
root@vm07pec4intel:/home/usuario# lsof -i -P -n | grep redis
redis-ser  7588           redis    6u  IPv4    47018      0t0  TCP 127.0.0.1:6379 (LISTEN)
redis-ser  7588           redis    7u  IPv6    47019      0t0  TCP [::1]:6379 (LISTEN)
redis-ser  7588           redis    8u  IPv4  2165083      0t0  TCP 127.0.0.1:6379->127.0.0.1:33996
(ESTABLISHED)
redis-ser  7588           redis    9u  IPv4  2165085      0t0  TCP 127.0.0.1:6379->127.0.0.1:33998
(ESTABLISHED)
redis-ser  7588           redis   10u  IPv4  2165088      0t0  TCP 127.0.0.1:6379->127.0.0.1:34000
(ESTABLISHED)
redis-ser  7588           redis   11u  IPv4  2165216      0t0  TCP 127.0.0.1:6379->127.0.0.1:34002
(ESTABLISHED)
redis-ser  7588           redis   12u  IPv4  2165218      0t0  TCP 127.0.0.1:6379->127.0.0.1:34004
(ESTABLISHED)
redis-ser  7588           redis   13u  IPv4  2165221      0t0  TCP 127.0.0.1:6379->127.0.0.1:34006
(ESTABLISHED)
redis-ser  7588           redis   14u  IPv4  2165224      0t0  TCP 127.0.0.1:6379->127.0.0.1:34008
(ESTABLISHED)
redis-ser  7588           redis   15u  IPv4  2165226      0t0  TCP 127.0.0.1:6379->127.0.0.1:34010
(ESTABLISHED)
redis-ser  7588           redis   16u  IPv4  2165228      0t0  TCP 127.0.0.1:6379->127.0.0.1:34012
(ESTABLISHED)
redis-ser  7588           redis   17u  IPv4  2165231      0t0  TCP 127.0.0.1:6379->127.0.0.1:34014
(ESTABLISHED)
redis-ser  7588           redis   18u  IPv4  2165433      0t0  TCP 127.0.0.1:6379->127.0.0.1:34016
(ESTABLISHED)
redis-ser  7588           redis   19u  IPv4  2165436      0t0  TCP 127.0.0.1:6379->127.0.0.1:34018
(ESTABLISHED)
redis-ser  7588           redis   20u  IPv4  2165438      0t0  TCP 127.0.0.1:6379->127.0.0.1:34020
(ESTABLISHED)
redis-ser  7588           redis   21u  IPv4  2165440      0t0  TCP 127.0.0.1:6379->127.0.0.1:34022
(ESTABLISHED)
redis-ser  7588           redis   22u  IPv4  2165442      0t0  TCP 127.0.0.1:6379->127.0.0.1:34024
(ESTABLISHED)
redis-ser  7588           redis   23u  IPv4  2165468      0t0  TCP 127.0.0.1:6379->127.0.0.1:34026
(ESTABLISHED)
redis-ser  7588           redis   24u  IPv4  2165476      0t0  TCP 127.0.0.1:6379->127.0.0.1:34028
(ESTABLISHED)
redis-ser  7588           redis   25u  IPv4  2165549      0t0  TCP 127.0.0.1:6379->127.0.0.1:34030
(ESTABLISHED)
redis-ser  7588           redis   26u  IPv4  2165553      0t0  TCP 127.0.0.1:6379->127.0.0.1:34032
(ESTABLISHED)
redis-ser  7588           redis   27u  IPv4  2428933      0t0  TCP 127.0.0.1:6379->127.0.0.1:34046
(ESTABLISHED)
redis-ser  7588           redis   28u  IPv4  2428939      0t0  TCP 127.0.0.1:6379->127.0.0.1:34048
(ESTABLISHED)
redis-ser  7588           redis   29u  IPv4  2428942      0t0  TCP 127.0.0.1:6379->127.0.0.1:34050
(ESTABLISHED)

