Implementación de un sistema de detección de intrusos IDS mediante la inspección del tráfico a través de la red

Miguel Ordóñez Sánchez

Análisis de datos

Nombre Tutor/a de TF
Joan Caparrós Ramírez

Profesor/a responsable de la asignatura
Andreu Pere Isern Deyà

Fecha Entrega 13/6/2023

Universitat Oberta
de Catalunya

FICHA DEL TRABAJO FINAL

Título del trabajo:	**Implementación de un sistema de detección de intrusos IDS mediante la inspección del tráfico a través de la red**
Nombre del autor:	*Miguel Ordóñez Sánchez*
Nombre del consultor/a:	*Joan Caparrós Ramírez*
Nombre del PRA:	*Andreu Pere Isern Deyà*
Fecha de entrega (mm/aaaa):	*06/2023*
Titulación o programa:	Máster Universitario en Ciberseguridad y Privacidad
Área del Trabajo Final:	*Análisis de datos*
Idioma del trabajo:	*Castellano*
Palabras clave	*IDS, raspberry, SIEM*

Resumen del Trabajo

El objetivo de este Trabajo final de Máster es la implementación de un sistema de detección de intrusos (IDS) mediante la inspección del tráfico de una red doméstica, utilizando una raspberry Pi que permita capturar y analizar el trafico por medio de una solución de bajo coste. Por medio de la herramienta Suricata se detectará el tráfico sospechoso y se generaran alertas en tiempo real utilizando un sistema SIEM que analizará y explotará los logs que el IDS genera. Como resultado de la investigación, se implementará un dashboard que permitirá visualizar las conexiones y alertas generades por el IDS y el SIEM. Pudiendo tener una mayor compresión del tráfico de la red y posibles amenazas en tiempo real.

Este TFM ofrece buscar una solución de bajo coste, para la detección de intrusos en una red doméstica mediante soluciones open source y de bajo coste.

Abstract

The aim of this Master's thesis is to implement an Intrusion Detection System (IDS) by inspecting the traffic of a home network, using a raspberry Pi to capture and analyze the traffic through a low-cost solution. Suspicious traffic will be detected using the Suricata tool and real-time alerts will be generated using a SIEM system that analyzes and exploits the logs generated by the IDS. As a result of the research, a dashboard will be implemented that allows visualization of the connections and alerts generated by the IDS and the SIEM, providing a better understanding of network traffic and possible threats in real-time.

This thesis aims to provide a low-cost solution for detecting intrusions in a home network using open-source and low-cost solutions.

Índice

Lista de figuras

Lista de tablas

1. Introducción

1.1. Contexto y justificación del Trabajo

La seguridad en las redes domésticas es un problema que cada vez más preocupante a medida que aumenta el número de dispositivos que se conectan en ellas y se transfieren datos sensibles y personales por la red.

En la mayoría de las casas, los dispositivos se conectan a internet de forma automática y no solemos configurar la seguridad adecuada, lo que convierte que la red sea vulnerable a ataques o intrusiones. Estos dispositivos tampoco suelen estar diseñados con la seguridad en mente, lo que suelen ser propensos a sufrir ataques.

Por ejemplo, un atacante puede explotar vulnerabilidades en un dispositivo normal o de tipo IoT para acceder a la red de casa y, a partir de ahí, obtener acceso a otros dispositivos o usuarios, pudiendo obtener datos sensibles y privados que transferimos por la red. Pudiendo obtener información de salud, financiera o datos privados que puedan llevar a sufrir fraudes, extorsiones o robos de nuestra identidad.

Estos ataques son cada vez más frecuentes en las redes domésticas. Por lo que es esencial que se implementen medidas de seguridad adecuadas para proteger las redes domésticas. La implementación de un sistema de detección de intrusos (IDS) en una red doméstica puede ser una solución efectiva para mejorar la seguridad de la red.

Un IDS es un software o dispositivo de hardware que se encarga de analizar el tráfico de la red y detectar cualquier comportamiento malicioso. Nos detecta y alterna de cualquier intento de intrusión, pudiendo tomar medidas para mitigar el riesgo y proteger nuestra información. Su implementación suele necesitar de una instalación de hardware como puede ser un Raspberry Pi, como punto de acceso de bajo coste y una planificación de una estructura de red que permita analizar las conexiones entrantes y salientes de internet. Además, se pueden utilizar software de análisis como el stack de ELK, que analiza los y ofrece un cuadro de mandos o dashboard de las conexiones y avisos ante las intrusiones o el tráfico malintencionado.

En resumen, la implementación de un IDS en una red doméstica puede ser una solución efectiva para mejorar la seguridad de la red y proteger los datos personales y profesionales de los usuarios. Al detectar y alertar sobre comportamientos malintencionados, los usuarios pueden tomar medidas para mitigar el riesgo y proteger sus dispositivos y datos. En este trabajo fin de máster se centrará en la implementación de un sistema de detección de intrusos en una red domestica utilizando un presupuesto de bajo coste que permita llevar su implementación a la gran mayoría de hogares.

1.2. Objetivos del Trabajo

Para poder medir el éxito del trabajo final de máster se han definido unos objetivos primarios a cumplir que constituirán el producto mínimo viable (MVP) del proyecto y otros secundarios que dada su complejidad podrían quedar fuera del alcance del proyecto.

A continuación se detallan cada uno de ellos:

Objetivos primarios:

- Realizar un análisis de las amenazas y riesgos existentes en la red doméstica.
- Investigar, seleccionar herramientas y tecnologías adecuadas para implementarlo de **bajo coste.**
- Configurar y poner en marcha el sistema de detección de intrusos **para mejorar la ciberseguridad de la red doméstica.**
- Implementar y configurar las herramientas básicas para el monitoreo y análisis del tráfico de la red doméstica de una forma **ágil.**
- Desarrollar un dashboard de conexiones y alertas en la suite ELK que permita visualizar y monitorear el estado de la red en tiempo real de **forma sencilla**.

Objetivos secundarios:

- Investigar que mejoras se podrían añadir al sistema IDS aplicando soluciones de Machine Learning o IA que permita ser más reactivo durante la detección de intrusiones.
- Configurar el monitoreo y análisis del tráfico de la red doméstica para aplicaciones específicas con el objetivo de **controlar el uso de la red** en una red doméstica.
- Realizar pruebas de funcionamiento para comprobar su eficacia como sistema de detención de intrusos.
- Documentar la implementación del IDS y su configuración para que sea fácil de seguir y pueda ser replicado fácilmente por un usuario medio.
- Realizar un video de presentación del proyecto para demostrar su eficacia y utilidad.
- Revisar soluciones complementarias al IDS propuesto, como podría ser la solución ZEEK.

1.3. Enfoque y método seguido

El enfoque de este trabajo es principalmente de investigación e implementación. Primero se busca investigar las mejores prácticas y tecnologías de seguridad en redes domésticas, con el fin de detectar intrusiones. Para desarrollar e implementar una solución de detección de intrusos de bajo coste para una red doméstica.

La metodología utilizada para la planificación y gestión del trabajo será una adaptación de las metodologías de desarrollo ágil SCRUM / Kanban.

El trabajo se divide en varios hitos muy marcados, siguiendo la propuesta de entregas del trabajo.

- Hito 1: PEC 1. Plan de trabajo
- Hito 2: PEC 2. Investigación
- Hito 3: PEC 3. Instalación y parametrización de la solución
- Hito 4: PEC 4. Memoria final
- Hito 5: Presentación en vídeo
- Hito 6: Defensa del TFM

Al finalizar cada hito se realiza un retrospectiva juntamente con el profesor, se analiza las tareas que han quedado pendientes y se decide cuales están bien cerradas y cuales deben de mejorarse y llevarse al siguiente hito.

Para la gestión de tareas y su seguimiento se utiliza una herramienta online con un tablero Kanban, que permite ver en todo momento el estado de las tareas. Estas tareas pasan por cinco estados:

- Abrir: Son las tareas pendientes por empezar
- En curso: tareas que actualmente están trabajándose.
- En espera: Tareas que dependen de otras para finalizar y están bloqueada.
- En revisión: Tareas pendientes de revisar.
- Cerrado: Tareas cerradas.

En la ilustración 1 se puede ver una apreciar el tablero Kanban utilizado.

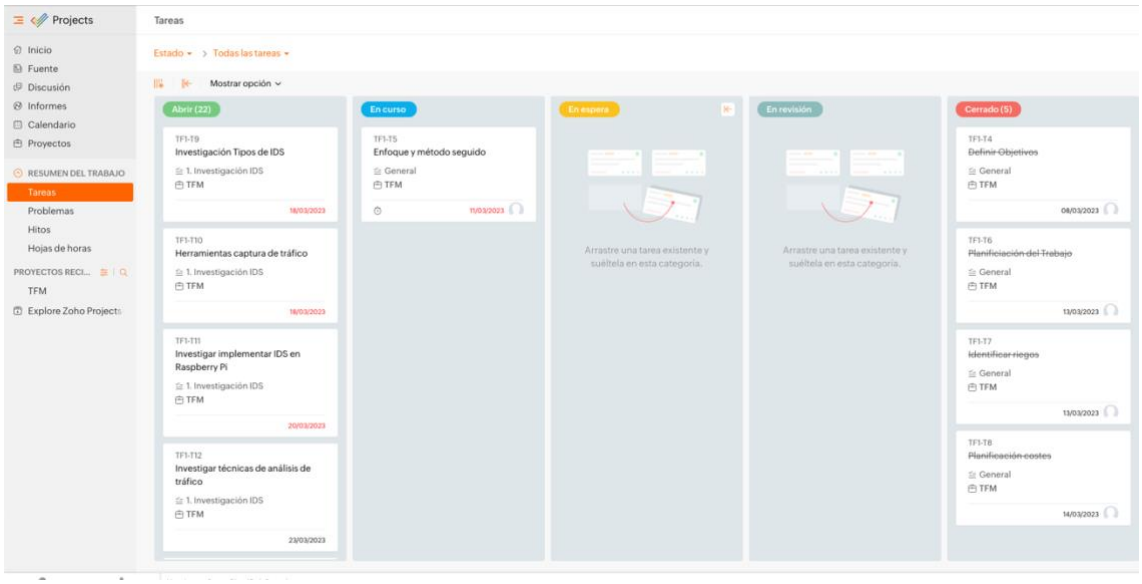


Ilustración 1 Kanban de tareas (Fuente: Elaboración propia)

La herramienta elegida para la gestión del proyecto ha estado Zoho Projects, una herramienta SASS con opción gratuita que ha permitido una gestión de proyecto tal como se desea.

Una gestión de proyecto ágil marcada por hitos, que permita una visualización y seguimiento de la planificación por medio de un diagrama de Gantt y a la vez permite seguimiento de tareas por tableros Kanban personalizable. Con un tablero personal como en la ilustración 2.

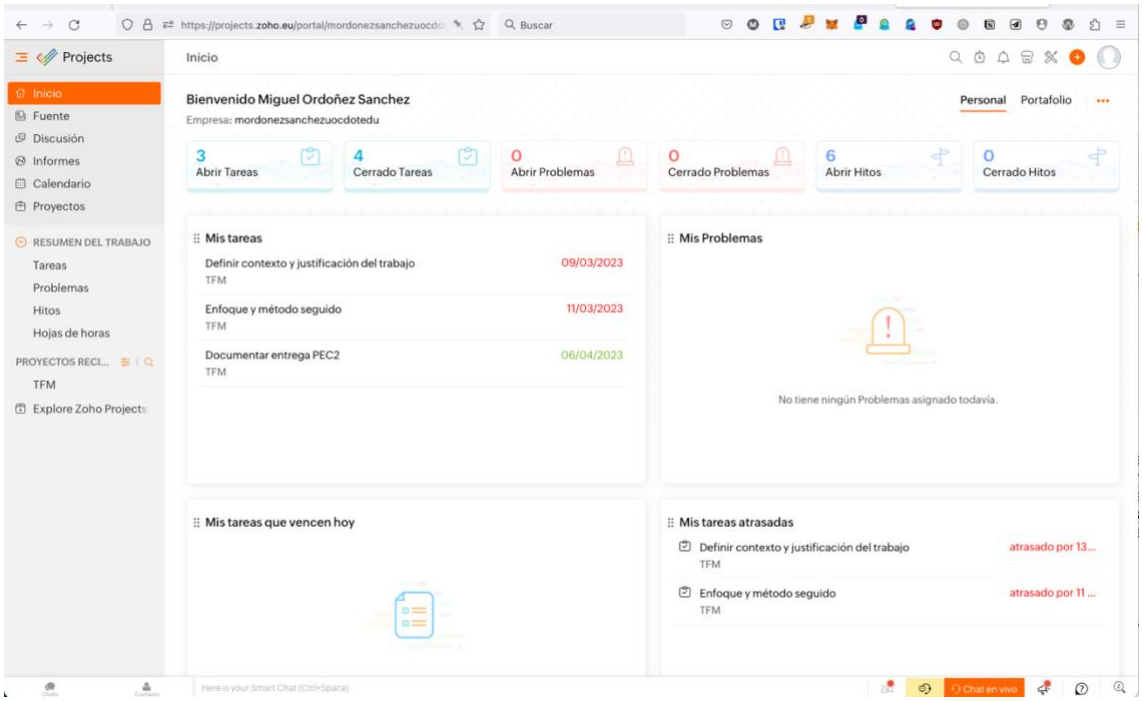


Ilustración 2 Zoho Projects – Dashboard (Fuente: Elaboración propia)

1.4. Planificación del trabajo

Para la correcta planificación y estimación del trabajo se ha realizado una planificación siguiendo los hitos comentados anteriormente y definiendo una a una las tareas previstas.
Para cada tarea se le ha asignado un código de tarea, una fecha de inicio, una fecha de fin y una pre-estimación en horas. Es importante esta estimación para poder planificar correctamente y prever que el trabajo no se desvíe de las **300 horas** iniciales previstas de dedicación.

A continuación, se detalla cada una de las tareas previstas:

Hito 1: Plan de trabajo

TF1-T29 Definir contexto y justificación del trabajo

Identificar el contexto en el que se desarrolla el TFM y la razón por la cual se lleva a cabo. Justificando la necesidad, la importante, los beneficios y/o problemas que se espera resolver de él.

TF1-T9 Definir Objetivos

Los resultados concretos que se espera lograr al final del TFM. Con objetivos claros, específicos, medibles y de modo que puedan ser alcanzados en el tiempo establecido.

TF1-T10 Enfoque y método seguido

Definir la metodología utilizada para llevar a cabo el TFM y el enfoque que seguido. Adecuar la metodología en función a los objetivos del proyecto.

TF1-T11 Planificación del trabajo

Establecer una planificación de trabajo, incluyendo los plazos, las tareas y los hitos necesarios para alcanzar los objetivos definidos. Planificando en base a una situación realista y los recursos disponibles.

TF1-T12 Identificar riegos

Identificar los riesgos que pueden surgir durante el TFM y establecer soluciones para mitigarlos. Es importante identificar los riesgos y prepararse antes de que sucedan para solucionarlos eficazmente en el caso que ocurran.

TF1-T13 Planificación costes

Realizar un presupuesto realista del proyecto que incluya los costes asociados a los materiales, recursos humanos, los equipos y cualquier otro gasto necesario para llevar a cabo el TFM.

Hito 2: Investigación

TF1-T14 Investigación Tipos de IDS

Investigar los diferentes tipos de IDS disponibles en el mercado, identificando sus características, ventajas y desventajas. Se busca encontrar el tipo de IDS que mejor se adapte a las necesidades del proyecto.

TF1-T15 Herramientas captura de tráfico

Investigar y evaluar diferentes herramientas de captura de tráfico de red que permiten capturar y analizar el tráfico de la red. Se exploran sus características, fortalezas y debilidades, así como su compatibilidad con los sistemas de detección de intrusos y las técnicas de análisis de tráfico.

TF1-T16 Investigar implementar IDS en Raspberry Pi

Investigar cómo implementar un sistema de detección de intrusos (IDS) en una Raspberry Pi. Se exploran los requisitos de hardware y software necesarios, las técnicas de configuración y las opciones de personalización para adaptar el IDS a las necesidades específicas del proyecto.

TF1-T21 Técnicas de análisis de logs o dashboards

Investigan las diferentes técnicas y herramientas disponibles para el análisis de los registros generados por el IDS y el SIEM, con el objetivo de poder detectar patrones y comportamientos anómalos. Se busca identificar las técnicas y herramientas más efectivas para el análisis de logs y la creación de dashboards.

TF1-T22 Técnicas de detección comportamientos malintencionados

Investigar las diferentes técnicas utilizadas para detectar comportamientos malintencionados en la red, identificando las herramientas y técnicas más efectivas para la detección de intrusos.

TF1-T23 Documentar entrega PEC2

TF1-T24 Como integrar el IDS con SIEM

Investigar las diferentes técnicas y herramientas que permiten la integración del IDS con el SIEM, para lograr una gestión más eficiente de los registros y alertas generados por el IDS. Se busca encontrar la mejor forma de integrar ambas herramientas.

TF1-T25 Investigar técnicas de análisis de tráfico

Investigar y evalúar diferentes técnicas de análisis de tráfico de red que permitan identificar patrones y comportamientos anómalos en la red. Explorar las técnicas de análisis de flujo de red, análisis de paquetes, análisis de logs, entre otras. Se evalúan sus características, fortalezas y debilidades, así como su compatibilidad con los sistemas de detección de intrusos y las herramientas de captura de tráfico de red.

TF1-T26 Investigar las herramientas SIEM disponibles

Investigar las herramientas SIEM disponibles en el mercado, identificando sus características, ventajas y desventajas. Se busca encontrar la herramienta que mejor se adapte a las necesidades del proyecto.

Hito 3: Parametrización

TF1-T18 Instalación Software SIEM

Llevar a cabo la instalación del software del SIEM (Security Information and Event Management) que se utilizará para recopilar, analizar y correlacionar los eventos y alertas generados por el sistema de detección de intrusos (IDS) y otros dispositivos de seguridad de la red.

TF1-T19 Configuración del SIEM

Configura el SIEM para recibir, procesar y almacenar los eventos y alertas generados por el IDS y otros dispositivos de seguridad de la red. Se definen las políticas de seguridad, se establecen los umbrales de alerta y se personaliza el panel del SIEM para mostrar los datos relevantes para el proyecto.

TF1-T20 Pruebas y Validación del SIEM

Realizar pruebas para validar que el SIEM está funcionando correctamente. Se simulan diferentes tipos de ataques para verificar que el SIEM es capaz de detectar y alertar sobre dichos ataques. Se ajustan las configuraciones y políticas según los resultados obtenidos.

TF1-T3 Configuración S.O en la Raspberry Pi

Configurar el sistema operativo de la Raspberry Pi para soportar el IDS y el software del SIEM. Se instalan las dependencias necesarias y se configura el sistema para optimizar el rendimiento y la seguridad.

TF1-T4 Instalación del software del IDS

Instalación del software del IDS en la Raspberry Pi. Se configura el IDS para que pueda detectar y alertar sobre los comportamientos maliciosos detectados en la red.

TF1-T5 Configuración del IDS

Configurar el IDS para que se adapte a las necesidades específicas del proyecto. Se definen las políticas de detección y se establecen los umbrales de alerta. También se configura el IDS para que envíe las alertas al SIEM para su posterior análisis.

TF1-T6 Pruebas y validación del IDS

Pruebas para verificar que el IDS está funcionando correctamente. Se simulan diferentes tipos de ataques para validar que el IDS es capaz de detectar y alertar sobre dichos ataques. Se ajustan las configuraciones y políticas según los resultados obtenidos.