root@vm07pec4intel:/home/usuario# lsof -i -P -n | grep intelmq
```

```
intelmq.b 24336         intelmq    4u  IPv4 13447671      0t0  TCP 192.168.29.88:39518-
>151.101.134.49:443 (CLOSE_WAIT)
intelmq.b 24336         intelmq    5u  IPv4  2165552      0t0  TCP 127.0.0.1:34032->127.0.0.1:6379
(ESTABLISHED)
intelmq.b 24336         intelmq    6u  IPv4  2937678      0t0  TCP 127.0.0.1:34086->127.0.0.1:6379
(ESTABLISHED)
intelmq.b 24337         intelmq    4u  IPv4 13448436      0t0  TCP 192.168.29.88:49702-
>104.27.116.104:443 (CLOSE_WAIT)
intelmq.b 24337         intelmq    5u  IPv4  2165087      0t0  TCP 127.0.0.1:34000->127.0.0.1:6379
(ESTABLISHED)
intelmq.b 24337         intelmq    6u  IPv4  2428784      0t0  TCP 127.0.0.1:34050->127.0.0.1:6379
(ESTABLISHED)
intelmq.b 24338         intelmq    4u  IPv4 13447257      0t0  TCP 192.168.29.88:45258-
>143.215.130.61:80 (CLOSE_WAIT)
intelmq.b 24338         intelmq    5u  IPv4  2164588      0t0  TCP 127.0.0.1:33998->127.0.0.1:6379
(ESTABLISHED)
intelmq.b 24338         intelmq    6u  IPv4  2428938      0t0  TCP 127.0.0.1:34048->127.0.0.1:6379
(ESTABLISHED)
intelmq.b 24339         intelmq    4u  IPv4 13448401      0t0  TCP 192.168.29.88:52502-
>104.18.104.225:443 (CLOSE_WAIT)
intelmq.b 24339         intelmq    5u  IPv4  2165082      0t0  TCP 127.0.0.1:33996->127.0.0.1:6379
(ESTABLISHED)
```

Comprobación funcional de IntelMQ Manager

A continuación se muestran las pantallas de la aplicación IntelMQ Manager, el cual se accede a través del navegador introduciendo la url `http://ip/index.php`

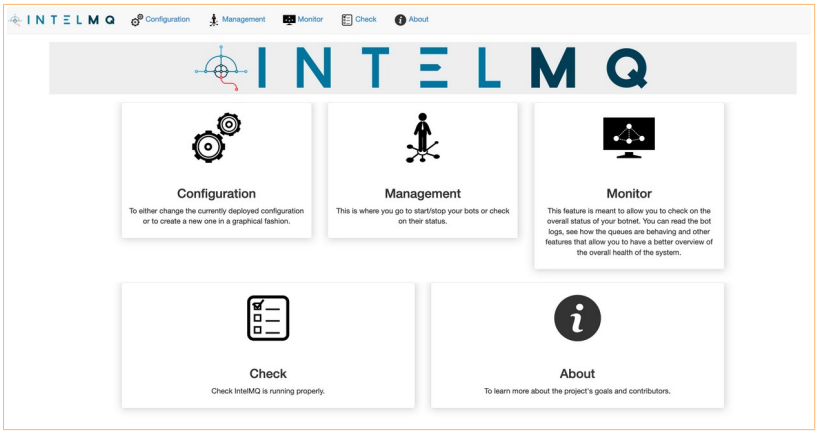

Figura 37: Portal de Inicio IntelMQ Manager

A continuación se describen los apartados referente a la gestión de IntelMQ Mananger mediante el acceso al portal web.

Configuration: donde podemos ver la configuración de nuestro flujo, añadir nodos, añadir colas, editar los parámetros de los nodos actuales y los nuevos, además de iniciar, parar, crear nuevas *botnets* diferentes, etc … en este apartado se muestra el core de manera gráfico de la herramienta. El entorno gráfico facilita enormemente la gestión mediante vistas sencillas.

Figura 38: IntelMQ Manager - Configuration

Al editar alguno de los *bot* nos aparece una ventana con los parámetros de configuración que modificar en base a nuestras necesidades como se observa en la siguiente imagen.

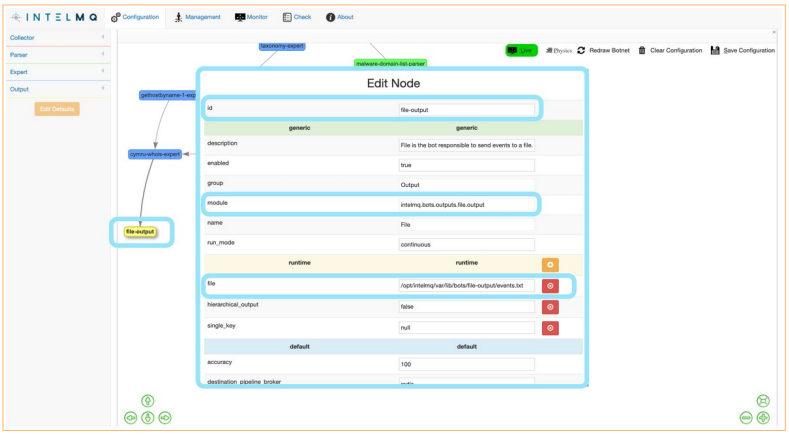

Figura 39: IntelMQ Manager - Configuration - Edit Node

Se debe tener en cuenta que cada *bot* debe conectarse a otro *bot* mediante una cola (*Queue*) para que el flujo tenga sentido. Además, un *bot* puede tener varias colas.

Management: donde se puede ver el estado de la *botnet* (agrupación de todos los nodos o *bots*) además de cada uno de los *bots* (*collector, parser, expert, output*). Si pulsamos sobre uno de los bot, directamente iremos a la pantalla de *Monitor*.

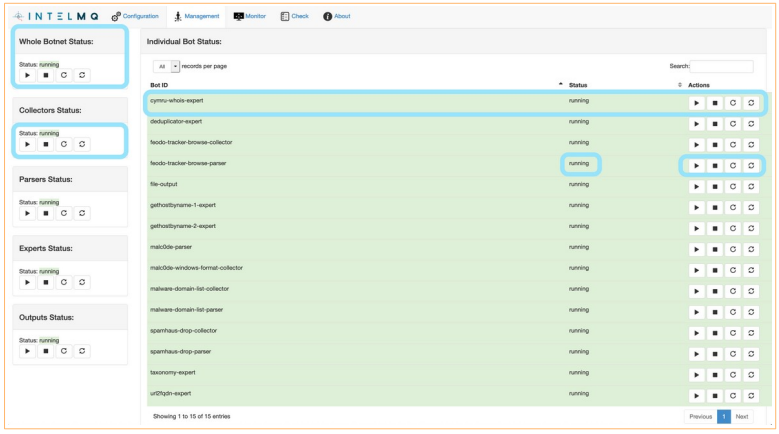

Figura 40: IntelMQ Manager - Manager

Monitor: donde podemos ver el estado de cada *bot*, verificar las colas, realizar comprobaciones de ejecución, de los logs e incluso los parámetros de configuración de modo que si observamos algún error o queremos cambiar algún valor, podemos ir a la pantalla *Configuration* y realizar los cambios oportunos. En las siguientes imágenes se presentan las pantallas de este apartado.

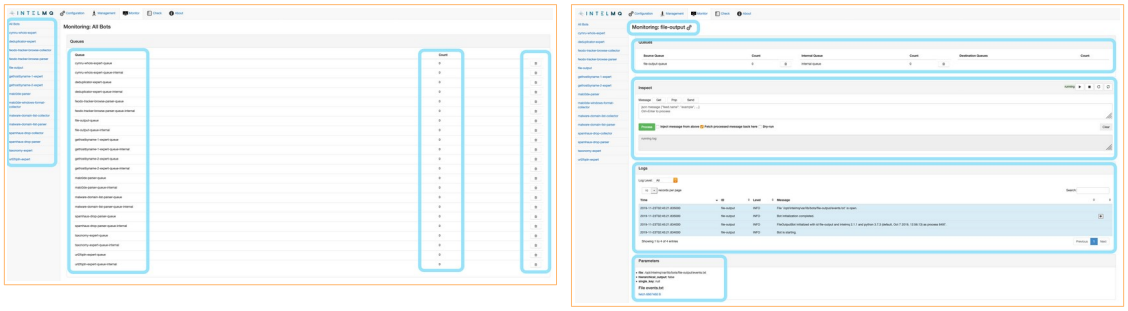

Figura 41: IntelMQ Manager - Monitor

Check / About: en esta apartado vemos la verificación funcional de la aplicación. En el caso de que hay algún error nos mostrará en color rojo el error con instrucciones orientativas del fallo. En *About* nos indica la versión de las aplicaciones.

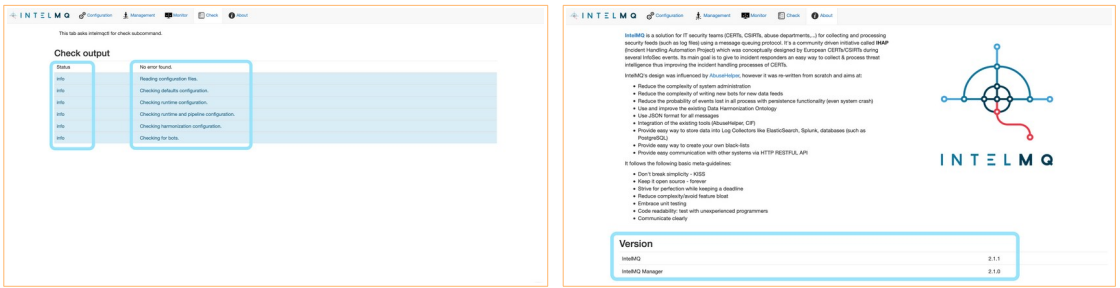