TF1-T17 Documentar entrega PEC3

Hito 4: Memoria

TF1-T7 Desarrollar Memoria final del TFM

Elaborar el documento final que describe el proceso realizado durante el desarrollo del Trabajo de Fin de Máster. Debe incluir el contexto y justificación del trabajo, los objetivos planteados, la metodología utilizada, resultados obtenidos, conclusiones y recomendaciones. También debe incluir referencias bibliográficas y anexos.

Hito 5: Presentación en video

TF1-T27 Presentación del video

Presentación del video de manera online desde la plataforma de la UOC

TF1-T8 Edición del video

Planificar, grabar y editar un video auto explicativo que demuestre la investigación y la puesta en marcha del proyecto.

Hito 6: Defensa del TFM

TF1-T28 Presentación y defensa del TFM

Presentación y defensa del TFM

1.5. Planificación general

Para la realización de la planificación en cada tarea se ha estimado el esfuerzo en días y asignando un 8% de dedicación (**2 horas trabajadas / día**) Excepto en la tarea "Documentar entrega PEC 3" que se ha considerado mejor una media de 1 hora por día, al ser un hito donde se dedica menos tiempo a documentar y más a parametrizar la solución.

Obteniendo un total de **302 horas de trabajo** con la siguiente relación:

- Hito 1: PEC 1. Plan de trabajo: 36 horas
- Hito 2: PEC 2. Investigación: 94 horas
- Hito 3: PEC 3. Instalación y parametrización de la solución: 90 horas
- Hito 4: PEC 4. Memoria final: 60 horas
- Hito 5: Presentación en vídeo: 14 horas
- Hito 6: Defensa del TFM: 8 horas

Periodo vacacional:

Se ha tenido en cuenta el periodo vacacional de semana santa del 2 de abril al 9 de abril, durante este periodo no se ha planificado ninguna tarea.

Es importante destacar, que en una modalidad de gestión de proyecto ágil la inclusión y estimación de las tareas deberían realizarse al inicio de cada hito, esta planificación es la inicial prevista para el trabajo, pero durante su ejecución ha podido tener variaciones.

Para una mejor comprensión de la planificación en las Ilustraciones 3,4 y 5 se puede apreciar el diagrama de Gantt del TFM.

TÍTULO	HORAS ... (P)	DURACIÓN	FECHA DE INICIO	FECHA DE VENCIMIENTO
Plan de trabajo	-	-	06/03/2023	14/03/2023
General	-	-	06/03/2023	14/03/2023
TF1-T3 Definir contexto y justificación del trabajo	6:00	2 días	06/03/2023 ...	08/03/2023 00:00
TF1-T4 Definir Objetivos	6:00	3 días	06/03/2023 ...	08/03/2023 00:00
TF1-T5 Enfoque y método seguido	6:00	3 días	09/03/2023 ...	11/03/2023 00:00
TF1-T6 Planificiación del Trabajo	10:00	5 días	09/03/2023 ...	13/03/2023 00:00
TF1-T7 Identificar riegos	6:00	3 días	11/03/2023 0...	13/03/2023 00:00
TF1-T8 Planificación costes	2:00	1 días	14/03/2023 0...	14/03/2023 00:00
Agregar tarea				
Investigación	-	-	15/03/2023	11/04/2023
3. Documentación	-	-	15/03/2023	02/04/2023
TF1-T25 Documentar entrega PEC2	38:00	18 días	15/03/2023 0...	02/04/2023 00:00
Agregar tarea				
2. Investigación sistemas SIEM	-	-	24/03/2023	03/04/2023
TF1-T13 Investigar las herramientas SIEM disponibles	6:00	3 días	24/03/2023 ...	26/03/2023 00:00
TF1-T14 Como integrar el IDS con SIEM	8:00	3 días	25/03/2023 ...	28/03/2023 00:00
TF1-T15 Técnicas de análisis de logs o dashboards	6:00	3 días	29/03/2023 ...	31/03/2023 00:00
TF1-T16 Técnicas de detección comportamientos malintencionados	6:00	4 días	31/03/2023 0...	03/04/2023 00:00
Agregar tarea				
1. Investigación IDS	-	-	15/03/2023	24/03/2023
TF1-T9 Investigación Tipos de IDS	8:00	3 días	15/03/2023 0...	18/03/2023 00:00
TF1-T10 Herramientas captura de tráfico	6:00	3 días	16/03/2023	18/03/2023 00:0
TF1-T11 Investigar implementar IDS en Raspberry Pi	8:00	3 días	19/03/2023 0...	22/03/2023 00:00
TF1-T12 Investigar técnicas de análisis de tráfico	8:00	3 días	21/03/2023 0...	24/03/2023 00:00

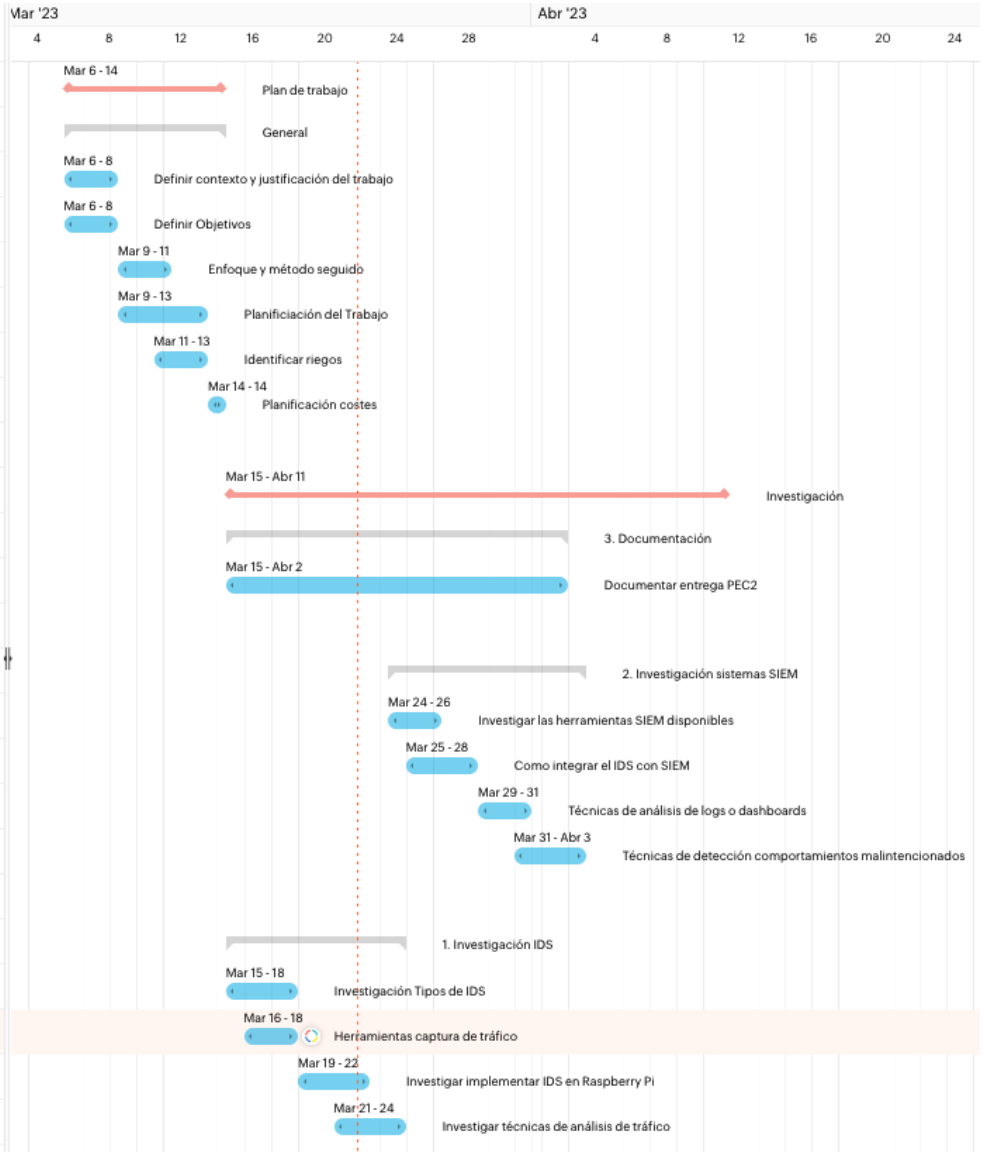


Ilustración 3 Diagrama Gantt 1 (Fuente: Elaboración propia)

TÍTULO	HORAS ... (P)	DURACIÓN	FECHA DE INICIO	FECHA DE VENCIMIENTO
Parametrización	-	-	12/04/2023	09/05/2023
Documentación	-	-	12/04/2023	09/05/2023
TF1-T24 Documentar entrega PEC3	28:00	27 días	12/04/2023 0...	09/05/2023 00:00
Agregar tarea				
Parametrización del SIEM	-	-	23/04/2023	02/05/2023
TF1-T21 Instalación Software SIEM	6:00	3 días	23/04/2023 ...	25/04/2023 00:00
TF1-T22 Configuración del SIEM	14:00	6 días	25/04/2023 ...	01/05/2023 00:00
TF1-T23 Pruebas y Validación del SIEM	10:00	4 días	28/04/2023 ...	02/05/2023 00:00
Agregar tarea				
Parametrización IDS en Raspberry-pi	-	-	12/04/2023	25/04/2023
TF1-T17 Configuración S.O en la Raspberry Pi	4:00	1 días	12/04/2023 0...	13/04/2023 00:00
TF1-T18 Instalación del software del IDS	4:00	1 días	15/04/2023 0...	16/04/2023 00:00
TF1-T19 Configuración del IDS	14:00	6 días	18/04/2023 0...	24/04/2023 00:00
TF1-T20 Pruebas y validación del IDS	10:00	4 días	21/04/2023 0...	25/04/2023 00:00
Agregar tarea				

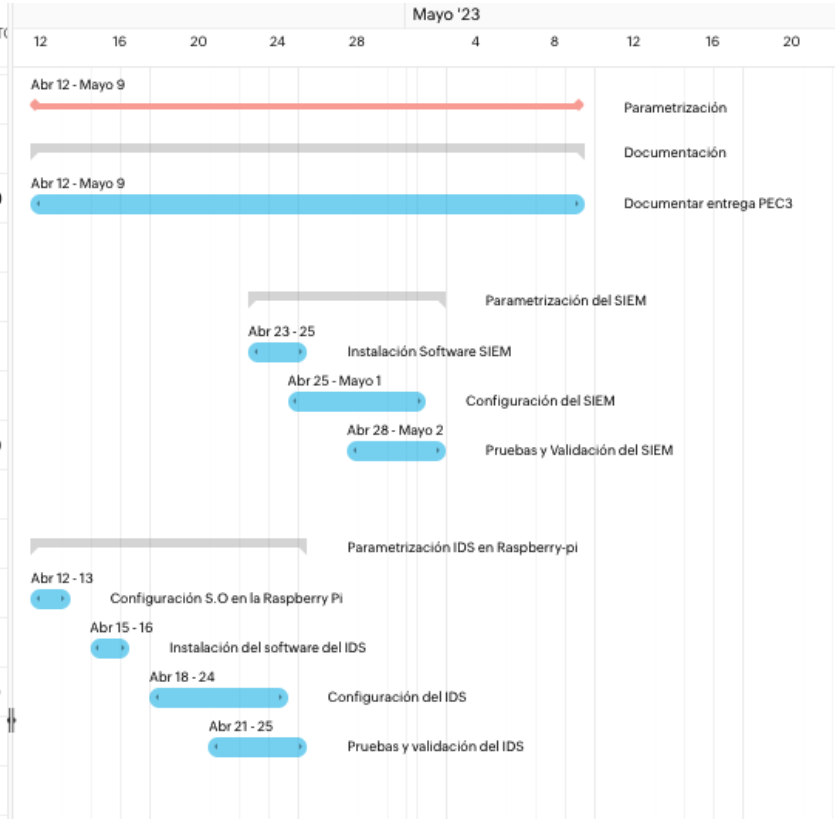


Ilustración 4 Diagrama Gantt 2 (Fuente: Elaboración propia)

Ilustración 5 Diagrama Gantt 3 (Fuente: Elaboración propia)

1.6. Revisión del estado del arte.

En la actualidad, la seguridad en las redes domésticas es un tema de gran importancia debido al aumento en el número de dispositivos conectados a internet en los hogares, lo que aumenta el riesgo de ataques informáticos y violaciones de nuestra privacidad.

Para fortalecer la seguridad en una red se pueden utilizar soluciones IDS que existen de dos tipos, los IDS basados en Host y los IDS basados en la red. Los primeros monitorizan el tráfico en una máquina individual solamente mientras que los de host monitorizan el tráfico en la red, es en este segundo tipo donde se enfocará el TFM especialmente en una red doméstica.

Existen varios proyectos IDS open source como:

- Bro/Zeek: open source y utilizado por varias organizaciones, capaz de analizar y generar alertas del en base al tráfico de la red.
- Snort: uno de los pioneros y más famosos desde el 1998 creado por Martin Roesch, donde por medio de unas reglas se definen eventos y alertas.
- **Suricata**: un Sistema de Detección de Intrusos de código abierto que permite detectar actividades sospechosas en la red, como intentos de acceso no autorizado o malware, y alertar al usuario para que pueda tomar medidas de seguridad.

Ilustración 6 Suricata Logo (Fuente: web suricata)

Para facilitar la gestión de eventos de seguridad producidos en parte por un IDS, es necesario utilizar un SIEM. Existen grandes soluciones tanto comerciales y de código abierto.

Como comerciales se pueden encontrar IBM QRadar, Splunk Enterprise Security y LogRhuthm. Y open source la más utilizada y famosa es el stack de ELK (Elasticsearch, logstash, Kibana), siendo esta la elegida para recolectar, analizar y alertar en tiempo real las alertas del IDS que analizaremos en el TFM.

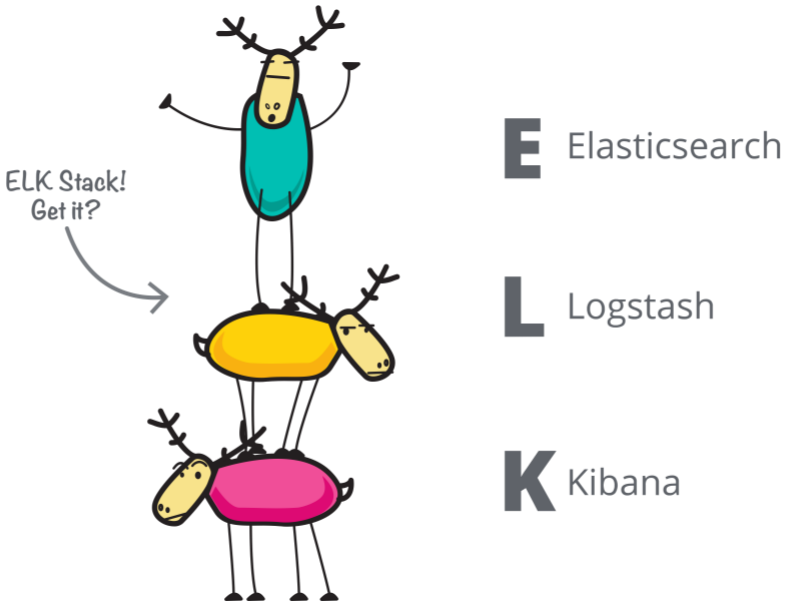


Ilustración 7 ELK Stack (Fuente: website elastic)

Para la implementación de Suricata y ELK en una red doméstica, la opción más sencilla es utilizar una **Raspberry Pi**, un miniordenador de bajo coste y bajo consumo de energía. Que puede ser configurado como IDS y, gracias al bajo consumo puede funcionar sin generar demasiados costes en electricidad.

Además de fortalecer la seguridad en la red doméstica, la implementación de Suricata y ELK en una Raspberry Pi puede brindar beneficios en el uso diario de la red. Por ejemplo.

- Por medio de las visualizaciones de datos en tiempo real con Kibana se puede permitir a los usuarios ver y controlar el tráfico de sus dispositivos, identificar el consumo de ancho de banda y/o la latencia, además de poder realizar ajustes para optimizar la calidad de la conexión.
- También ayudar a los usuarios a identificar dispositivos maliciosos o aplicaciones que generan problemas en la red.

En resumen, la integración de Suricata y ELK en una Raspberry Pi podrá ser una solución efectiva y de bajo coste para fortalecer la seguridad en una red doméstica. En este TFM se verá la viabilidad de su instalación y los beneficios de su implementación en una red doméstica.

1.7 Listado de posibles riesgos y soluciones que pueden afectar al desarrollo y cumplimiento de los objetivos del TFM propuesto.

A continuación, se detallan la lista de riesgos y soluciones que pueden afectar al desarrollo y cumplimiento del TFM

Falta de experiencia en la implementación de sistemas de detección de intrusos.

Existe el riesgo de no tener la suficiente experiencia en la implementación de sistemas de detección de intrusos que podría llevar a una mala configuración o efectividad en la detección de intrusos.

Solución: Durante la fase de investigación y parametrización, se deberá realizar un buen análisis de las soluciones IDS y SIEM para adquirir los conocimientos necesarios.

Falta de compatibilidad entre componentes

Si no se realiza una selección y prueba de componentes como la Raspberry Pi y el software, podría darse el caso de incompatibilidad entre algunos de ellos, lo que podría retrasaría la implementación del sistema

Solución: Durante la fase de investigación se deberá asegurar de la selección de herramientas y software sea compatible son la solución propuesta.

Proyecto no escalable:

Se podría dar el caso que el proyecto no se adaptara a las necesidades de la red actual, que no pudiera hacer frente a la demanda de tráfico y no fuera escalable.

Solución: En este aspecto durante la fase de investigación se deberá analizar las dimensiones de red que soporta la solución para prever que pueda ser escalable.

Fallos en el hardware.
Podría darse problemas en el hardware como en la raspberry PI, que pudiera presentar fallos lo que podría llevar a la perdida de información o la interrupción del trabajo.

Solución: En este aspecto se guardará un presupuesto más para desastres que pueda hacer frente a estos imprevistos.

Falta de tiempo

Se podría dar el caso donde la planificación y la realidad no vayan a la par, y falte tiempo para redactar la memoria.

Solución: Durante el desarrollo de todo el TFM se irá documentando todo el proceso con entregas parciales.

1.7. Listado de los costes asociados al material.

Para la realización del trabajo se ha realizado una estimación inicial del material asociado al trabajo para validar su viabilidad y el objetivo inicial del trabajo.

Los costes asociados se pueden ver en la Tabla 1 Costes asociados.

Tabla 1 Costes asociados (Elaboración propia)

Tipo	Tipo	Coste
Router/Modem (Livebox+)	Hardware	0 € (Reutilizado compañía internet)
Raspberry Pi	Hardware	62,90 € (Reutilizado)
Switch	Hardware	30 €
Router (Access point TP-Link N750)	Hardware	0 € (Reutilizado)
Suricata (IDS)	Software	0 €
Elastic Search /Logstash / Kibana (SIEM)	Software	0 €
		Total: 122,90 €

Se ha obviado el coste de internet y el uso de un ordenador personal, por entenderse que en la unidad doméstica se da por hecho que existe una contratación de internet y un ordenador como mínimo.

Es importante destacar, que en los últimos años a raíz de la pandemia del coronavirus y el aumento de los costes de los chips. La raspberry PI está siendo un producto que ha encarecido su coste, pero aún sigue siendo una solución mucho más rentable que otras soluciones, pero sería interesante buscar alternativas.

1.8. Implicaciones ético-legales del producto.

El presente TFM tiene una gran importancia en la dimensión de la sostenibilidad, especialmente en el **ODS 8 crecimiento económico**.

El trabajo está enfocado para el uso doméstico, utilizando soluciones de bajo coste como la Raspberry Pi y herramientas open source para soluciones IDS y SIEM. Esto permite reducir los costes económicos y ambientales asociados a la adquisición de soluciones comerciales de alta gama, lo que contribuye a la reducción del impacto ambiental y a la mejora de recursos. Aportando un crecimiento a la economía global por ser un producto de bajo coste disponible para la mayoría de la población.

También, la detección de amenazas a través de soluciones IDS es fundamental para garantizar la seguridad de la información, reduciendo riesgos de ataques informáticos y protegiendo la privacidad y confidencialidad de los datos. Para implementar este tipo de soluciones suele ser costoso por su complejidad de la implementación y costes asociados al hardware. Siendo poco accesible para el resto de la población, con ayuda de este trabajo se consigue reducir los costes y simplificar su implementación para que todo el mundo pueda utilizarlo en casa.
Muy alineado con la dimensión III en cuestión de derechos humanos, reduciendo así las desigualdades entre los seres humanos **(ODS 10 Reducir las desigualdades)**

En resumen, el TFM contribuye a la sostenibilidad utilizando herramientas y tecnologías de bajo coste y de código abierto reduciendo el impacto ambiental y protegiendo la privacidad y confidencialidad de la información para todos los humanos.

2. Investigación

Se procede a realizar una investigación de los diferentes sistemas IDS/SIEM, sus tipologías y como pueden interaccionar para el reporte de alertas.

2.1 Investigación Tipos de IDS

Un IDS es como un sistema de seguridad de videovigilancia en una casa. Del mismo modo que este sistema, el IDS monitorea continuamente la red o un sistema operativo en busca de actividad sospechosa o intruso. En el caso que se detecte una actividad que pueda ser amenaza, el IDS genera una alerta para notificar al dueño de seguridad. Siendo muy parecido cuando una alarma de seguridad en casa puede sonar para alertar al propietario de un posible intruso.

Siguiendo el símil de la vigilancia en casa, el sistema de vigilancia puede tener diferentes sensores y cámaras para monitorear varias habitaciones o áreas, el IDS también tiene diferentes tipos de sensores o tecnologías que monitorizan diferentes tipos de tráfico o actividad.

Un **IDS es como tener un sistema de seguridad en la red para detectar y notificar amenazas** de la misma manera que haría un sistema de seguridad en el hogar.

En el mercado hay tres tipos de IDS:

1. Basado en la **red** (NIDS): Este tipo monitorea el tráfico de red para buscar actividad maliciosa. Este tipo de IDS puede ser implementado en un equipo separado dedicado exclusivamente a la detección de amenazas como es el caso del proyecto actual.

2. Basado en **host** (HIDS): Este tipo puede ser más sencillo porque únicamente monitoriza la actividad en un sistema operativo, para la detección de amenazas dentro del host.

3. Modelo híbrido (**HIDS/NIDS**): Es una fusión de características de ambos sistemas que proporciona una mayor cobertura de seguridad. Los IDS híbridos pueden monitorear el tráfico de red y la actividad del host para detectar amenazas en ambos entornos.

Cada uno tiene sus ventajas y desventajas, su elección depende de los objetivos de seguridad, en nuestro caso al querer implementar una solución sencilla para aplicar en toda la red doméstica, se utiliza del tipo NIDS, como veremos más adelante.

Los NIDS además de poder detectar y monitorizar la actividad en busca de intrusiones o actividad maliciosa, también puede tomar la posición de responder a ellas. Esta última es capaz de tomar medidas al momento para mitigar o detener un ataque en tiempo real. Podría bloquear el tráfico malicioso, o no dejar conectar con un host malicioso o enviar alertas para que se tomen medidas. Este último tipo se conocen como IPS un tipo de NIDS reactivo que además de detectar y registrar, también puede tomar medidas preventivas automáticamente sin necesidad de interactuar con él.

Algunas acciones típicas que puede realizar en base a reglas y políticas de seguridad podrían:

- Bloquear tráfico
- Filtrar paquetes de red específicos
- Bloquear IP

Uno de los puntos negativos de los IPS es que, si no están bien configurados, podrían generar falsos positivos, bloqueando tráfico que es correcto. Por lo que su uso y configuración hay que realizarlo cuidadosamente.

2.2 Topología de la red

Como hemos visto es posible diseñar una solución IDS de forma pasiva o reactiva (también llamado IPS) para poder asegurar la red y proteger los dispositivos o datos personales de los usuarios.

IDS Pasivo

En la ilustración 8 se muestra como quedaría un IDS pasivo con el material disponible.

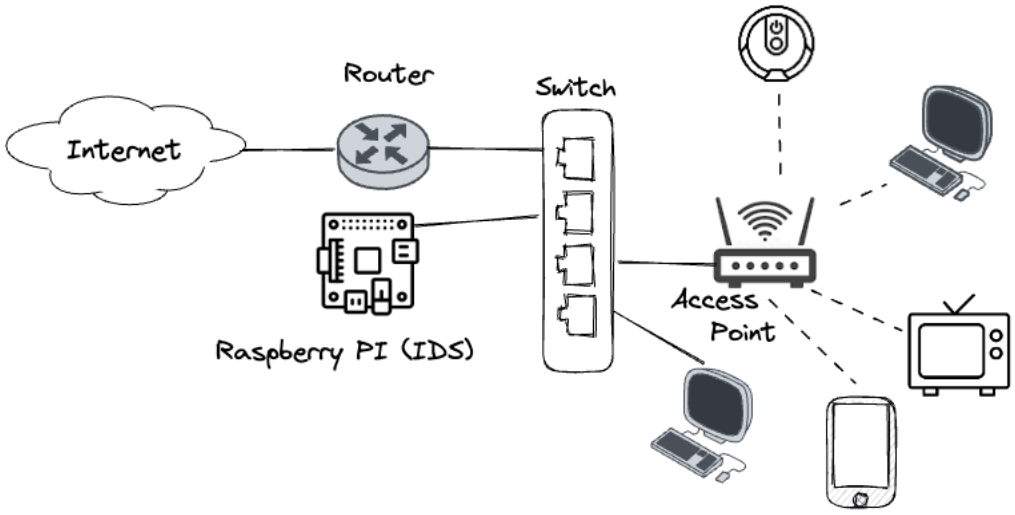


Ilustración 8 Topología IDS Pasivo (Elaboración propia)

La Raspberry al solo tener un solo puerto de red y poder conectarla al switch es necesario utilizar la técnica de "**port mirroring**". Esta técnica consiste en configurar el switch para que copie todos los paquetes que le llegan a un determinado puerto (donde se conecta el router) y los envíe también al puerto donde está conectada la Raspberry Pi.

De esta manera la Raspberry Pi puede leer todos los paquetes que pasan por la red tanto de entrada como salida sin tener que estar en línea con el resto de los dispositivos.

Esta topología como hemos visto tiene una limitación, el IDS al pasar a un segundo plano y solo estar de forma pasiva en la lectura de datos no podrá utilizarse como IPS y

pueda reaccionar automáticamente ante intrusiones en los dispositivos, solamente podrá registrar las intrusiones.

Las ventajas principales son:

- El rendimiento de la red no se ve afectado
- No se necesita un hardware con mucho procesamiento, como la Raspberry PI

Las desventajas que se encuentran:

- No se puede reaccionar en el tráfico malicioso, porque no se puede utilizar como IPS.
- No es posible restringir el acceso en los dispositivos.
- Requiere más dispositivo en la solución, con lo que complica su implementación.

IPS - IDS Reactivo

En la ilustración 9 la Raspberry Pi se conecta directamente al router actuando como dispositivo IDS y Access Point a la vez, haciendo de **gateway**. Al canalizar y monitorizar toda entrada y salida de la red a través de la Raspberry PI se podría configurar de maneras como un IPS.

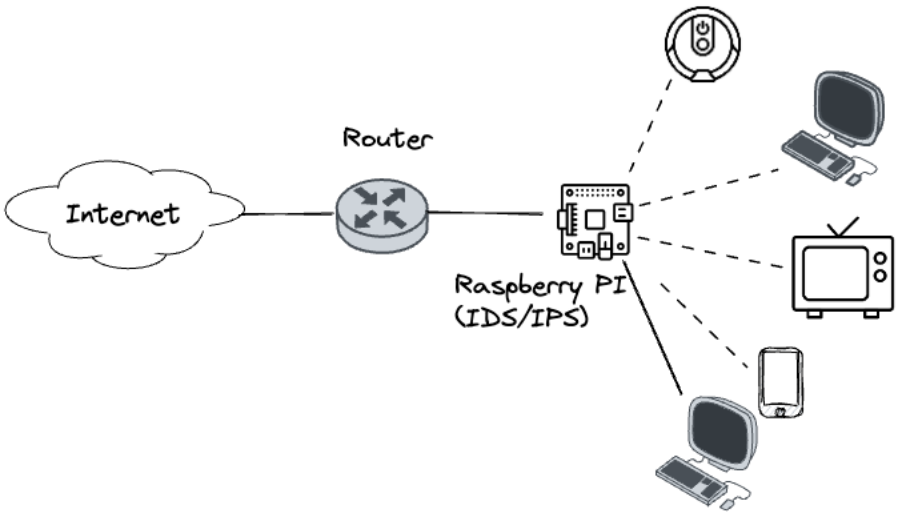


Ilustración 9 IDS Reactivo (Elaboración propia)

Es importante tener en cuenta que una Raspberry Pi, no está diseñada para manejar grandes cantidades de tráfico de red. es muy probable que se convierta en un cuello de botella, por lo que puede ralentizar la velocidad de internet.

Una opción a esta topología sería utilizar un dispositivo con mayor capacidad que la raspberry como un servidor dedicado o un equipo de red especializado.

Las ventajas principales son las siguientes:

- Es posible utilizar el IDS como IPS también y poder actuar antes amenazas
- Es posible configurar el IDS como cortafuegos también
- Simplifica la solución con menos dispositivos, además la raspberry pi quedaría como punto de acceso también.

Tiene algunas desventajas:

- Al depender de la raspberry Pi, es probable que se creen cuellos de botella que reduzcan el rendimiento de la red.
- Al añadir más funciones también reducirá el rendimiento de la propia raspberry pi. No está bien dimensionado para las funciones que debe realizar.

IPS - IDS Reactivo 2

Una solución alternativa para poder quitarle carga a la raspberry PI, sería añadir un Access point para el tráfico de la LAN. Pero aun así todo el cálculo y procesamiento deberá realizarlo la PI, por lo que llevaría a otro cuello de botella. La ilustración 10 muestra esta variación.

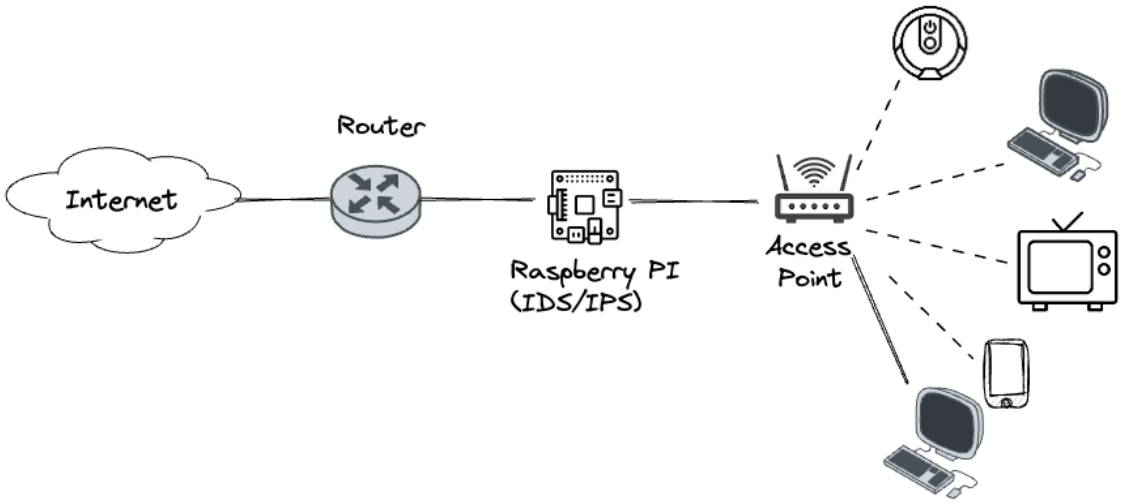


Ilustración 10 IDS Reactivo 2 (Elaboración propia)

Por el posible cuello de botella y falta de rendimiento que se podría dar con las dos últimas tipologías, la opción seleccionada para implementar la solución IDS en una red doméstica ha estado la opción **IDS pasivo.**

2.3 Herramientas captura de tráfico

Como se ha visto la función principal de los IDS es la captura y lectura del tráfico para que pueda ser analizado. Para ello es necesario el uso de ciertos programas que son esenciales para dos puntos importantes: la detección y la monitorización de la seguridad de la red.

Existen muchas herramientas de captura de tráfico, cada una de ellas con sus pros y contras. Las que se han analizado son de código abierto y de bajo coste. Además, ligeras en cuanto a consumo de recursos, para que pueda ser procesada por una raspberry PI y con interfaz sencilla de usar y configurar.

Las dos herramientas investigadas han estado Snort y Suricata.

Snort

Es uno de los IDS open source más famosos y con mucho tiempo en el sector (1988) y con una buena comunidad por detrás.

Tiene un motor de detección de firmas para identificar patrones de tráfico malicioso en la red.

Analiza los protocolos TCP, UDP, ICMP, DNS, HTTP, FTP entre otros, además tiene soporte para redes IPv6
Es una solución multihilo desde su versión 3.0, lo que puede aprovechar las arquitecturas de los sistemas actuales multinúcleo (utilizando todos los núcleos de la CPU) siendo una opción interesante para mostrar amenazas en tiempo real.

Snort se basa principalmente en escuchar el tráfico y comparándolo con unas reglas que tiene en la base de datos y en base a estas reglas puede identificar intrusiones o anomalías.

Otro de los beneficios más importantes de Snort es la comunidad que hay detrás de ella, en snort se pueden crear y compartir reglas por medio de repositorios públicos que todos pueden utilizar.

La interfaz de snort es la línea de comandos

Suricata

Es otro IDS open source con interfaz por la línea de comandos que utiliza la inspección profunda de paquetes (DPI) para analizar el tráfico y ver amenazas. Donde puede realizar análisis en tiempo real y con un catálogo de reglas ya predefinidos. Fue definida como mejora para dispositivos ligeros y una mayor optimización en comparación con Snort.

Suricata utiliza un motor de reglas para definir las políticas de detección de amenazas. Y los mismos usuarios pueden crear reglas propias o importar reglas de reglas de los repositorios de Snort, lo que lo hace muy interesante.

Además, es multi-hilo por lo que es posible optimizarlo con procesadores multicore, permitiendo un mejor rendimiento en red.

Puede integrarse con otras herramientas como SIEM, para mejorar la detección y respuesta.

Analiza varios protocolos como: TCP, UDP, ICMP, DNS, HTTP, FTP entre otros y tiene soporte para IPv6 de manera que es capaz de procesar tráfico de las redes que lo utilizan.

Una de las características interesantes de suricata frente a snort en una red domestica como es el caso del proyecto, es que suricata permite detectar escaneo de puertos mediante el análisis del tráfico TCP y UDP, mientras que Snort se enfoca en la identificación de patrones de tráfico en paquetes de red específicos.

Por ser una solución Open Source, que permite multi-hilo, y permite aprovechar las reglas de snort y crear de nuevas, siendo concebido para integrarse con soluciones con menos recursos como la raspberry Pi, **Suricata es el IDS que se elige para el proyecto.**

2.4 **Investigar implementar Suricata IDS en Raspberry Pi**

Como la elección del IDS ha estado Suricata lo primero que se va a investigar, son las características de suricata y Raspberry PI para ver si es viable y como podría implementarse el IDS en la red doméstica.

Para poder valorar las capacidades técnicas del IDS es necesario conocer las especificaciones técnicas de la raspberry PI y ver si está bien dimensionada la topología anterior.

Características de la raspberry PI utilizada en el proyecto (Modelo 3 B)

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI

Como se puede ver las especificaciones de la raspberry son muy modestas, solamente por el tipo de red y la capacidad de memoria RAM **sería insuficiente para montar suricata en un ámbito profesional**, pero al ser un proyecto en una red doméstica donde el tráfico de la red es mucho más limitado y su uso es menor, se intentará demostrar que puede actuar como IDS pasivo sin problemas.

Para conocer mejor el IDS veamos cómo funciona realmente y como se utiliza suricata como IDS.

En siguiente diagrama de procesos se puede ver las etapas de un IDS y los componentes involucrados en cada etapa de la detección de amenazas y en cuales suricata está involucrado.

Ilustración 11 Etapas de un IDS (Elaboración propia)

La captura de paquetes de red se realiza utilizando la tarjeta de red que hay en la raspberry Pi. Hay que recordar que todo el tráfico es enviado a través del switch gracias a la opción "**port mirroring**" que se ha comentado anteriormente.

Suricata tiene un motor de análisis que extrae los metadatos de los paquetes de red, inspecciona el contenido y compara los paquetes capturados con el conjunto de reglas de detección. En el caso que detecte una coincidencia, el motor de alertas de Suricata genera una alerta, y en el caso que se necesite visualizar en un entorno SIEM que veremos en el siguiente apartado, la alerta se guardará en una base de datos como (Elasticsearch) y visualizará las alertas por medio de una herramienta de visualización (Kibana).

2.5 **Funcionamiento de las reglas**

Una de la parte más importante de entender y conocer el funcionamiento de Suricata, es conocer bien las reglas con las que analiza el tráfico y genera alertas Suricata en un caso real.

Digamos que un atacante intenta iniciar sesión por SSH en cualquier dispositivo de nuestro hogar utilizando diferentes combinaciones de usuario o contraseñas, comúnmente conocido como ataque de fuerza bruta.

Para detectar este tipo de ataque, Suricata utiliza una regla que se divide en tres dimensiones:

- La acción: Es lo que debe acceder si se cumple la condición. En el ejemplo sería generar la alerta y registrarla en la base de datos.
- La cabecera: Define el protocolo, las direcciones IP, puertos de la regla
- Opciones o condiciones: Define las especificaciones de la regla, podría entenderse como las condiciones que se deben de dar.

La regla del caso de ejemplo tendría el siguiente formato:

```
alert tcp any any -> $ HOME_NET 22 (msg:"SSH ataque de fuerza bruta
detectado"; flow:to_server,established; content:"authentication
failed"; nocase; threshold:type threshold,track by_src,seconds 60,
count 5; classtype:attempted-admin; sid:1000001; rev:1;)
```

En el ejemplo, lo rojo es la acción, en verde la cabecera y en azul las opciones o condiciones.

Acciones (Actions)

Suricata tiene las siguientes acciones posibles:

- alert - genera una alerta
- pass - detiene la inspección adicional del paquete
- drop - descarta el paquete y genera una alerta
- reject - envía un error RST/ICMP unreachable al remitente del paquete que coincide.

- rejectsrc - lo mismo que solo reject
- rejectdst - envía un paquete de error RST/ICMP al receptor del paquete coincidente.
- rejectboth - envía paquetes de error RST/ICMP a ambos lados de la conversación.