Figura 42: IntelMQ Manager - Check/About

A continuación se describe brevemente el funcionamiento de la *botnet* que hay creada a modo de ejemplo, y que sirve de ayuda para la creación de nuevos *bots* o *botnets*.

Collectors: recoge la información de varias direcciones de internet, entre las que encontramos de tipo fichero, tipo html, etc

- **feodo-tracker-browse** = https://feodotracker.abuse.ch/browse
- **malc0de-windows-format** = https://malc0de.com/bl/BOOT
- **malware-domain-list** = http://www.malwaredomainlist.com/updatescsv.php
- **spamhaus-drop** = https://www.spamhaus.org/drop/drop.txt

Parsers: analizan la información para procesarla según el modelo de datos de cada fuente.

- **feodo-tracker-browse**
- **malc0de**
- **malware-domain-list**

Experts: implementa una mejora de la información a la botnet

- **cymru-whois** = añade información relevante (BGP, ASN, AS Name, Country, etc..)
- **gethostbyname-1** = de ip a fqdn
- **gethostbyname-2** = de ip a fqdn
- **deduplicator** = identifica duplicados y los elimina
- **taxonomy** = normaliza los eventos según "*eCSIRT Taxonomy*"
- **url2fqdn** = analiza la url para pasar a fqdn

Output: se trata de la salida, en este caso un fichero de texto.

- **file-output** = exportación de lista de reputación a fichero `event.txt` ubicado en la ruta `/opt/intelmq/var/lib/bots/file-output/`

Por tanto, mediante **intelMQ** y la *botnet* que hay desplegada en la herramienta obtenemos una lista de ip actualizada diariamente de 3 fuentes diferentes.

- Feodo Tracker
- Malc0de
- Spamhaus

13.3 ANEXO III. Ficheros de configuración

Apartado donde de adjunta los ficheros de configuración de la aplicación Zeek que han sido utilizados para el desarrollo del trabajo y que se consideran relevantes. También se aporta la configuración de Filebeat realiza sobre el host.

Ficheros de configuración de Zeek