Cabeceras (Headers)

Dentro de las cabeceras se pueden diferenciar el protocolo, el origen, la dirección y el destino

```
alert tcp any any -> $ HOME_NET 22 (msg:"SSH ataque de fuerza bruta
detectado"; flow:to_server,established; content:"authentication
failed"; nocase; threshold:type threshold,track by_src,seconds 60,
count 5; classtype:attempted-admin; sid:1000001; rev:1;)
```

Los protocolos son la primera parte de la cabecera:

```
alert tcp any any -> $ HOME_NET 22 (msg:"SSH ataque de fuerza bruta
detectado"; flow:to_server,established; content:"authentication
failed"; nocase; threshold:type threshold,track by_src,seconds 60,
count 5; classtype:attempted-admin; sid:1000001; rev:1;)
```

Los básicos son:

- tcp (para tráfico tcp)
- udp
- icmp
- ip (ip stands for ‘all’ or ‘any’)

Luego a nivel de aplicación están disponibles los siguientes:

http, ftp, tls, smb, dns, dcerpc, ssh, smtp, imap, modbus, dnp3, enip, nfs, ikev2, krb5, ntp, dhcp, rfb, rdp, snmp, tftp, sip, http2.

Origen y Destino

El origen y destino se compone de la dirección más el puerto. El primer grupo pertenece al origen donde proviene el tráfico y el segundo el destino.

```
alert tcp any any -> $HOME_NET 22 (msg:"SSH ataque de fuerza bruta
detectado"; flow:to_server,established; content:"authentication
failed"; nocase; threshold:type threshold,track by_src,seconds 60,
count 5; classtype:attempted-admin; sid:1000001; rev:1;)
```

any puede ser cualquier dirección o cualquier puerto. Para más detalle revisar la documentación de suricata

Opciones

Las opciones de la regla o condiciones consiste en una serie de parámetros separados por punto y coma **;** y englobados por paréntesis **()**. Algunas son comunes para todo tipo de protocolo y otras son específicas para un protocolo determinado.

Todas ellas se pueden encontrar en la documentación de Suricata

2.6 Técnicas de análisis de tráfico

Para poder implementar un buen IDS es necesario conocer bien las técnicas de análisis de tráfico porque permiten identificar patrones y comportamientos sospechosos, ayudando a detectar mejor las intrusiones o ataques.

Dentro de las técnicas de análisis de tráfico para examinar y detectar amenazas o anomalías, podemos diferenciar de diferentes tipos en base a las reglas de suricata.

Para entender mejor las posibles técnicas que podremos utilizar en la implementación a continuación se define un ejemplo para cada una de ellas:

Análisis de patrones de tráfico: Ej: Buscar patrones en los paquetes HTTP que contengan una palabra en especial, por ejemplo "malware"

```
alert http any any -> any any (msg:"Detected malware traffic";
content:"malware"; sid:10001; rev:1;)
```

Análisis de firmas: Detectar consultas DNS que contengan el dominio malwaredomain.com.

```
alert dns any any -> any any (msg:"DNS Query for known malicious
domain"; dns_query; content:"|09|malwaredomain|03|com"; nocase;)
```

Análisis de comportamientos: Tráfico anómalo como el ejemplo anterior que intenta en un periodo corto de tiempo varios intentos anomalos

```
alert tcp any any -> $HOME_NET 22 (msg:"SSH ataque de fuerza bruta
detectado"; flow:to_server,established; content:"authentication
failed"; nocase; threshold:type threshold,track by_src,seconds 60,
count 5; classtype:attempted-admin; sid:1000001; rev:1;)
```

Análisis de protocolo: Buscar Inyección SQL en el protocolo HTTP, que se utiliza en aplicaciones web.

```
alert http any any -> any any (msg:"Potential SQL Injection Attack";
flow:to_server,established; content:"\' or \'1\'=\'1"; nocase;
http_uri; classtype:web-application-attack; sid:1000001; rev:1;)
```

Análisis de anomalías: Por ejemplo detectar conexiones salientes a puertos no estándar, podría indicar alguna presencia de malware o actividad sospechosa.

```
alert tcp any any -> any !80:!443 (msg:"Possible malware activity -
Non-standard outgoing port"; sid:100001; rev:1;)
```

2.4 Investigar las herramientas SIEM disponibles

Hasta el momento la implementación de un IDS como Suricata es un paso importante para la seguridad de la red. Sin embargo, para la gestión completa es necesario contar un sistema de gestión de eventos e información de Seguridad, llamado SIEM. En este apartado se va a repasar para qué es, que tipos hay y cuál ha sido la elección para integrar en la topología definida al inicio.

Un Sistema SIEM (Security Information and Event Managment) es una solución de gestión de eventos e información de seguridad. Sus características principales es detectar, responder y neutralizar las amenazas.

SIEM nace de dos soluciones (SEM y SIM):

La tecnología de **SIEM** surge de la unión de **SEM** (gestión de eventos de seguridad) y **SIM** (gestión de información de seguridad). SEM **recopila** y **analiza** los datos en tiempo real, lo que permite una mayor visibilidad y la detección de patrones anormales en los sistemas de seguridad. Mientras que SIM **almacena** los datos a largo plazo para **generar informes** automatizados de seguridad.

Al combinarlas, el SIEM permite detectar amenazas tanto internas como externas en tiempo real, lo que permite una respuesta rápida a los ataques más difíciles de detectar. A diferencia de otras tecnologías de seguridad, el SIEM ofrece una capacidad de detección de amenazas más amplia y proporciona informes para mejorar la capacidad de investigación y cumplir con los objetivos de seguridad.

En el caso de este proyecto Suricata actuará como detector de eventos de seguridad y lo en enviará al SIEM a través de Syslog. Y la Raspberry Pi actuará como un dispositivo de recolección de datos, ejecutando Suricata y enviando los datos de seguridad al SIEM.

Esta integración permitirá al SIEM detectar y responder de manera más rápida a las amenazas de la red doméstica, ya que Suricata puede detectar patrones de tráfico maliciosos y ejecutar reglas personalizadas para la detección de amenazas propias de una red doméstica.

Uno de los beneficios principales al implementar SIEM:

1. Facilidad en detención de amenazas: Al tener los datos más fáciles de interpretar se agiliza su detección.
2. Reduce el tiempo de respuesta: Al agilizar su detección, también se reduce el tiempo de respuesta.
3. Visibilidad en tiempo real: Al leer los datos en tiempo real ayuda a tener una visibilidad también en tiempo real de la seguridad de la red.

2.5 SIEM con ELK Stack

De los SIEM analizados en esta fase de investigación se ha decidido utilizar la solución ELK. Es una solución que utiliza varias herramientas que permite monitorizar y analizar logs de múltiples fuentes y servidores.

Su elección ha sido principalmente por ser un proyecto open source con una gran comunidad por detrás, permite leer los logs producidos por Suricata además de cualquier otro log del sistema, siendo su interfaz bastante amigable y fácil de utilizar.

ELK son la suma de tres soluciones **Elastic Search, Log stash y Kibana.**

La compañía Elastic también ofrece dos soluciones para enviar datos a la suite ELK **Beats** o **Elastic Agent** que se verá más adelante y se decidirá cuál es la más conveniente para el proyecto

Logstash: La herramienta de procesamiento de datos para la ingestión, transformación y envío de logs y otros datos.

Elasticsearch: Es el motor de búsqueda y análisis de datos distribuido que permite el almacenamiento y búsqueda de datos en tiempo real.

Kibana: La herramienta de visualización de datos que permite la exploración, análisis y presentación de datos almacenados en Elasticsearch.

Beats: Son unos agentes ligeros que se utilizan para enviar datos a Logstash o Elasticsearch, permitiendo la recopilación y envío de datos en tiempo real.

Elastic Agent: es un solo agente que unifica logs, métricas, datos de seguridad y prevención de agujeros de seguridad. Puede desplegarse de dos formas “Managed by Fleet” o “Standalone mode”

Logstash

Se utiliza para recopilar, transformar y enrutar datos en tiempo real. Su función principal es la de recibir, transformar y enviar logs y otro tipo de datos a diferentes destinos. Las características principales son:

- **Flexibilidad**: es altamente personalizable y se puede utilizar para procesar todo tipo de datos, desde logs de servidores web hasta datos de sensores IoT.
- **Varias fuentes de información**: puede procesar datos de diversas fuentes, como logs de servidores, bases de datos, APIs, mensajes de sistemas de mensajería, entre otros.
- **Transformación de datos**: tiene una amplia gama de filtros integrados para transformar datos, como filtros de grok para analizar logs, filtros de geoip para agregar información de ubicación geográfica a los datos, entre otros.
- **Enrutamiento de datos**: Puede enrutar datos a diferentes destinos, como Elasticsearch, sistemas de almacenamiento en la nube o servicios de mensajería.
- **Escalabilidad**: procesa grandes cantidades de datos y es altamente escalable para manejar grandes flujos de datos en tiempo real.

Tiene un sistema de plugins que facilita la integración con diferentes fuentes de datos y destinos, además que puede trabajar en una arquitectura distribuida, aumentando la capacidad de proceso y poder escalar horizontalmente.

Elasticsearch

Es el motor de búsqueda y análisis de datos de forma distribuida, escalable y permite almacenar, buscar y analizar grandes cantidades de datos en tiempo real. Con su alta disponibilidad y su escalabilidad permite manejar grandes volúmenes de datos.

Entre las características principales de Elasticsearch se encuentran:

- **Búsqueda y análisis en tiempo real de datos**: Permite realizar búsquedas en tiempo real y analizar los datos de forma dinámica para detectar patrones y tendencias.
- **Escalabilidad**: altamente escalable y puede utilizarse en clústeres de varios nodos para manejar grandes volúmenes de datos.
- **API RESTful**: Se integra con otras aplicaciones a través de una API RESTful.
- **Motor de búsqueda y análisis**: proporciona herramientas de análisis de datos para detectar patrones y tendencias en los datos.
- **Potentes capacidades de consulta**: permite realizar consultas complejas y avanzadas sobre los datos almacenados, lo que permite obtener información valiosa para la toma de decisiones.
- **Facilidad de uso**: fácil de instalar, configurar y utilizar, lo que lo hace ideal para una amplia gama de aplicaciones.

Se puede decir que es una herramienta poderosa para el análisis y búsqueda de grandes volúmenes de datos en tiempo real, con una amplia gama de funcionalidades y una fácil integración con otras aplicaciones a través de su API RESTful.

Kibana

es la solución de visualización de datos que permite la creación de dashboards y gráficos interactivos para analizar y visualizar grandes volúmenes de datos en tiempo real.

Entre sus principales características se incluyen:

- **Creación de visualizaciones personalizadas**: ofrece una gran cantidad de tipos de gráficos y herramientas para personalizar la visualización de los datos, como gráficos de barras, líneas, mapas y tablas.
- **Filtros y búsquedas avanzadas**: permite aplicar filtros y búsquedas avanzadas para explorar de forma eficiente y efectiva.
- **Dashboards interactivos**: permite la creación de dashboards interactivos con múltiples paneles que se actualizan en tiempo real
- **Integración con otros sistemas**: se integra fácilmente con Elasticsearch, Logstash, Beats y otras herramientas de análisis de datos.
- **Acceso y seguridad**: puede controlar el acceso y las acciones que se pueden realizar sobre los datos.

Es la herramienta esencial para visualizar y analizar grandes volúmenes de datos en tiempo real, lo que permite tomar decisiones rápidas y efectivas. A modo de ejemplo en la ilustración 12 es posible ver un dashboard en acción con suricata.

Ilustración 12 Kibana - Suricata - Dashboard (Elaboración propia)

Beats:

Son uno agentes ligeros que se encargan de recolectar datos de diferentes fuentes y enviarlos a Elasticsearch o Logstash para su almacenamiento, procesamiento y visualización en Kibana. Dependiendo si se quiere transformar el dato o no se enviará elasticsearch o a logstash.

Están diseñados para ser fáciles de instalar y configurar, tienen bajo impacto en el rendimiento del sistema y están optimizados para trabajar con Elasticsearch y se integran perfectamente con el resto de la pila ELK.

Existen diferentes tipos de beats, cada uno diseñado para recolectar datos de diferentes fuentes:

- **Filebeat**: recolecta datos de archivos de registro, como logs de sistemas, aplicaciones y servicios. Este será el tipo de beat que se utilizará para recolectar los datos de Suricata.
- **Metricbeat**: recolecta métricas del sistema y servicios, como CPU, memoria, uso de red y bases de datos.
- **Packetbeat**: analiza el tráfico de red en busca de información sobre el protocolo y la aplicación, y permite la detección de amenazas y la monitorización del rendimiento.

- **Winlogbeat**: recolecta eventos de registro de Windows, como eventos de seguridad, aplicación y sistema.
- **Auditbeat**: recolecta eventos de auditoría del sistema, como cambios de archivo, acceso a archivos y actividades de usuario.

Es una solución muy útil para la recolección de datos en tiempo real de diferentes fuentes, que facilita la integración con Elasticsearch y Logstash, y ofrece diferentes tipos de agentes especializados para recolectar datos específicos de cada fuente.

Elastic Agents

Elastic Agent: es un solo agente que unifica logs, métricas, datos de seguridad y prevención de amenazas. Es un agente que se añade al servidor e instala y configura "**beats**" por medio de políticas que se guardan en un fichero YAML pero pudiendo configurar y gestionar desde el propio kibana o desde el fichero YAML directamente.

Managed by fleet: Es administrado de manera centralizada por la aplicación "Fleet" en Kibana por medio de políticas ya preconfiguradas. Tiene por defecto integraciones con otros servicios y sistemas, como Suricata. Es la opción recomendada para la gran mayoría de usuarios, porque la instalación de estas integraciones se realiza a través de la UI de Kibana

En la ilustración 13 se muestra algunas de las integraciones que se pueden añadir por medio de Kibana directamente desde la UI.

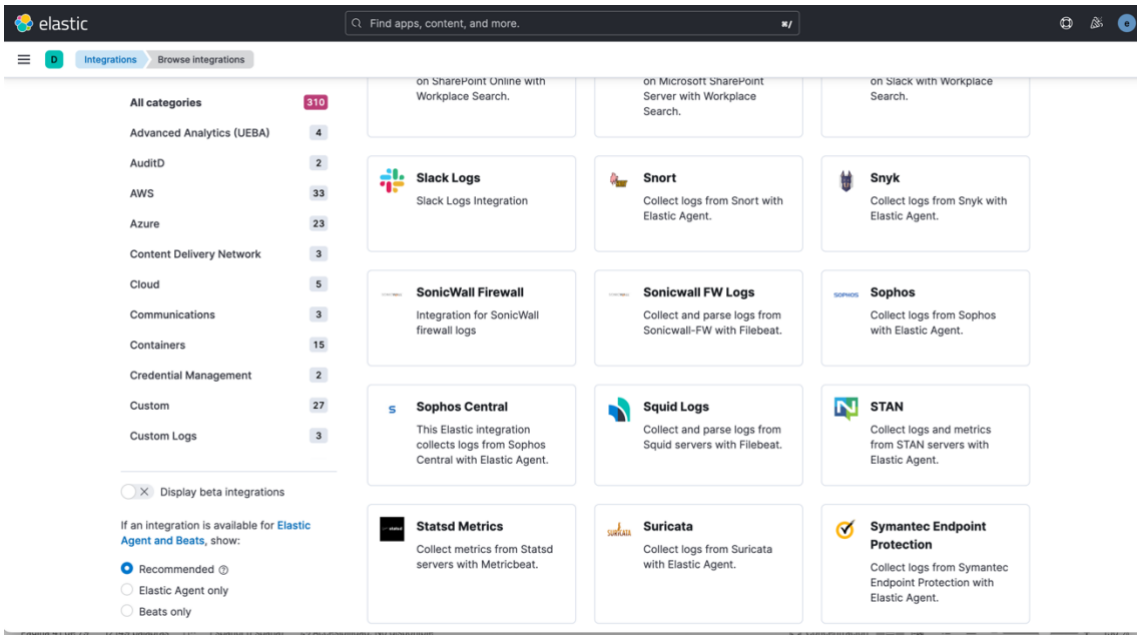


Ilustración 13 Kibana integraciones (Elaboración propia)

Standalone mode: Toda la configuración se aplica por medio de políticas igual que con fleet pero manualmente por medio de un fichero YAML. Está pensada para usuarios

más avanzados ya que permite una gran personalización, pero conlleva un gran esfuerzo inicial.

Beats vs Elastic Agent

Para el envío de los logs de suricata y del servidor a kibana es necesario decidir que método utilizar para ello. Si utilizar los beats que conlleva instalar y configurar individualmente cada beats uno a uno para cada caso de uso o instalar el "Elastic agent" que ofrece ya un agente con todos los beats preconfigurados.

Tras revisar las diferencias de la tabla de comparación que se puede encontrar en su web y realizar una prueba de concepto, se decide utilizar **elastic agent** para el envío de logs y datos hacía kibana por las siguientes razones:

- El agente ofrece logs, métricas y datos ya preconfigurados para el mismo caso de uso. En el caso de los beats se tendría que configurar individualmente.
- La instalación y configuración de integraciones por medio de la UI de ELK. Estas instalaciones por la UI, instala y configura recursos de Elasticsearch, plantillas de índices, paneles, etc… ya preconfigurados. Con Beats se debería de instalar manualmente por el CLI, diseñar las plantillas, paneles, recursos, etc. y configurarlos únicamente por medio de ficheros YAML, añadiendo demasiada complejidad.
- Es la recomendación por defecto de elastic.

2.6 Riesgos de integrar ELK con Raspberry pi

Mirando las capacidades técnicas de procesar y almacenar de la Raspberry-pi definidas al inicio, se tendrá que decidir donde instalar el SIEM del stack de ELK si en la propia Raspberry-pi o en otra máquina con más recursos.

En las tablas 2 y 3 se muestran los requerimientos del stack ELK para una solución empresarial.

Tabla 2 ELK Enterprise - Requerimientos RAM

Memory	**Coordinators**	**Directors**	**Proxies**	**Allocators**
Minimum to install	8 GB RAM	8 GB RAM	8 GB RAM	8 GB RAM
Minimum recommended	16 GB RAM	8 GB RAM	8 GB RAM	128 GB to 256 GB RAM[1]
Small deployment[2]	32 GB RAM	32 GB RAM	16 GB RAM	128 GB RAM
Medium deployment[2]	32 GB RAM	32 GB RAM	32 GB RAM	256 GB RAM
Large deployment[3]	128 GB RAM	128 GB RAM	128 GB RAM	256 GB RAM

Tabla 3 ELK Enterprise – Almacenamiento

Storage	Coordinators	Directors	Proxies	Allocators
Minimum to install	10 GB	10 GB	15 GB	10 GB
Minimum recommended	1:4 RAM-to-storage ratio[1]	1:4 RAM-to-storage ratio[1]	1:4 RAM-to-storage ratio[1]	Enough storage to support the RAM-to-storage ratio[2]

Como depende mucho del uso de que se dé a la red en el escenario doméstico, que no es equiparable a un entorno empresarial como se detalla en las tablas anteriores, durante la fase de implementación se hará una prueba instalando y validando el rendimiento del entorno para ver su viabilidad.

El objetivo de esta prueba de concepto es validar que el rendimiento no afecta al IDS y que la capacidad de almacenamiento de información es viable.

Es evidente que en base a los requerimientos de las tablas anteriores la raspberry no podrá aguantar el rendimiento seguramente. Pero como estos requerimientos mínimos están pensados para una solución empresarial, deberemos de validar si en el entorno doméstico con menos trabajo y menos uso de la red sería posible.

Para la falta de capacidad de la raspberry es posible corregirlo con un simple disco externo conectado por el puerto USB.

Pero en el caso que la prueba de concepto no fuera satisfactoria, habrá que utilizar un segundo servidor para hospedar ELK y habrá que modificar la solución ELK para adaptarse a este nuevo servidor.

3. Implementación de la solución IDS/SIEM

En este primer apartado se va a describir como se va a configurar la red y su topología, y las acciones necesarias que se han realizado para poder montar el laboratorio de ejemplo para dar solución al proyecto

3.1 Configuración de la red

Siguiendo el modelo pasivo elegido en la fase de investigación, la topología de la red se ha estructurado como la ilustración 14.

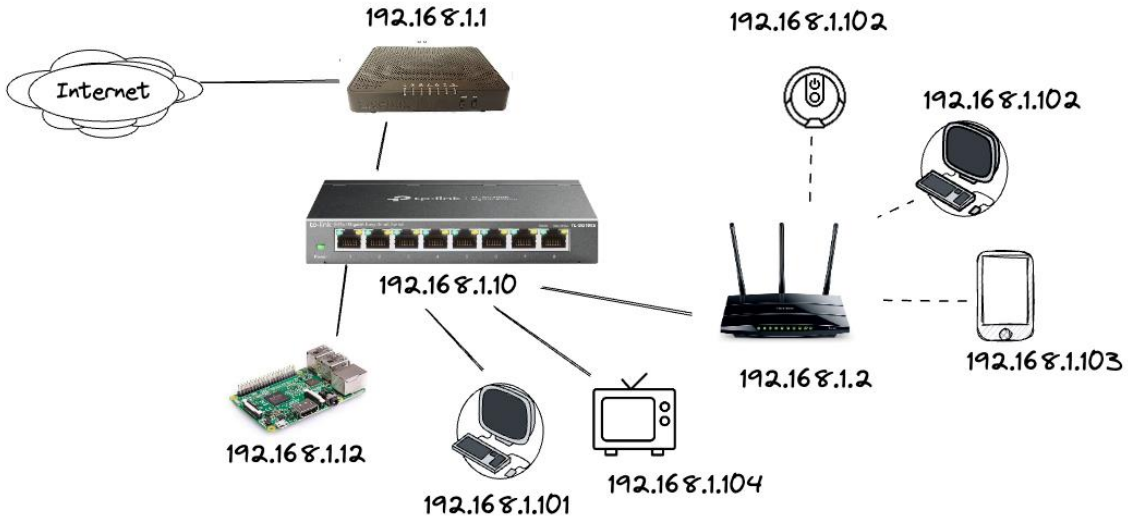


Ilustración 14 Topología de la red (Elaboración propia)

Como se puede ver en la ilustración hay un switch donde están conectados dos routers, la raspberry pi que hace de IDS y dos dispositivos de ejemplo, un ordenador y la televisión de casa.
El resto de los dispositivos se conectan a través de la red wifi del router secundario que hace de punto de acceso.

A continuación, se enumeran los cambios necesarios realizados en cada dispositivo:

1. Router principal: Se ha desactivado la red wifi para que los dispositivos se conecten únicamente a través del punto de acceso wifi del segundo router y a cada dispositivo de la red se le ha dado una dirección estática. Ya que este router será el que se encargará de asignar las IP a los dispositivos.
2. Router secundario: Se ha dado una dirección estática dentro de la misma red que el router principal 192.168.1.2
3. Switch: Se le ha dado una dirección estática 192.168.1.10 y activada la función de port mirroring, de manera que todo el tráfico de la red se envía al puerto donde esté colocada la sonda IDS. Quedando la siguiente configuración de puertos:

    - Puerto 1: Raspberry Pi (Port mirroring)
    - Puerto 2: Router Livebox

- Puerto 3: Router/Access Point
- Puerto 4: Ordenador personal
- Puerto 5: Televisión

La ilustración 15 muestra una captura de la configuración final del switch con la función de port mirroring configurada:

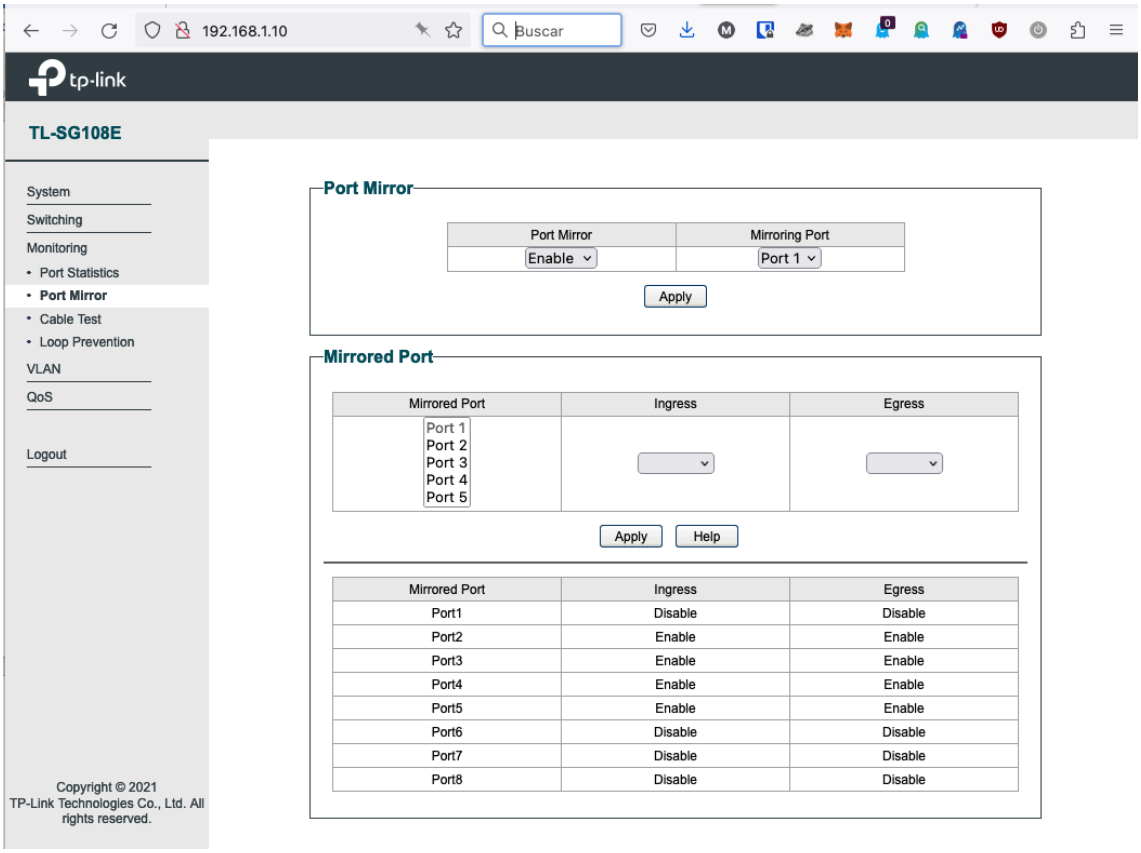


Ilustración 15 Configuración Switch - Port mirroring (Elaboración propia)

Los puertos del al 6-8 están vacíos y se han dejado desactivada la función de clonado, para no saturar demasiado la red.

En la ilustración 16 se puede ver la configuración de la dirección estática definida para el switch.

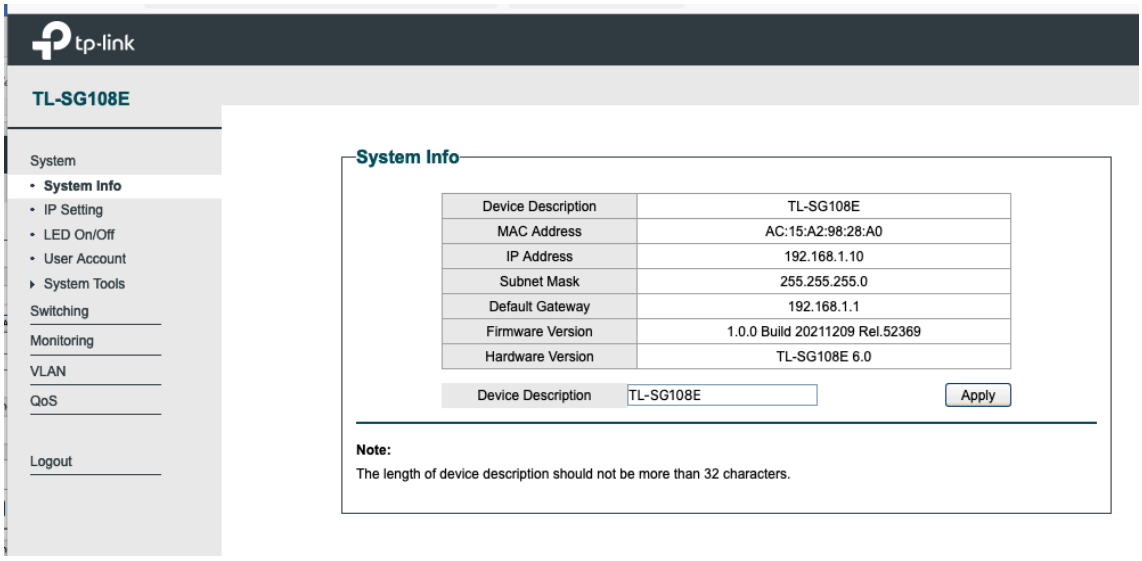


Ilustración 16 Switch. Dirección estática (Elaboración propia)

3.2 Configuración S.O en la Raspberry Pi

Lo primero ha estado configurar el sistema operativo de la Raspberry Pi que da soportarte al IDS.

Para configurar el S.O en la raspberry se elige el sistema operativo *Raspberry Pi OS* (antiguo raspbian) con la versión de escritorio de 924 MB.

Los pasos seguidos para la puesta a punto han estado:

1. Descarga del Sistema operativo

Desde la propia web Raspberry Pi se descarga un software instalador, el Raspberry Pi Imager (Ilustración 17) que permite crear una imagen a medida, el software está disponible tanto para Windows, macOS y Ubuntu. La versión utilizada es la versión de macOS.

Ilustración 17 Raspberry Pi Imager 1.6 (Elaboración propia)

Una vez descargado se ha seleccionado la versión Raspberry Pi OS (64-bit). Es importante reflejar que no es la opción recomendada por Raspberry, que aún sigue

recomendando 32 bits. Pero ha estado necesario seleccionar esta **versión de 64 bits** por compatibilidad de librerías con la suite de ELK tal como veremos más adelante.

La unidad seleccionada para instalar el S.O ha estado una memoria microSD de 32 GB de calidad 10, importante que sea de buena calidad para que la lectura y escritura sea más rápida.

2. Instalación del S.O

Una vez finalizada la creación de la instalación de la imagen en la microSD. Se conecta la raspberry a un monitor, un teclado y un ratón para seguir con la instalación.

La instalación de Raspberry OS ha estado por defecto, seleccionando la siguiente configuración:

Paso 1: Configuración del idioma y zona horaria

Pais: España
Idioma: European Spanish
Timezone: Madrid

Paso 2: Selección del usuario y contraseña con el que se identificará

Username: ids
Password: ******

Paso 3: Selección de la wifi de casa

Paso 4: Selección resolución

Paso 5: Actualización el software: Realiza una actualización de las librerías con las últimas actualizaciones.

Paso 6: Parametrización completada

Paso 7: Reiniciar

Una vez reiniciada la máquina, se identifica con el usuario dado de alta durante la instalación.

Paso 8. Activar Servicio SSH

La distribución viene con el servicio SSH desactivado, el servicio SSH permitirá configurar más cómodamente el IDS desde el ordenador personal.

```
sudo service ssh start
sudo systemctl enable ssh.service
```

Con ello activamos el servicio y lo dejamos activado por defecto al inicio de la máquina. La ilustración 18 muestra el servicio SSH activado.

Ilustración 18 Activar servicio SSH (Ilustración propia)

Para probar la conexión, se comprueba la IP de la máquina en la red actual (wlan0-wifi) **192.168.1.70** que es la ip que el router ha asignado automáticamente.

```
ip a
```

En la ilustración 19 se muestra la ip de la máquina.

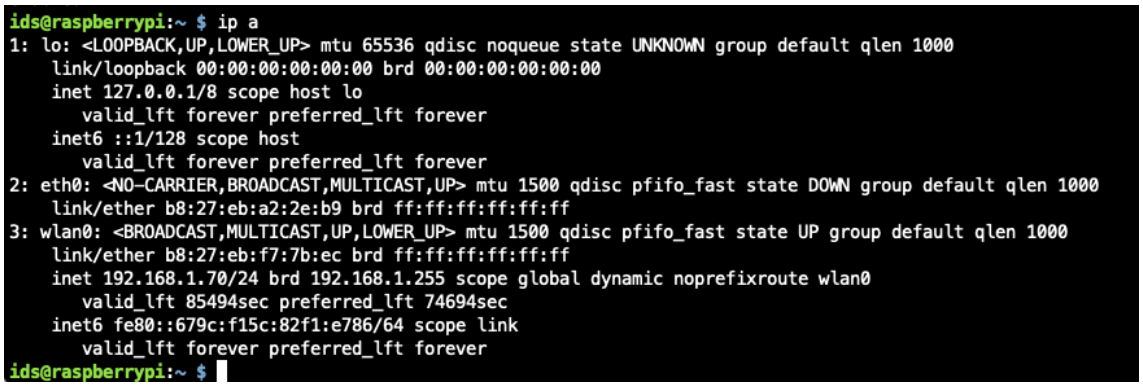


Ilustración 19 IP sonda IDS (Elaboración propia)

Paso 9. Hostname

Se cambia el hostname de la máquina a “idspi” en los ficheros /etc/hosts y /etc/hostname, como se puede ver en la ilustración 20. Dándole un nombre propio e identificándolo mejor dentro de la red.

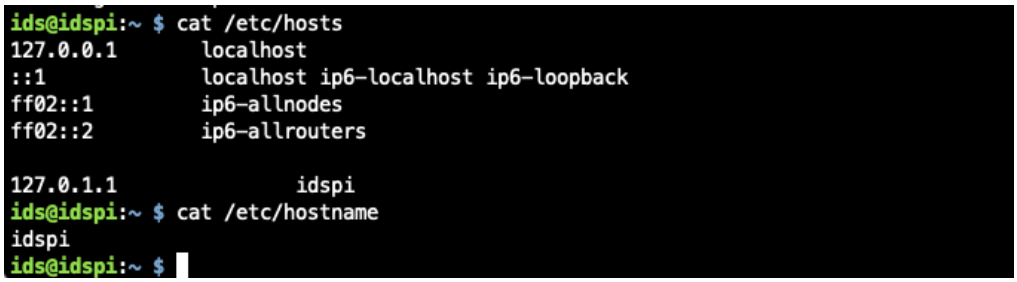


Ilustración 20 hosts IDS (Elaboración propia)

Para poder identificar el IDS se asigna una dirección estática al servicio de DHCPCD del S.O.

Se edita el fichero **/etc/dhcpcd.conf** y se asigna la configuración para las dos redes de red y wifi (wlan0) como se puede ver en la ilustración 21.

```
# Example static IP configuration:
interface eth0
static ip_address=192.168.1.12/24
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.1.1
static domain_name_servers=192.168.1.1 8.8.8.8 fd51:42f8:caae:d92e::1

interface wlan0
static ip_address=192.168.1.11/24
static routers=192.168.1.1
static domain_name_servers=192.168.1.1
```

Ilustración 21 dhcpcd.conf (Elaboración propia)

Una vez aplicado los cambios, se reinicia la máquina con las direcciones IP fijas y asignadas correctamente. En la ilustración 22 se muestra la ip fija de la sonda.

```
ids@idspi:~ $ ip a
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: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether b8:27:eb:a2:2e:b9 brd ff:ff:ff:ff:ff:ff
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether b8:27:eb:f7:7b:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global noprefixroute wlan0
       valid_lft forever preferred_lft forever
    inet6 fe80::679c:f15c:82f1:e786/64 scope link
       valid_lft forever preferred_lft forever
ids@idspi:~ $
```

Ilustración 22 Sonda IP fija (Elaboración propia)

Paso 10. Instalación VNC (Opcional)

La sonda IDS físicamente se deja conectada sin monitores y sin teclados. Por comodidad se procede a activar el servidor VNC que viene preinstalado para un acceso más cómodo si fuera necesario acceder por escritorio remoto.

Se lanza el comando **raspi-config** que permite activar el servicio más cómodamente. Los pasos para seguir son los siguientes:

Interfaces options (3) -> Enable graphical remote Access using RealVNC

Una vez configurado el S.O se sigue con la instalación y configuración del software de IDS Suricata.

3.3 Instalación del software del IDS

A continuación se procede a instalar el software de Suricata como IDS.

Para la instalación de la sonda IDS con suricata principalmente se ha seguido la guía de instalación del proyecto.

Paso 1: Instalar Suricata

Para instalar suricata se actualiza la lista de paquetes del sistema y tener la última versión (Ilustración 23).

```
ids@idspi:~ $ sudo apt-get update
Obj:1 http://raspbian.raspberrypi.org/raspbian bullseye InRelease
Obj:2 http://archive.raspberrypi.org/debian bullseye InRelease
Leyendo lista de paquetes... Hecho
```

Ilustración 23 sudo apt-get update (Elaboración propia)

Se procede a instalar el paquete de suricata y sus dependencias. En la ilustración 24 se muestra las librerías de suricata y sus dependencias.

```
ids@idspi:~ $ sudo apt install suricata
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libauthen-sasl-perl libclone-perl libdata-dump-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfile-listing-perl
  libfont-afm-perl libhiredis0.14 libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhtp2
  libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl
  liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnet1 libnetfilter-log1
  libnetfilter-queue1 libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl oinkmaster perl-openssl-defaults python3-yaml
  snort-rules-default suricata-update
Paquetes sugeridos:
  libdigest-hmac-perl libgssapi-perl libcrypt-ssleay-perl libauthen-ntlm-perl snort | snort-pgsql | snort-mysql libtcmalloc-minimal4
Se instalarán los siguientes paquetes NUEVOS:
  libauthen-sasl-perl libclone-perl libdata-dump-perl libencode-locale-perl libevent-core-2.1-7 libevent-pthreads-2.1-7 libfile-listing-perl
  libfont-afm-perl libhiredis0.14 libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhtp2
  libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl
  liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnet1 libnetfilter-log1
  libnetfilter-queue1 libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl oinkmaster perl-openssl-defaults python3-yaml
  snort-rules-default suricata suricata-update
0 actualizados, 42 nuevos se instalarán, 0 para eliminar y 35 no actualizados.
Se necesita descargar 4.373 kB de archivos.
Se utilizarán 14,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Ilustración 24 sudo apt install suricata (Elaboración propia)

Una vez instalado el paquete, se activa el servicio de suricata en el sistema para que se active cada vez que se inicia la máquina (Ilustración 25).