```
root@vm07pec4zeek:/home/usuario# cat /opt/zeek/etc/networks.cfg
# List of local networks in CIDR notation, optionally followed by a
# descriptive tag.
# For example, "10.0.0.0/8" or "fe80::/64" are valid prefixes.

10.0.0.0/8          Private IP space
172.16.0.0/12       Private IP space
192.168.0.0/16      Private IP space
```

```
root@vm07pec4zeek:/home/usuario# cat /opt/zeek/etc/node.cfg
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# 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

## Below is an example clustered configuration. If you use this,
## remove the [zeek] node above.

#[logger]
#type=logger
#host=localhost
#
[manager]
type=manager
host=localhost
#
[proxy-1]
type=proxy
host=localhost
#
[worker-1]
type=worker
host=localhost
interface=ens160
lb_method=pf_ring
lb_procs=2
#
[worker-2]
type=worker
host=localhost
interface=ens160
lb_method=pf_ring
lb_procs=2
```

```
root@vm07pec4zeek:/home/usuario# cat /opt/zeek/share/zeek/site/scripts/json-logs.zeek
# Load tuning for JSON log (Time and Parse)
@load tuning/json-logs
redef LogAscii::json_timestamps = JSON::TS_ISO8601;
redef LogAscii::use_json = T;
```

```
root@vm07pec4zeek:/home/usuario# cat /opt/zeek/share/zeek/site/local.zeek
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Apply the default tuning scripts for common tuning settings.
@load tuning/defaults

# Estimate and log capture loss.
@load misc/capture-loss

# Enable logging of memory, packet and lag statistics.
@load misc/stats

# Load the scan detection script.  It's disabled by default because
# it often causes performance issues.
#@load misc/scan

# Detect traceroute being run on the network. This could possibly cause
# performance trouble when there are a lot of traceroutes on your network.
# Enable cautiously.
#@load misc/detect-traceroute
```

```
# Generate notices when vulnerable versions of software are discovered.
# The default is to only monitor software found in the address space defined
# as "local".  Refer to the software framework's documentation for more
# information.
@load frameworks/software/vulnerable

# Detect software changing (e.g. attacker installing hacked SSHD).
@load frameworks/software/version-changes

# This adds signatures to detect cleartext forward and reverse windows shells.
@load-sigs frameworks/signatures/detect-windows-shells

# Load all of the scripts that detect software in various protocols.
@load protocols/ftp/software
@load protocols/smtp/software
@load protocols/ssh/software
@load protocols/http/software
# The detect-webapps script could possibly cause performance trouble when
# running on live traffic.  Enable it cautiously.
#@load protocols/http/detect-webapps

# This script detects DNS results pointing toward your Site::local_nets
# where the name is not part of your local DNS zone and is being hosted
# externally.  Requires that the Site::local_zones variable is defined.
@load protocols/dns/detect-external-names

# Script to detect various activity in FTP sessions.
@load protocols/ftp/detect

# Scripts that do asset tracking.
@load protocols/conn/known-hosts
@load protocols/conn/known-services
@load protocols/ssl/known-certs

# This script enables SSL/TLS certificate validation.
@load protocols/ssl/validate-certs

# This script prevents the logging of SSL CA certificates in x509.log
@load protocols/ssl/log-hostcerts-only

# Uncomment the following line to check each SSL certificate hash against the ICSI
# certificate notary service; see http://notary.icsi.berkeley.edu .
# @load protocols/ssl/notary

# If you have GeoIP support built in, do some geographic detections and
# logging for SSH traffic.
@load protocols/ssh/geo-data
# Detect hosts doing SSH bruteforce attacks.
@load protocols/ssh/detect-bruteforcing
# Detect logins using "interesting" hostnames.
@load protocols/ssh/interesting-hostnames

# Detect SQL injection attacks.
@load protocols/http/detect-sqli

#### Network File Handling ####

# Enable MD5 and SHA1 hashing for all files.
@load frameworks/files/hash-all-files

# Detect SHA1 sums in Team Cymru's Malware Hash Registry.
@load frameworks/files/detect-MHR

# Extend email alerting to include hostnames
@load policy/frameworks/notice/extend-email/hostnames

# Uncomment the following line to enable detection of the heartbleed attack. Enabling
# this might impact performance a bit.
# @load policy/protocols/ssl/heartbleed

# Uncomment the following line to enable logging of connection VLANs. Enabling
# this adds two VLAN fields to the conn.log file.
# @load policy/protocols/conn/vlan-logging

# Uncomment the following line to enable logging of link-layer addresses. Enabling
# this adds the link-layer address for each connection endpoint to the conn.log file.
```

```
# @load policy/protocols/conn/mac-logging
# Load policy for JSON output
@load scripts/json-logs
```

Ficheros de configuración de Filebeat (Host Zeek)

```
root@vm07pec4zeek:/home/usuario# cat /etc/filebeat/filebeat.yml
###################### Filebeat Configuration Example #########################

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

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

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

filebeat.inputs:

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

- type: log

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

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    ## Logs from Zeek
    - /opt/zeek/logs/current/*.log
    ## Logs from IntelMQ (parsed)
    #- /opt/intelmq/parsed.log
    #- /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*

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

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

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

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

  ### Multiline options

  # Multiline can be used for log messages spanning multiple lines. This is common
  # for Java Stack Traces or C-Line Continuation

   # The regexp Pattern that has to be matched. The example pattern matches all lines
starting with [
  #multiline.pattern: ^\[

   # Defines if the pattern set under pattern should be negated or not. Default is
false.
  #multiline.negate: false
```

```
   # Match can be set to "after" or "before". It is used to define if lines should be
append to a pattern
  # that was (not) matched before or after or as long as a pattern is not matched based
on negate.
   # Note: After is the equivalent to previous and before is the equivalent to to next
in Logstash
  #multiline.match: after


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

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

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

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

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

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

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

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

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

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


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

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

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

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

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

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

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

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

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

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

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

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

#-------------------------- Elasticsearch output ------------------------------
#output.elasticsearch:
  # Array of hosts to connect to.
 # hosts: ["localhost:9200"]

  # Optional protocol and basic auth credentials.
  #protocol: "https"
  #username: "elastic"
  #password: "changeme"

#----------------------------- Logstash output --------------------------------
output.logstash:
  # The Logstash hosts
  #hosts: ["localhost:5044"]
  hosts: ["192.168.29.87:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

#================================ Processors =====================================

# Configure processors to enhance or manipulate events generated by the beat.

processors:
  - add_host_metadata: ~
  - add_cloud_metadata: ~
  - add_docker_metadata: ~
  - add_kubernetes_metadata: ~

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

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

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

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

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

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

```
#monitoring.cluster_uuid:

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

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

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

Ficheros de configuración de Elasticsearch

```
root@vm07pec4elk:/home/transferencia# cat /etc/elasticsearch/elasticsearch.yml
# ======================== Elasticsearch Configuration =========================
#
# NOTE: Elasticsearch comes with reasonable defaults for most settings.
#       Before you set out to tweak and tune the configuration, make sure you
#       understand what are you trying to accomplish and the consequences.
#
# The primary way of configuring a node is via this file. This template lists
# the most important settings you may want to configure for a production cluster.
#
# Please consult the documentation for further information on configuration options:
# https://www.elastic.co/guide/en/elasticsearch/reference/index.html
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: ClusterElkZeekService
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
#node.name: node-1
#
# Add custom attributes to the node:
#
#node.attr.rack: r1
#
# ----------------------------------- Paths ------------------------------------
#
# Path to directory where to store the data (separate multiple locations by comma):
#
path.data: /var/lib/elasticsearch
#
# Path to log files:
#
path.logs: /var/log/elasticsearch
#
# ----------------------------------- Memory -----------------------------------
#
# Lock the memory on startup:
#
#bootstrap.memory_lock: true
#
# Make sure that the heap size is set to about half the memory available
# on the system and that the owner of the process is allowed to use this
# limit.
#
# Elasticsearch performs poorly when the system is swapping the memory.
#
# ---------------------------------- Network -----------------------------------
#
# Set the bind address to a specific IP (IPv4 or IPv6):
#
#network.host: 192.168.0.1
network.host: 0.0.0.0
```

```
#
# Set a custom port for HTTP:
#
#http.port: 9200
http.port: 9200
#
# For more information, consult the network module documentation.
#
# ---------------------------------- Discovery -----------------------------------
#
# Pass an initial list of hosts to perform discovery when this node is started:
# The default list of hosts is ["127.0.0.1", "[::1]"]
#
#discovery.seed_hosts: ["host1", "host2"]
discovery.seed_hosts: ["host1"]
#
# Bootstrap the cluster using an initial set of master-eligible nodes:
#
#cluster.initial_master_nodes: ["node-1", "node-2"]
cluster.initial_master_nodes: ["node-1"]
#
#  For more information, consult the discovery and cluster formation module
documentation.
#
# ---------------------------------- Gateway -----------------------------------
#
# Block initial recovery after a full cluster restart until N nodes are started:
#
#gateway.recover_after_nodes: 3
#
# For more information, consult the gateway module documentation.
#
# ---------------------------------- Various -----------------------------------
#
# Require explicit names when deleting indices:
#
action.destructive_requires_name: true
xpack.security.enabled : false
xpack.watcher.enabled : true
# Posible problema de rotado de logs.
# action.auto_create_index: .watches,.triggered_watches,.watcher-history*

xpack.notification.email.account:
    gmail_account:
        profile: gmail
        smtp:
            auth: true
            starttls.enable: true
            host: smtp.gmail.com
            port: 587
            user: not.watcher.mail@gmail.com
```

Ficheros de configuración de Kibana