```
ids@idspi:~ $ sudo systemctl enable suricata.service
Synchronizing state of suricata.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
ids@idspi:~ $
```

Ilustración 25 Activación servicio suricata (Elaboración propia)

Se comprueba que se ha instalado correctamente, revisando el directorio de log de suricata

```
ids@idspi:~ $ ls -l /var/log/suricata/
total 23272
-rw-r--r-- 1 root root 20300220 may  4 13:41 eve.json
-rw-r--r-- 1 root root        0 may  4 11:54 fast.log
-rw-r--r-- 1 root root  3505254 may  4 13:41 stats.log
-rw-r--r-- 1 root root     8208 may  4 12:29 suricata.log
ids@idspi:~ $
```

Ilustración 26 Logs suricata (Elaboración propia)

- **eve.json** Registra todos los eventos que lanza suricata
- **fast.log** Registra únicamente las alertas.
- **suricata.log** Registra el output del servicio.
- **stats.log** Las estadísticas del servicio.

Para comprobar que la sonda IDS está leyendo las conexiones correctamente, se realiza a modo de ejemplo, una conexión **ssh** desde el ordenador personal *192.168.1.145* hacia el IDS *192.168.1.12*

En la ilustración 27 se puede comprobar una captura del eve.json donde suricata informa del evento **SSH** que se ha producido en la red.

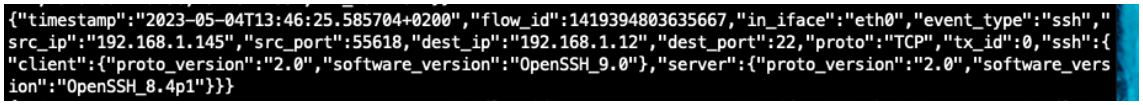


Ilustración 27 eve.json (Elaboración propia)

Para validar otro tipo de evento, como puede ser conectarse a una web por HTTPS. Se realiza una prueba de navegación a https://www.uoc.edu pero desde el móvil personal 192.168.1.213 y por medio de la conexión wifi, y de esta manera validamos otros dispositivos también (Ilustración 28).

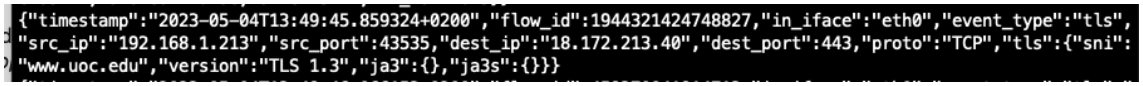


Ilustración 28 eve.json conexión http (Elaboración propia)

Se verifica que está funcionando correctamente y se está registrando el evento y el resto de los dispositivos de la red.

Con esta última prueba también se confirma que el IDS también está registrando correctamente el tráfico de los dispositivos conectados por wifi desde el punto de acceso.

3.4 Configuración del IDS

A continuación, se configurará la red y suricata para que se adapte a las necesidades específicas del proyecto.

Se definen las políticas de detección y se establecen los umbrales de alerta. Al final de la sección se configura las alertas que se envían al SIEM para su posterior análisis.

Configuración de suricata

Toda la configuración de suricata se realiza desde el fichero de configuración que se encuentra en /etc/suricata/suricata.yaml en formato YAML

Paso 1: Información la red

Se añade la variable de red de casa en el rango 192.168.1.0/24:

```
vars:
        address-groups:
                HOME_NET: “[192.168.1.0/24]”
```

Se revisa la configuración por defecto y se deja tal cual a excepción del apartado de reglas.

Por defecto suricata viene sin ninguna regla activadas, por lo que es necesario especificar los ficheros con las reglas. Estas reglas se añaden en la sección "*rules-files*"

Para el proyecto se han dejado activadas las siguientes listas:

- /etc/suricata/rules/suricata.rules. Las reglas personalizadas creadas específicas para el proyecto.
- /etc/suricata/rules/dns-events.rules. Reglas básicas tráfico dns
- /etc/suricata/rules/http-events.rules. Reglas básicas tráfico http
- /etc/suricata/rules/tls-events.rules. Reglas básicas tráfico https
- /etc/suricata/rules/smb-events.rules. Reglas básicas de tráfico samba

Actualización de las reglas

Suricata aparte de las reglas o firmas que vienen por defecto, también tiene una gestión de listas de reglas que periódicamente se pueden descargar y aplicar.

La forma de actualizar el repositorio es lanzar el comando "**suricata-update**" que se encarga de gestionar los repositorios de reglas de suricata, tanto comerciales como open source.

Por defecto suricata si no se activa un repositorio en especial, configura el repositorio Emerging Threats en su versión libre que se guardan en /var/lib/suricata/rules/suricata.rules

En la ilustración 29 se muestra las 42.029 reglas descargadas del repositorio.

Ilustración 29 suricata-update (Elaboración propia)

Es importante destacar que en este proyecto no ha sido posible añadir la lista de reglas de Emerging Threats en el dispositivo y poder probarlas.

Por falta de recursos en el dispositivo y tantas reglas en la propia lista de Emerging Threats. Suricata no ha podido gestionarlas correctamente y en el momento de activarlas dejaba inservible la máquina, sin recursos y bloqueada. Por lo que sería necesario un dispositivo con mucha más RAM.

3.5 Pruebas y validación del IDS

Para verificar que el IDS está funcionando correctamente, se simulan diferentes tipos de ataques para validar que el IDS es capaz de detectar y alertar

Se crea una regla propia que comprueba si se realizan demasiadas conexiones SSH seguidas en la red

Se edita el fichero /etc/suricata/rules/suricata.rules y se añade la siguiente regla que alertará cuando detecte cualquier conexión por el puerto 22 que tenga como destino cualquier ip de la red de casa y realice más de 5 conexiones en menos de 30 segundos (Simulando un ataque de fuerza bruta)

```
alert tcp any any -> $HOME_NET 22 (msg:"Possible SSH brute forcing!";
classtype: misc-attack; flags: S+; threshold: type both, track by_src,
count 5, seconds 30; sid:1000001; rev: 1;)
```

Se crean varias conexiones seguidas SSH desde la máquina personal (192.168.1.101) hacia la sonda IDS.

Se comprueba que suricata genera la alerta correctamente en el log **/var/log/suricata/fast.log**

```
05/08/2023-14:49:59.916411  [**] [1:1000001:1] Possible SSH brute
forcing! [**] [Classification: Misc Attack] [Priority: 2] {TCP}
192.168.1.101:52668 -> 192.168.1.12:22
```

3.6 PiHole

En el apartado 2 se ha demostrado que la sonda IDS no es posible utilizarla como IPS real, porque influye en el rendimiento de la red y no es óptimo. Para controlar mejor el tráfico de la red se ha instalado un servidor de DNS propio para gestionar todo el tráfico de DNS. De esta manera siendo la sonda IDS como proveedor de DNS se permite bloquear ciertos dominios si fuera necesario.

Para ello se ha elegido utilizar el proyecto PiHole, pensado principalmente como bloqueador de publicidad y diseñado para una raspberry Pi.

PiHole añade protección en toda la red, porque permite bloquear las conexiones antes que lleguen a los dispositivos. Permitiendo gestionar listas de páginas webs prohibidas, bloqueando su acceso directamente desde la sonda IDS, también permite bloquear anuncios en los dispositivos regulares y en los no tradicionales, como aplicaciones móviles o televisores inteligentes.

Además, se mejora el rendimiento de la red. Dado que los anuncios se bloquean antes de que se descarguen, el rendimiento de la red mejora y da la sensación de más rapidez en la navegación web.

En el anexo 1 es posible revisar los pasos seguidos para su instalación y configuración.

3.7 Mejoras rendimiento

A continuación, se detallan los cambios realizados para poder mejorar el rendimiento de la raspberry PI.

- Se ha incrementado la memoria swap a 1024 MB en /etc/dphys-swapfile. Que por defecto viene configurado con 100MB.
- También se ha procedido a desactivar el servidor VNC y activarlo solamente cuando es necesario y así ahorrar un poco más de memoria.

3.7 Instalación SIEM

En esta sección se procede a instalar el SIEM y configurar la suite completa de elastic.

La suite ELK se compone de tres herramientas principalmente: Elasticsearch, Logstash y Kibana. Estas herramientas funcionan juntas para recopilar, analizar y visualizar datos de registro en tiempo real.

Tras una prueba de concepto realizando la instalación en la raspberry Pi, la instalación de la suite se ha tenido que migrar fuera de la sonda IDS, porque no podía soportarlo por requerimientos, principalmente por la falta de memoria RAM como se preveía en el estudio inicial.

Se ha procedido a instalar el SIEM en la máquina de escritorio personal (192.168.1.101)

Docker-elk

Para su instalación se ha elegido el proyecto "Docker-elk". Docker-ELK es un proyecto de Docker Compose que facilita la instalación y configuración de manera cómoda con contenedores dockers pre-configurados. Ahorrando tiempo en su configuración e instalación.

Los requerimientos mínimos para utilizarlo en la máquina son los siguientes:

- Docker Engine 18.06.0 o superior
- Docker Compose 1.28.0 or superior (incluyendo compose v2)
- 1.5 GB of RAM

Su instalación es sencilla solo hay que seguir los siguientes pasos:

Paso 1. Clonar el repositorio de docker-elk:

```
git clone https://github.com/deviantony/docker-elk.git
```

Paso 2. Levantar el proyecto docker compose

```
docker-compose up -d
```

Una vez arrancado Docker compose expone los siguientes En este momento se habrá levantado los siguientes servidores y en los siguientes puertos:

Logstash

- 5044: entrada de beats (elastic agent)
- 50000: TCP input
- 9600: API (http://localhost:9600/)

Elasticsearch

- 9200: HTTP (http://localhsot:9200)
- 9300: TCP transport

Kibana:

5601: Kibana (http://localhost:5601)

Paso 3. Acceder Kibana por http://localhost:5601 como en la ilustración 30 e identificarse con el usuario y contraseña definido en el fichero .env

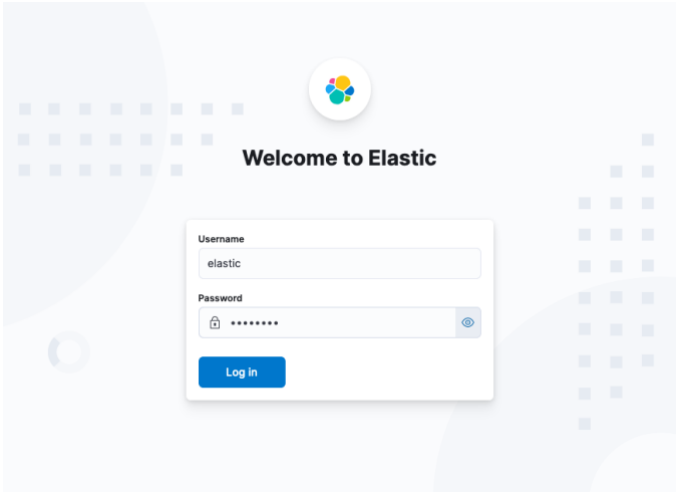


Ilustración 30 Kibana login (Elaboración propia)

En esta guía no contempla la personalización de los servicios configurados en el docker compose, porque la instalación básica tiene lo suficiente para configurar el SIEM.

En el caso que fuera necesario añadir o modificar configuración en cualquiera de los servicios, habría que editar el fichero de configuración para cada servicio, cada uno tiene su propio fichero de configuración dentro de la carpeta config.

A modo de ejemplo para editar y acceder a la configuración de kibana, se accede desde *kibana/config/kibana.yml*

3.8 Configuración SIEM

A continuación, se procede a configurar las kibana para que pueda integrarse con el IDS Suritcata, configurar un Fleet Server e instalar el elastic agent en el IDS que enviará las alertas generadas.

Fleet Server

Un fleet server es un componente que centraliza y gestiona los agentes de elastic que se utilizan para enviar los datos de cada host a elastic search. Al instalar un fleet server automáticamente instala un agente que enviará los datos al SIEM.

Se añade el fleet server siguiendo los siguientes pasos:

Paso 1: Ir a http://localhost:5601/app/fleet/agents > “Add Fleet Server” > Get started with Fleet Server y añadir el host ya configurado (http://fleet-server:8200) > Continue

Paso 2: Install Fleet Server to a centralized host

Ahora se procede a descargar e instalar el agente de elastic que envía los datos a elasticsearch.

```
ids@idspi:/opt $ sudo curl -L -O
https://artifacts.elastic.co/downloads/beats/elastic-agent/elastic-
agent-8.7.1-arm64.deb

ids@idspi:/opt $ sudo dpkg -i elastic-agent-8.7.1-arm64.deb

(Leyendo la base de datos ... 103990 ficheros o directorios instalados
actualmente.)
Preparando para desempaquetar elastic-agent-8.7.1-arm64.deb ...
Desempaquetando elastic-agent (8.7.1) sobre (8.7.1) ...
Configurando elastic-agent (8.7.1) ...
found symlink /usr/share/elastic-agent/bin/elastic-agent, unlink
create symlink /usr/share/elastic-agent/bin/elastic-agent to
/var/lib/elastic-agent/data/elastic-agent-10dc6a/elastic-agent
```

Una vez finalizado debe mostrar el mensaje como en la ilustración 31.

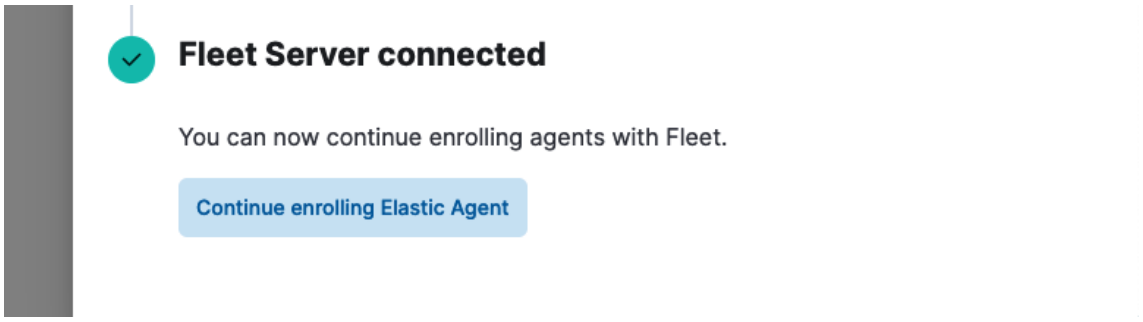


Ilustración 31 Confirmación Fleet server conectado (Elaboración propia)

Paso 3. En este punto se registra el agente con la dirección del fleet server configurado en el Paso 1 y el token que aparece en el paso 2

En este caso el fleet server se ha configurado en 192.168.1.101 que es la máquina donde se ha levantado Docker-elk

```
ids@idspi:/opt $ sudo elastic-agent enroll \
  --fleet-server-es=http://192.168.1.101:9200 \
  --fleet-server-service-
token=AAEAAWVsYXN0aWMvZmxlZXQtc2VydmVyL3Rva2VuLTE2ODQxNDkwNDkzNjE6NFVh
dkk3dG9RSENSSTF6ZnlVMFJkZw \
  --fleet-server-policy=fleet-server-policy
This will replace your current settings. Do you want to continue?
[Y/n]:..
```

Paso 4. Se activa el servicio para que se inicie el agente cada vez que se arranca el IDS

```
ids@idspi:/opt $ sudo systemctl enable elastic-agent

Synchronizing state of elastic-agent.service with SysV service script
with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable elastic-agent
Created symlink /etc/systemd/system/multi-user.target.wants/elastic-
agent.service → /lib/systemd/system/elastic-agent.service.
```

Paso 5. Se inicia el elastic-agent