```
root@vm07pec4elk:/home/transferencia# cat /etc/kibana/kibana.yml
# Kibana is served by a back end server. This setting specifies the port to use.
#server.port: 5601
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host
names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to
connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
#server.host: "localhost"
server.host: "0.0.0.0"

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

# Specifies whether Kibana should rewrite requests that are prefixed with
```

```
# `server.basePath` or require that they are rewritten by your reverse proxy.
# This setting was effectively always `false` before Kibana 6.3 and will
# default to `true` starting in Kibana 7.0.
#server.rewriteBasePath: false

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

# 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
server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the
host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "home"

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

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

# Optional settings that provide the paths to the PEM-format SSL certificate and key
files.
# These files validate that your Elasticsearch backend uses the same 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
certificate
# authority for your Elasticsearch instance.
#elasticsearch.ssl.certificateAuthorities: [ "/path/to/your/CA.pem" ]

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

# Time in milliseconds to wait for Elasticsearch to respond to pings. Defaults to the
value of
# the elasticsearch.requestTimeout setting.
#elasticsearch.pingTimeout: 1500

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

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

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

```
configuration.
#elasticsearch.customHeaders: {}

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

# Time in milliseconds to wait for Elasticsearch at Kibana startup before retrying.
#elasticsearch.startupTimeout: 5000

# 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
# and all requests.
#logging.verbose: false

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

# Specifies locale to be used for all localizable strings, dates and number formats.
# Supported languages are the following: English - en , by default , Chinese - zh-CN .
#i18n.locale: "en"
```

Ficheros de configuración de Nginx

```
root@vm07pec4elk:/home/transferencia# cat /etc/nginx/sites-available/kibana
server {
    listen 80 default_server;
    server_name _;
    return 301 https://$server_name$request_uri;
} server {
    listen 443 default_server ssl http2;
    server_name _;

    ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem;
    ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key;
    ssl_session_cache shared:SSL:10m;
    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/htpasswd.kibana;
    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
} }
```

Ficheros de configuración de Logstash (tag = zeek)

```
root@vm07pec4elk:/home/transferencia# cat /etc/logstash/conf.d/01-input-beats.conf
##################################################
# Inputs are used to ingest logs from remote logging clients
##################################################
input {
  # Ingest logs that match the Beat template
  beats {
    # Accept connections on port 5044
```

```
    port => 5044
    codec => "json"
    }
 }
```

```
root@vm07pec4elk:/home/transferencia# cat /etc/logstash/conf.d/11-filter-beats.conf
###################################################################
# Filters are used to transform and modify the logs
###################################################################
filter {
# Only apply these transformations to logs that contain the "zeek_filebeat" tag
    if "zeek" in [tags] {
         translate {
          field => "[id.resp_h]"
          destination => "malicious_IP"
          dictionary_path => "/home/transferencia/events.csv"
          override => true

         }
    }
 }
```

```
root@vm07pec4elk:/home/transferencia# cat /etc/logstash/conf.d/21-elasticsearch-output.conf
###################################################################
# Outputs take the logs and output them to a long term storage
###################################################################
output {
  # Send logs that contain the zeek tag too
  if "zeek" in [tags] {
    # Outputting logs to elasticsearch
    elasticsearch {
      # ES host to send logs too
      hosts => ["http://localhost:9200"]
      # Index to store data in
      index => "filebeat-zeek-%{+YYYY.MM.dd}"
      }
       stdout {
       codec => rubydebug
       }
   }
 }
```