```
ids@idspi:/opt $ sudo systemctl start elastic-agent
```

En este momento el agente ya está conectado con kibana y el servidor fleet. y debe estar enviando datos de la sonda hacía el servidor SIEM correctamente

Al haber añadido el servidor fleet y el agente, kibana automáticamente añade unas integraciones por defecto, bastante útil con logs, paneles, métricas ya preconfiguradas enviándose desde el agente.

- System: Información sobre el servidor
- Fleet server: Información del fleet server
- Elastic Agent: Información sobre todos los agentes instalados en el fleet server.

A continuación, se muestra ciertos paneles y sus respectivas direcciones

En la ilustración 32 muestra el panel con toda la información sobre los hosts donde hay un agente instalado. Tiene como título [Metrics System] Host overview”

URL: http://localhost:5601/app/dashboards#/view/system-79ffd6e0-faa0-11e6-947f-177f697178b8?_g=(filters:!(),refreshInterval:(pause:!t,value:60000),time:(from:now-15m,to:now))

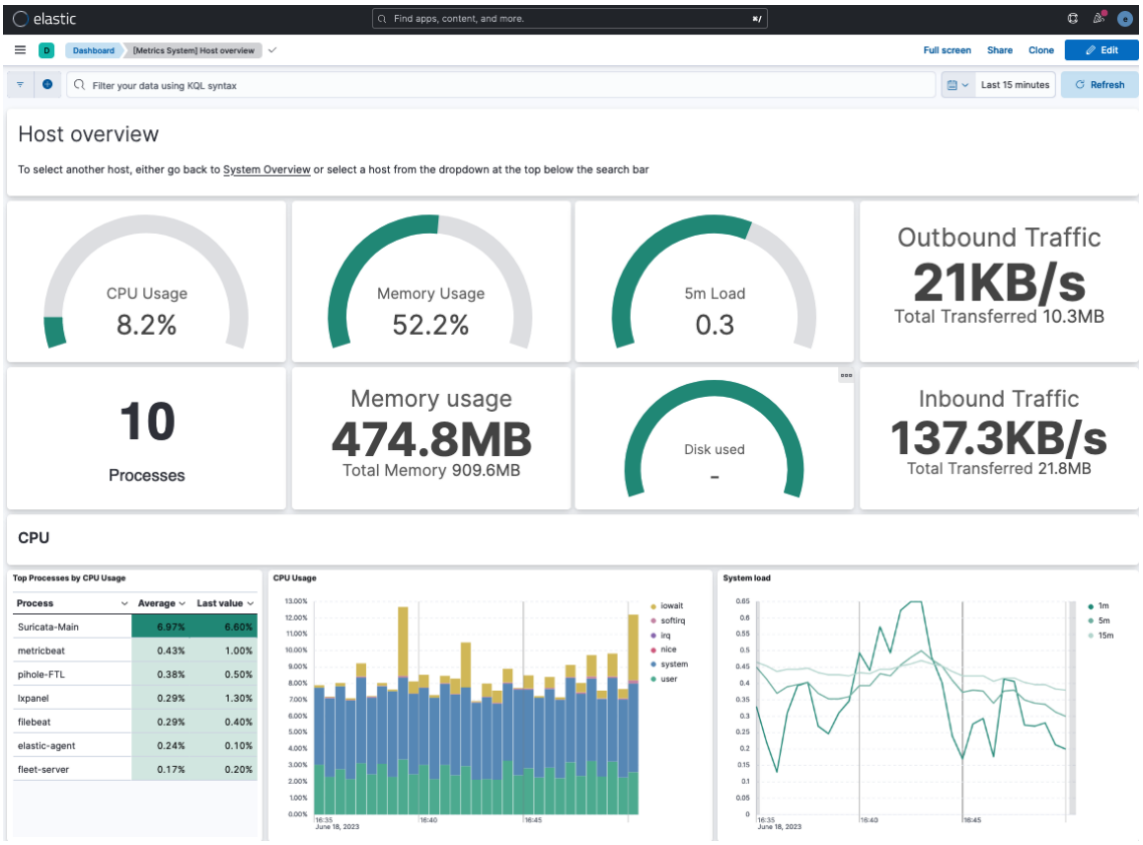


Ilustración 32 Ilustración 32 [Metrics System] Host overview (Elaboración propia)

En la ilustración 33 se muestra el panel con la información del syslog, identificaciones por SSH, comandos sudo, usuarios nuevos y grupos. Tiene como título "[Logs System] Syslog dashboard"

URL: http://localhost:5601/app/dashboards#/view/system-Logs-syslog-dashboard?_g=(filters:!(),refreshInterval:(pause:!t,value:60000),time:(from:now-15m,to:now))

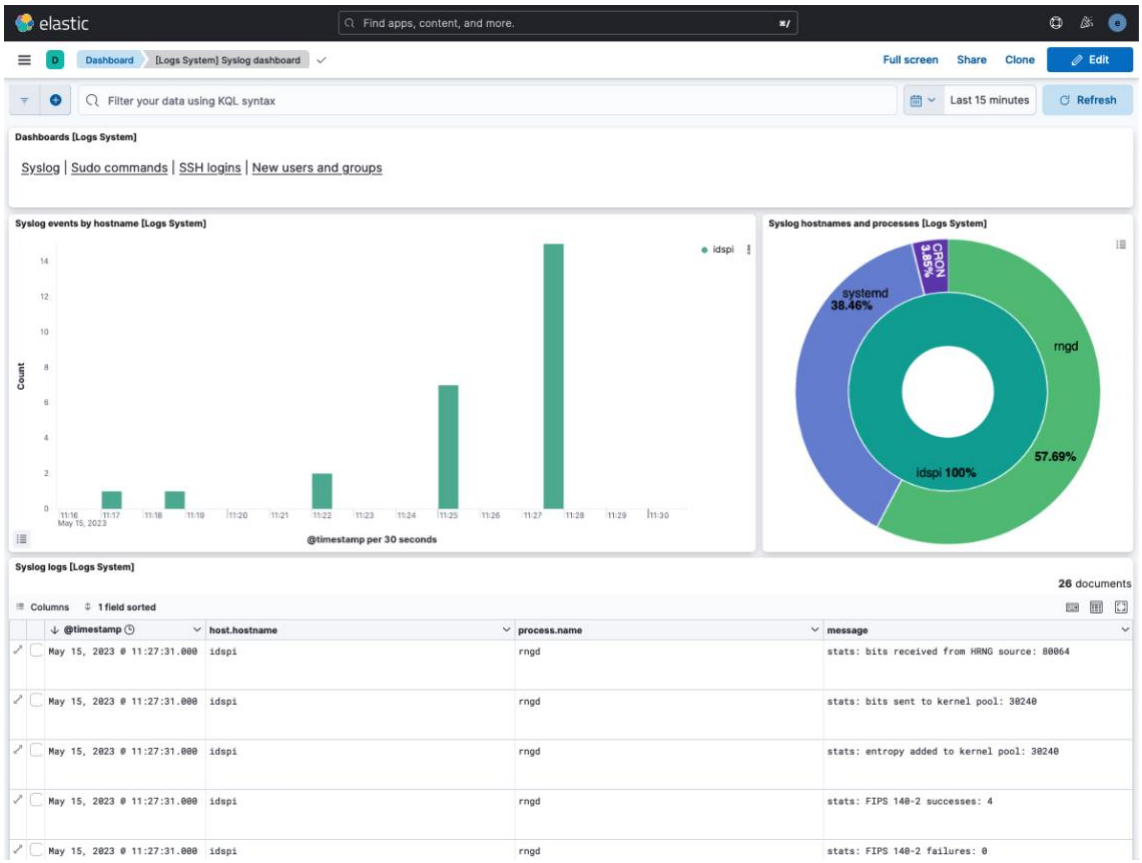


Ilustración 33 [Logs System] Syslog dashboard (Elaboración propia)

En la ilustración 34 se muestra el panel con el resumen del estado del elastic agent instalado en la sonda. Este panel se puede encontrar con el título "[Elastic Agent] Overview"

URL: http://localhost:5601/app/dashboards#/view/elastic_agent-a148dc70-6b3c-11ed-98de-67bdecd21824?_g=(filters:!(),refreshInterval:(pause:!t,value:60000),time:(from:now-15m,to:now))

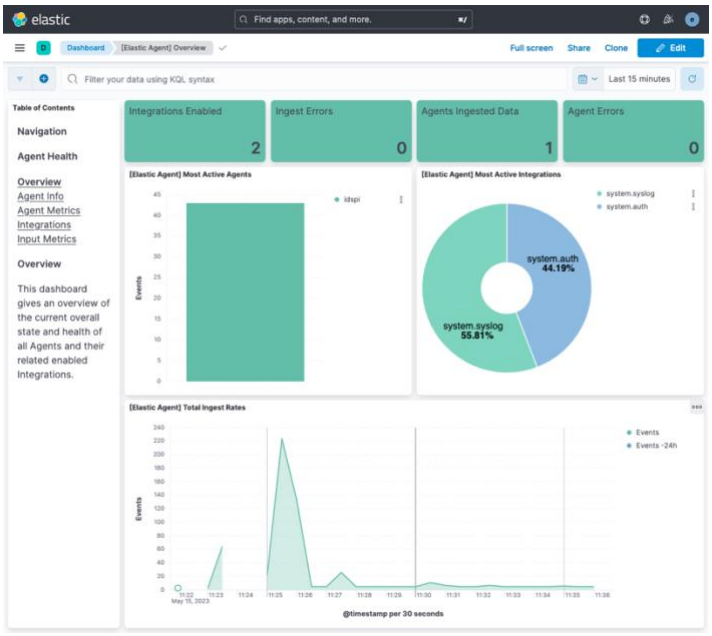


Ilustración 34 [Elastic Agent] Overview (Elaboración propia)

Integración Suricata

Una vez instalado el agente en la sonda y configurado el Fleet server, se procede a instalar la integración con Suricata para poder leer desde el SIEM las alertas del IDS

Para añadir la integración de suricata hay que seguir los siguientes pasos

Paso 1. Seleccionar “Add Integrations” disponible desde el menú de navegación

Paso 2. Seleccionar Suricata” y “Add Suricata”.

Paso 3. “Configure integration” no es necesario realizar modificaciones

Paso 4. Añadir la poliza del agente. Where to add this integration? > Existing hosts > Agent policy > Fleet Server Policy

Paso 5. Save and continue

Paso 6. Confirmar de nuevo

En la ilustración 35 se muestra el formulario con la configuración final.

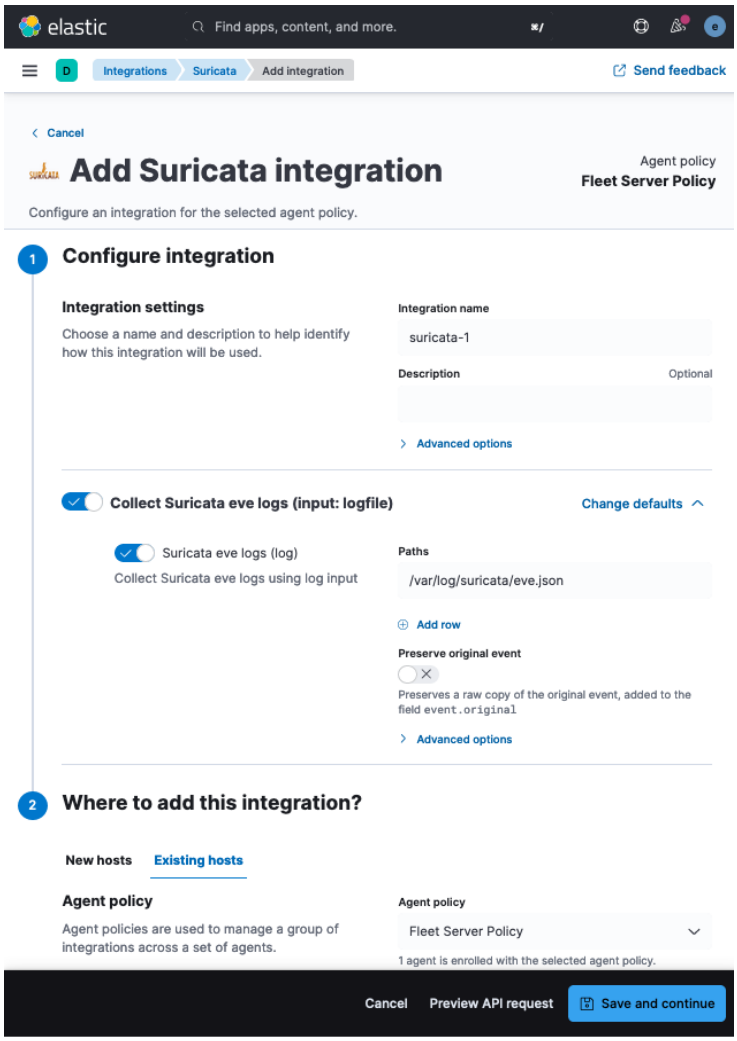


Ilustración 35 Integración Suricata (Elaboración propia)

Una vez añadida la integración, se añaden dos paneles nuevos para ver los eventos y alertas de Suricata desde Kibana configurados.

En la ilustración 36 se muestra una captura del panel de eventos y alertas de Suricata

Ilustración 36 Suricata Dashboard - Events (Elaboración propia)

Se puede apreciar todos los mensajes que llegan del log de suricata eve.json y muestra diferentes gráficos dependiendo del tipo de evento, origen, destino de la conexión.

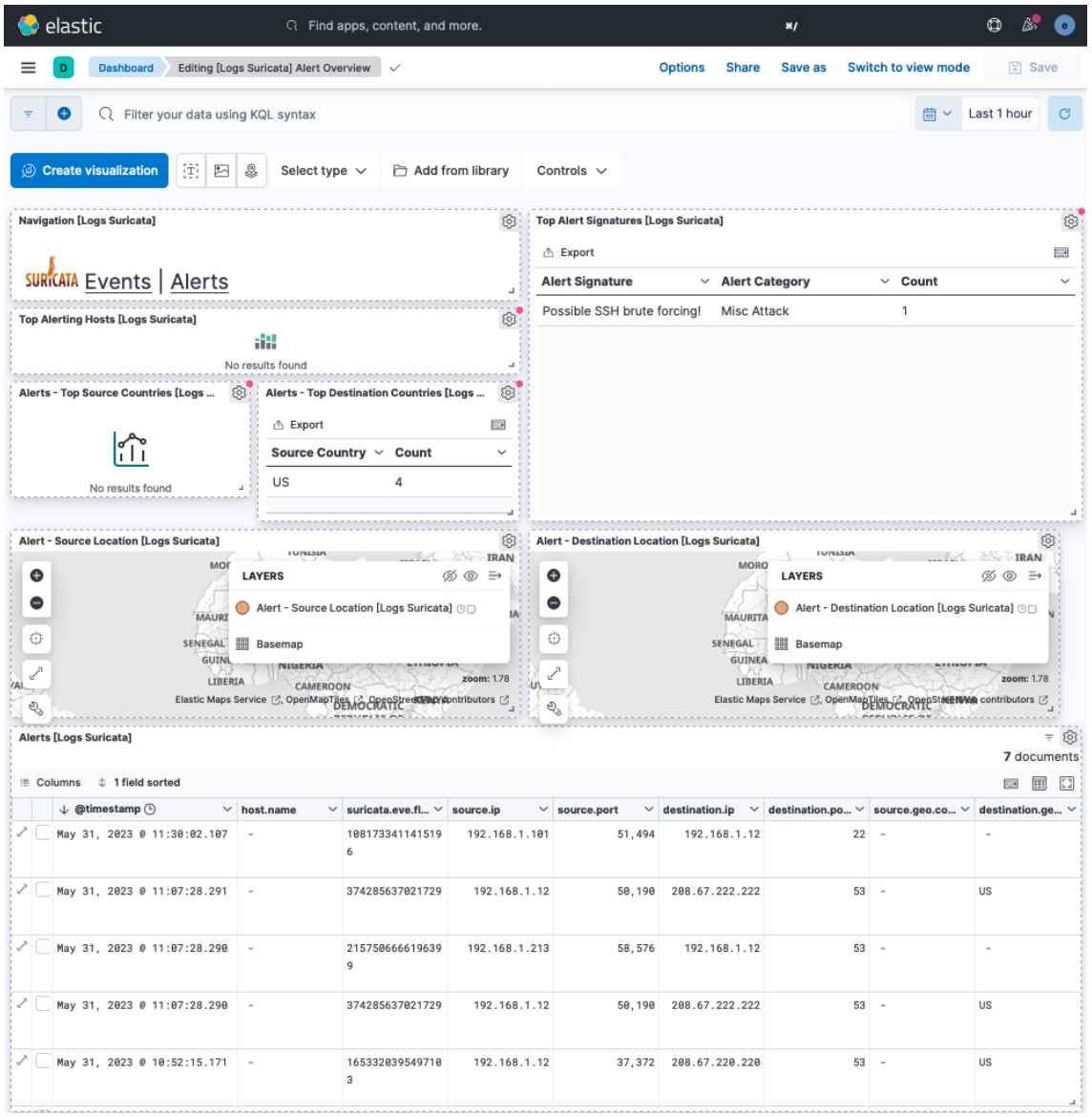


Ilustración 37 Suricata Dashboard - Alertas (Elaboración propia)

En la ilustración 37 se puede ver las alertas generadas por suricata, es importante que la alerta tenga una clasificación sino no aparecerá en la lista de alertas.

3.9 Configuración notificaciones en el SIEM

Una vez integrado el IDS de Suricata con ELK, se procede a configurar ciertas alertas que procedan del IDS para que el SIEM genere alertas y pueda usarse como medio para notificaciones.

Se crea una regla en kibana que envía una notificación a la herramienta “Slack” cada vez que se detecte una alerta de Suricata. En este contexto se ha elegido la herramienta Slack por preferencia, pero pueden configurarse para otro tipo de herramientas de mensajería o incluso notificar por correo electrónico.

Para este escenario se utiliza la alerta generada en el apartado de “Pruebas de IDS” donde se genera cuando detecta un posible ataque de fuerza bruta por el puerto 22

```
alert tcp any any -> $HOME_NET 22 (msg:"Possible SSH brute forcing!";
classtype: misc-attack; flags: S+; threshold: type both, track by_src,
count 5, seconds 30; sid:1000001; rev: 1;)
```

que genera la siguiente alerta en Suricata

```
05/08/2023-14:49:59.916411  [**] [1:1000001:1] Possible SSH brute
forcing! [**] [Classification: Misc Attack] [Priority: 2] {TCP}
192.168.1.101:52668 -> 192.168.1.12:22
```

Para crear la regla en kibana y generar la notificación cada vez que se detecte alertas del tipo 1000001 se realiza los siguientes pasos:

Paso 1. Ir a Security > Alerts > Manage rules > Create new rule

Paso 2. Custom query > Index Patterns (Por defecto)

Paso 3. En la custom query se añade la siguiente consulta **rule.id : "100001"** que pertenece al **ID de la alerta definido en suricata**

Paso 4. Añadir un nombre, descripción a la regla y el nivel de severidad

Paso 5. Cada cuanto se quiere lanzar la alerta

Paso 6. Añadir una acción cada vez hora que detecta la alerta y selecciona la aplicación Slack

Paso 7. Añadir un conector, se añade un nombre descriptivo y el webhook URL que se puede generar desde https://my.slack.com/services/new/incoming-webhook

Paso 8. Create & Enable Rule

En la ilustración 38 se muestra un resumen de como quedaría la regla.

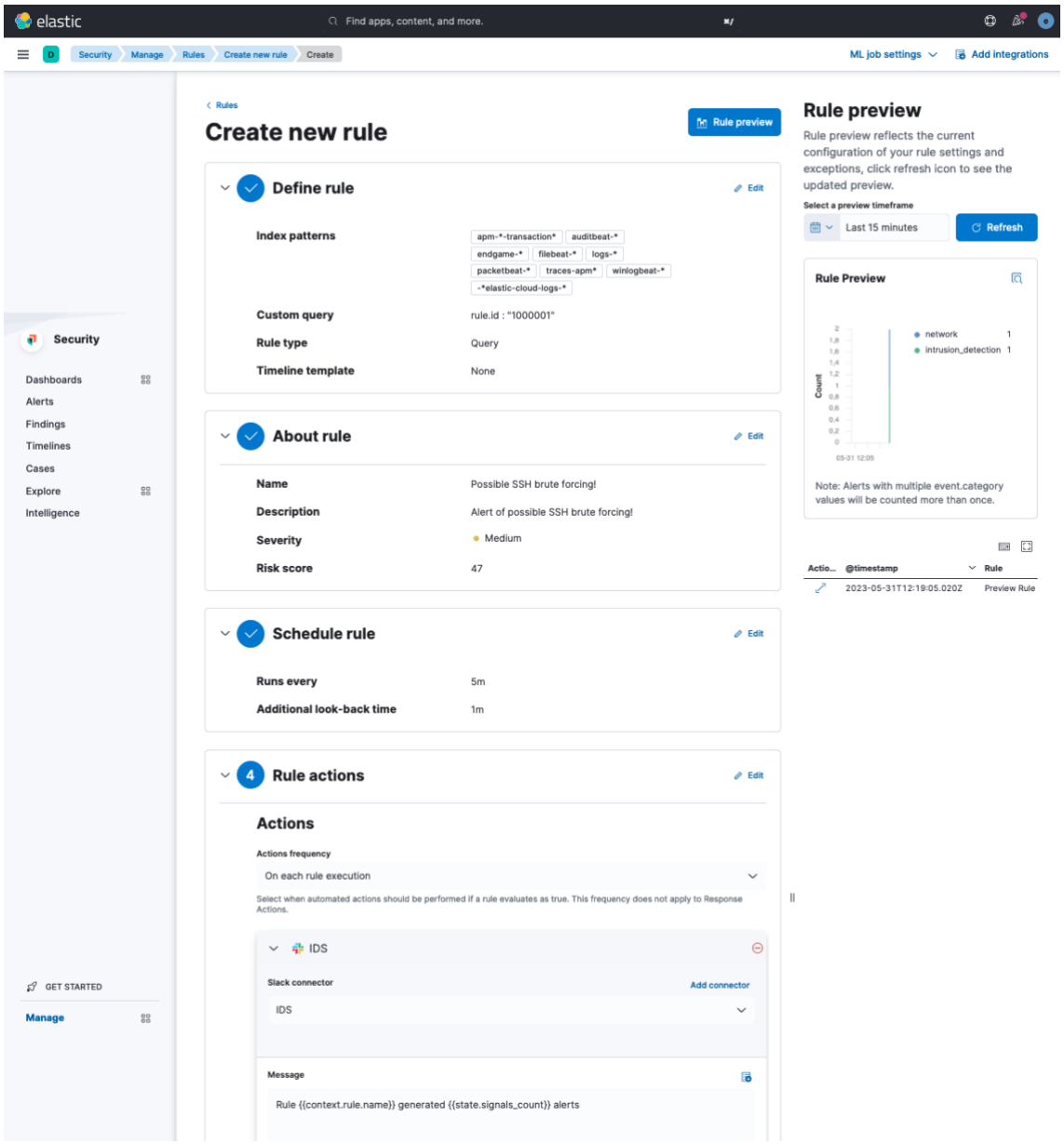


Ilustración 38 Alta regla SIEM (Elaboración propia)

3.10 Pruebas y validación del SIEM

A continuación, se procede a validar la integración final entre la sonda IDS y el servicio SIEM por completo, donde se genera el siguiente escenario:

- Se crean N conexiones por SSH que hacer saltar la alerta en suricata y registrando la alerta en el log de suricata /var/log/eve.log
- El elastic agent del IDS envía el log de suricata a elastic search para ser procesado
- Kibana registra la alerta en su panel de control
- Kibana hace saltar la regla generada porque el registro coincide con la regla definida y envía una notificación por slack al canal definido.

- Slack recibe la notificación y envía notificación por medio de la app, en el caso que esté desconectado de slack, se envía un correo electrónico a la cuenta.

En las ilustraciones 39 y 40 se muestra una captura de la notificación recibida por el Slack y la notificación del correo electrónico por Slack.

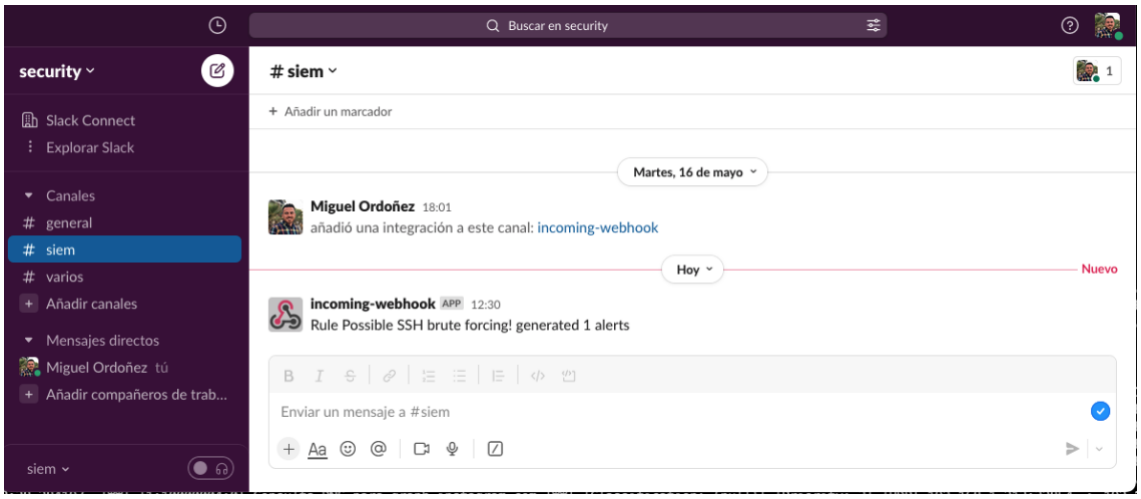


Ilustración 39 Notificación por Slack (Elaboración propia)

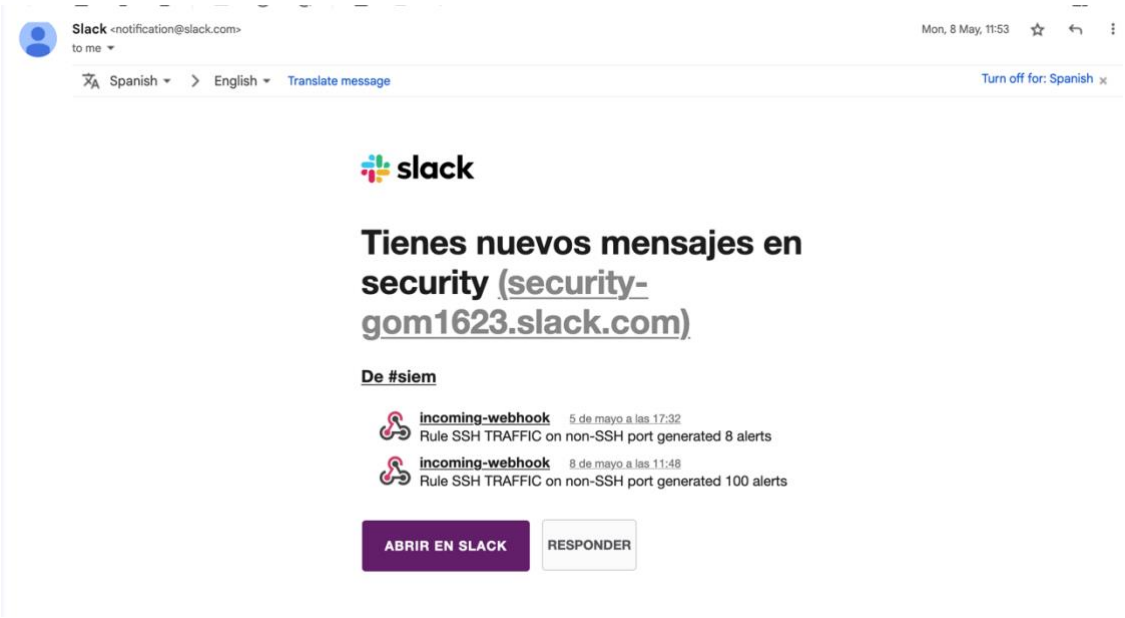


Ilustración 40 E-mail notificación (Elaboración propia)

Reglas control parental y malware

A continuación, se muestran más reglas de suricata que se han creado a medida para controlar la red.

Envío de mensajes por Whatsapp

En esta regla está pensada para control parental, por ejemplo para controlar el uso de la herramienta de Whatsapp.

Whatsapp realiza consultas de dns cada vez que se envía imágenes o mensajes a través de él. Se aprovecha este mecanismo para detectar la actividad del uso de Whatsapp en cualquier dispositivo conectado en la red.