Ficheros de configuración de Logstash (tag = intelmq)

```
root@vm07pec4elk:/home/transferencia# cat /etc/logstash/conf.d/intelmq.conf
###################################################
# Inputs are used to ingest logs from remote logging clients
###################################################
input {
  # Ingest logs that match the Beat template - INTELMQ
  beats {
    # Accept connections on port 5045
    port => 5045
    }
}
###################################################################
# Filters are used to transform and modify the logs
###################################################################
filter {
# Only apply these transformations to logs that contain the "intelmq" tag
    if "intelmq" in [tags] {
        # Extract the json into Key value pairs
        json {
            source => "message"
        }
       # Removee fields name
        mutate {
        remove_field => ["message"]
        }
    }
}
###################################################################
# Outputs take the logs and output them to a long term storage
```

```
#################################################################

output {
  # Send logs that contain the intelmq tag too
  if "intelmq" in [tags] {
    # Outputting logs to elasticsearch
    elasticsearch {
      # ES host to send logs too
      hosts => ["http://localhost:9200"]
      # Index to store data in
      index => "filebeat-intelmq-%{+YYYY.MM.dd}"
    }
} }
```

Ficheros de configuración de watcher

Alta de watcher "alarma_ip_maliciosa_ultimo_5m"

```
PUT _watcher/watch/alarma_ip_maliciosa_ultimo_5m
{
  "trigger": {
    "schedule": {
      "interval": "30s"
    }
  },
  "input": {
    "search": {
      "request": {
        "search_type": "query_then_fetch",
        "indices": [
          "filebeat*"
        ],
        "rest_total_hits_as_int": true,
        "body": {
          "query": {
            "bool": {
              "must": [
                {
                  "query_string": {
                    "query": "malicious_IP"
                  }
                },
                {
                  "range": {
                    "@timestamp": {
                      "gte": "now-5m"
                    }
                  }
                }
              ]
            }
          },
          "_source": [
            "message"
          ],
          "sort": [
            {
              "@timestamp": {
                "order": "desc"
              }
            }
          ]
        }
      }
    }
  },
  "condition": {
    "compare": {
      "ctx.payload.hits.total": {
        "gt": 0
      }
    }
  },
  "actions": {
```

```
    "log": {
      "logging": {
        "level": "info",
         "text": "ALERTA. [{{ctx.metadata.name}}] Identificada Acceso IP Maliciosa. Se
ha excedido el umbral: {{ctx.payload.hits.total}}"
      }
    },
    "email_1": {
      "email": {
        "profile": "standard",
        "to": [
          "not.watcher.mail@gmail.com"
        ],
        "subject": "ALERTA [{{ctx.metadata.name}}] - Acceso a IP Maliciosa",
        "body": {
              "text": "Hola, \nSe ha detectado el acceso a una IP identificada como
maliciosa dentro la Red de la Organización. \n N° de accesos:
{{ctx.payload.hits.total}}. Se ha configurado el umbral 0.\n Saludos cordinales.\n
Notificación automática elasticsearch (watcher)"
        }
      }
    }
  },
  "metadata": {
    "xpack": {
      "type": "json"
    },
    "name": "Alarma IP MALICIOSA detectada "
  }
}
```

Ejecución de watcher "alarma_ip_maliciosa_ultimo_5m"

```
POST _watcher/watch/alarma_ip_maliciosa_ultimo_5m/_execute
```

Eliminación de watcher "alarma_ip_maliciosa_ultimo_5m"

```
DELETE _watcher/watch/alarma_ip_maliciosa_ultimo_5m
```

Script convertidor JSON a CSV

```
root@vm07pec4intel:/home/usuario# cat /home/usuario/json2csvrun/json2csv.py
import json

file_json = '/opt/intelmq/var/lib/bots/file-output/events.txt'
file_csv = '/home/usuario/exchangefiles/events.csv'

with open(file_json, "r") as f:
    lines = f.readlines()

data_list = (json.loads(line) for line in lines)
with open(file_csv, "w") as f:
    for data in data_list:
        if 'source.ip' in data:
            f.write(f'{data["source.ip"]},malicious_IP\n')
```

Tarea programada en host IntelMQ

```
root@vm07pec4intel:/home/usuario# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
```

```
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
11 * * * * python3 /home/usuario/json2csvrun/json2csv.py
12 * * * * scp /home/usuario/exchangefiles/* intelip@192.168.29.87:/home/transferencia/
```