```
alert dns any any -> any any (msg:"Consulta DNS para
graph.instagram.com"; classtype: misc-activity; dns.query;
content:"graph.instagram.com"; sid:1000004;)

alert dns any any -> any any (msg:"Consulta DNS para Whatsapp";
dns.query; classtype: misc-activity; content:"g.whatsapp.net";
sid:10000005;)

alert tls any any -> any any (msg:"Imagen enviada/recibida por
WhatsApp"; classtype: misc-activity; tls.sni;
pcre:"/media.*\.whatsapp\.net/"; sid:1000006;)
```

Se crea una regla en el SIEM para que genere una alerta y notifique por medio de Slack cada vez que detecte actividad por Whatsapp en unas horas determinadas (desde las 22:00 hasta las 08:00) De esta manera permite saber si alguien de la casa ha utilizado whatsapp fuera del horario establecido. La ilustración 41 muestra la regla configurada para el control parental.

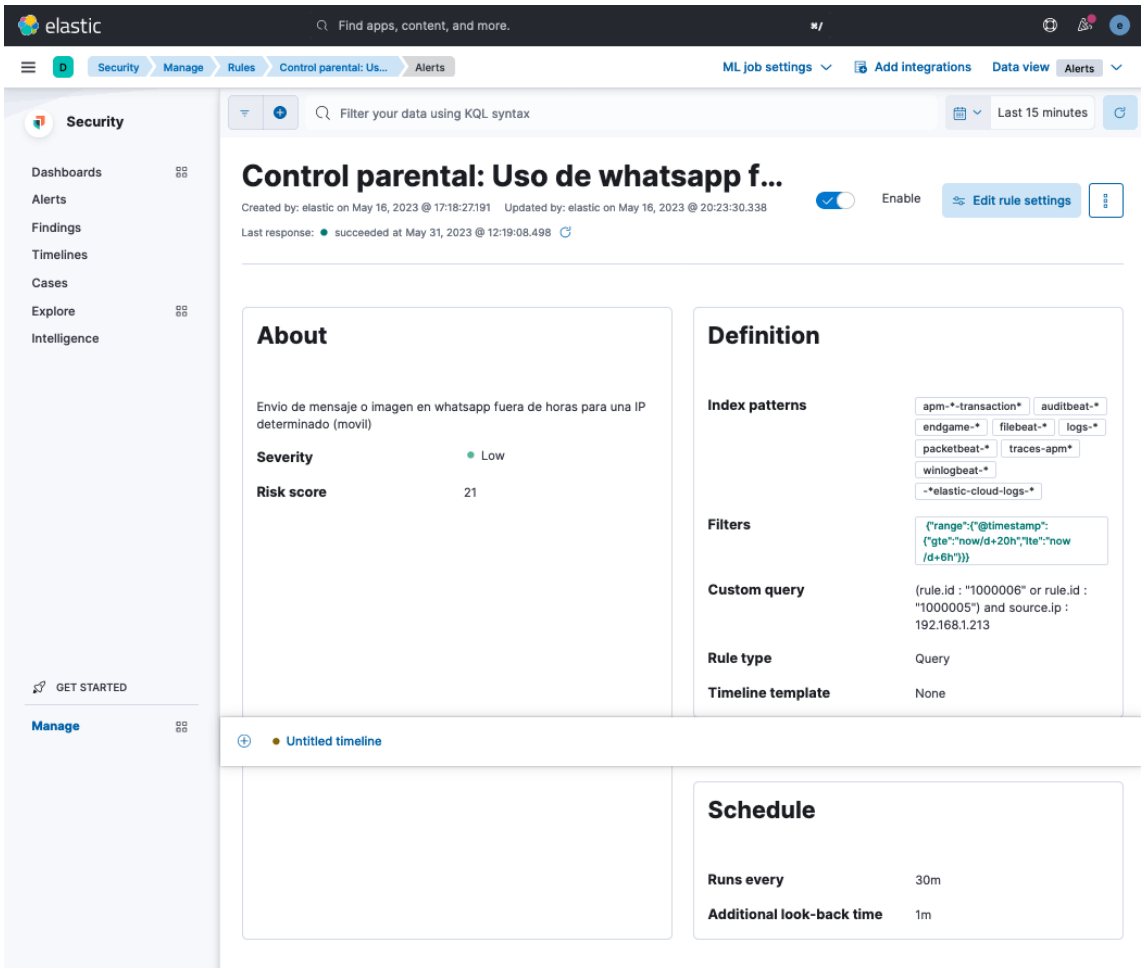


Ilustración 41 Control parental Regla (Elaboración propia)

Descarga de malware

En el caso que alguien en la red descargue un archivo .exe por http hará saltar la siguiente alerta

```
alert http any any -> any any (msg:"Posible descarga de malware";
classtype: suspicious-filename-detect; flow:established,to_server;
content:"GET"; http_method; content:".exe"; http_uri; sid:1000002;)

05/31/2023-13:50:30.102923  [**] [1:1000002:0] Posible descarga de
malware [**] [Classification: (null)] [Priority: 3] {TCP}
192.168.1.101:51742 -> 192.168.1.12:8080
```

4. Conclusiones

En este trabajo, se ha llevado a cabo la integración del IDS Suricata con el SIEM de la suite ELK de elastic, con el objetivo de mejorar la capacidad de detectar y dar respuesta ante posibles amenazas y/o ataques en una red doméstica. A lo largo del desarrollo de este trabajo, se ha realizado la instalación y configuración adecuada de los componentes necesarios, donde se ha probado la funcionalidad de esta integración y se han obtenido resultados satisfactorios y comprobado sus limitaciones.

En este apartado de conclusiones, se analizan los principales logros y aprendizajes derivados de este proyecto, así como las implicaciones y beneficios que ofrece esta integración en una red doméstica.

Se ha demostrado la eficiencia de Suricata como un IDS versátil y potente en la detección y control del tráfico de red. La capacidad de crear firmas y reglas personalizadas, con la activa comunidad de código abierto que respalda a Suricata, permitiendo mantener el IDS actualizado.

La integración de Suricata con Elastic, se ha demostrado como una opción viable y de bajo coste para implementar también en entornos doméstico. La combinación de estas herramientas ofrece una mejora significativa en la capacidad de detección y respuesta ante posibles amenazas.

El SIEM de Elastic ha demostrado ser una herramienta potente para la creación de paneles de control, no solo en el ámbito de SIEM, sino también en otros tipos de soluciones. Sin embargo, se ha identificado que la versión destinada al proyecto no es óptima para su ejecución en una Raspberry Pi 3 debido a limitaciones de rendimiento. Se recomienda utilizar una máquina con mayor potencia para garantizar un rendimiento óptimo.

Si bien la solución propuesta puede ser efectiva para el control parental, es importante tener en cuenta que un enfoque basado en un router con funciones específicas de control parental podría ser más eficiente y adecuado en este escenario.

Se destaca que la mayoría de las conexiones en la red están cifradas, lo que limita la capacidad del IDS para inspeccionar el contenido de dicho tráfico. Lo que destaca la importancia de considerar otras capas de seguridad complementarias.

Resumiendo, la integración de Suricata con el SIEM de Elastic ha demostrado ser una solución efectiva y accesible para mejorar la seguridad en redes domésticas. Aunque es necesario considerar las limitaciones de rendimiento, así como complementar esta solución con enfoques adicionales para abordar las conexiones cifradas y otros desafíos de seguridad.

4.1 Evaluación de los objetivos

A continuación, se listan los objetivos que se definieron en la planificación inicial y se realiza una valoración sobre ellos.

Objetivos primarios:

- **Realizar un análisis de las amenazas y riesgos existentes en la red doméstica.**

Se ha podido comprobar que es posible analizar el tráfico de la red doméstica sin demasiado esfuerzo, pero hay ciertas amenazas que como los paquetes cifrados en las conexiones https que es difícil de detectar las amenaza.

También se ha podido comprobar el gran número de peticiones que realizan a diario a los servidores centrales de Google, Amazon, etc. de los aparatos dispositivos IoT, televisores inteligentes, etc.

- **Investigar, seleccionar herramientas y tecnologías adecuadas para implementarlo de bajo coste.**

Se ha comprobado que con una Raspberry PI de bajo coste es posible implementar una solución de IDS con SIEM en el hogar, pero siempre siendo una máquina más actualizada como Raspberry PI 4.

Viendo también la falta de stock de este modelo, sería recomendable buscar alternativas en el mercado a la Raspberry Pi.

- **Configurar y poner en marcha el sistema de detección de intrusos para mejorar la ciberseguridad de la red doméstica.**

La investigación y la implementación de Suricata ha permitido comprobar que no es realmente complicado implementar un IDS en la red del hogar. Que crear reglas para mejorar la seguridad del hogar es eficiente, sobre todo para ser conscientes del tráfico de la red.

- **Implementar y configurar las herramientas básicas para el monitoreo y análisis del tráfico de la red doméstica de una forma ágil.**

Con este proyecto se ha podido comprobar como instalar y configurar la herramienta de Elastic como herramienta de monitoreo y análisis, de forma sencilla y accesible para todos. Siendo una herramienta muy importante en soluciones SIEM.

- **Desarrollar un panel de conexiones y alertas en la suite ELK que permita visualizar y monitorear el estado de la red en tiempo real de forma sencilla.**

Con Kibana es posible crear paneles sencillos y la opción de crear alertas también ha sido sencillo, se han adquirido los conocimientos necesarios y su objetivo ha sido cumplido.

Objetivos secundarios:

- **Investigar que mejoras se podrían añadir al sistema IDS aplicando soluciones de Machine Learning o IA que permita ser más reactivo durante la detección de intrusiones.**

Es un área que hubiera sido interesante investigar, pero por falta de tiempo en la implementación no se ha podido conseguir el objetivo, pero aún queda como objetivo futuro.

- **Configurar el monitoreo y análisis del tráfico de la red doméstica para aplicaciones específicas con el objetivo de controlar el uso de la red en una red doméstica.**

Por medio de las reglas de suricata se ha podido demostrar y controlar ciertas aplicaciones en la red doméstica, como es el caso de Whatsapp.

- **Realizar pruebas de funcionamiento para comprobar su eficacia como sistema de detención de intrusos.**

Se han creado varias pruebas de inicio a fin con suricata y elastic comprobando la eficacia de juntar un IDS con un SIEM.

- **Documentar la implementación del IDS y su configuración para que sea fácil de seguir y pueda ser replicado fácilmente por un usuario medio.**

Se ha documentado la implementación lo más sencillo posible, para que cualquier persona pueda replicarlo en el hogar.

- **Revisar soluciones complementarias al IDS propuesto, como podría ser la solución ZEEK.**

Por falta de tiempo en la fase de implementación este objetivo no ha sido cumplido.

4.2 Problemas encontrados durante el proyecto.

A continuación, se listan los problemas que se ha encontrado durante el transcurso de la implementación y desarrollo del trabajo.

Elección de hardware

La premisa del trabajo era demostrar la eficiencia de implementar la solución con una raspberry-pi de bajo coste, a la hora de implementarlo en el mercado no había stock para realizar un pedido e implementar la solución con una raspberry 4 y se tuvo que reutilizar un modelo de raspberry pi que no estaba preparado.

Distribución Linux de 64 bits

Se ha podido comprobar que la raspberry con la distribución de 64 bits ha ido más lenta que con la distribución recomendada de 32 bits. Pero los agentes que hay que instalar en el IDS de elastic, solo da soporte para distribuciones de 64 bits con lo que se ha tenido un sistema menos optimizado.

Listas compartidas de Suricata

Con la versión de 64 bits no se ha podido utilizar las fuentes con las reglas de Emerging Threats porque al activarlas, el IDS se bloqueaba por falta de recursos.

Zeek

Uno de los objetivos que se querían conseguir era estudiar y validar el uso la herramienta zeek con suricata como herramientas complementarias, pero por falta de tiempo no se ha podido implementar.

5. Trabajos futuribles

Una vez finalizado el trabajo y su investigación se han identificado algunas áreas que podrían beneficiarse de trabajos futuros y mejoras adicionales.

Mejoras en el rendimiento:

Aunque la solución de integración entre Suricata y el SIEM de Elastic ha demostrado ser efectiva, se reconoce la necesidad de utilizar hardware más potente para garantizar un rendimiento óptimo. El siguiente paso que se realizará será mover el IDS y el SIEM en una máquina con más recursos.

También se podría explorar la optimización de recursos y limitar los recursos al mínimo necesario.

Ampliación de reglas y firmas:

A medida que evolucionan las amenazas, es importante mantener actualizadas las reglas y firmas utilizadas por Suricata para la detección de ataques, que ya lo realiza. En trabajos futuros, se podría realizar un análisis periódico de las nuevas técnicas y patrones de ataque, y colaborar con los repositorios públicos de reglas, ofreciendo propuestas con nuevas reglas y firmas.

Integración con otras herramientas de seguridad:

La solución propuesta se centra en la integración entre Suricata y el SIEM de Elastic. Sin embargo, se podría explorar oportunidades de integración con otras herramientas de seguridad, como firewalls, sistemas de prevención de intrusiones (IPS) u otras soluciones de detección de amenazas, para fortalecer aún más la seguridad de la red doméstica. Una de las herramientas interesantes de integrar sería la herramienta PFSense, es como un todo-en-uno de firewall, ids, vpn y proxy que ayudaría a fortalecer y centralizar la gestión de la seguridad de la red.

En conclusión, existen varias áreas de trabajos futuros para mejorar y expandir la solución propuesta. Desde mejoras en el rendimiento hasta la ampliación de reglas y firmas o integración con otras herramientas de seguridad son los trabajos futuribles que podrían fortalecer aún más la seguridad de la red.

6. Bibliografía

Raspberry Pi. (2023) Home https://www.raspberrypi.com/

Kubii. (2023). Raspberry Pi 4 Modelo B - 2GB. https://www.kubii.es/raspberry-pi-3-2-b/2771-nuevo-raspberry-pi-4-modelo-b-2gb-3272496308794.html?src=raspberrypi

Suricata. (2023). Suricata User Guide. https://suricata.readthedocs.io/en/latest/

Suricata. (2023). Rules Format. https://docs.suricata.io/en/suricata-6.0.10/rules/intro.html

Suricata. (2023). Binary packages. https://suricata.readthedocs.io/en/suricata-6.0.0/install.html#install-binary-packages

Snort. (2023). Documents. https://www.snort.org/documents

AT&T Cibersecurity. (22 Maig del 2020). Open Source IDS Tools: Comparing Suricata, Snort, Bro (Zeek), Linux. https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

Elastic. (2023). Beats and Elastic Agent capabilities - Capabilities comparison. https://www.elastic.co/guide/en/fleet/current/beats-agent-comparison.html#additional-capabilities-beats-and-agent

Elastic. (2023). Elastic Docs. https://www.elastic.co/guide/index.html

Elastic. (2023). Hardware prerequisites. https://www.elastic.co/guide/en/cloud-enterprise/current/ece-hardware-prereq.html

Elastic. (2023). Beats: Lightweight Data Shippers for Elasticsearch & Logstash. https://www.elastic.co/beats/

Elastic. (2023). Logstash Reference. https://www.elastic.co/guide/en/logstash/current/index.html

Elastic. (2023). Elasticsearch Reference. https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html

Elastic. (2023) Kibana User Guide. https://www.elastic.co/guide/en/kibana/current/index.html

7. Vocabulario

Logstash: herramienta de procesamiento de datos para la ingestión, transformación y envío de logs y otros datos.

Elasticsearch: motor de búsqueda y análisis de datos distribuido que permite el almacenamiento y búsqueda de datos en tiempo real.

Kibana: herramienta de visualización de datos que permite la exploración, análisis y presentación de datos almacenados en Elasticsearch.

Beats: agentes ligeros que se utilizan para enviar datos a Logstash o Elasticsearch, permitiendo la recopilación y envío de datos en tiempo real.

ELK: ELK es un acrónimo que hace referencia a la combinación de tres herramientas de código abierto: Elasticsearch, Logstash y Kibana

Docker: Docker es una plataforma de contenedores que permite empaquetar, distribuir y ejecutar aplicaciones de manera eficiente y portátil.

Docker Compose: Herramienta para definir y administrar aplicaciones multi-contenedor utilizando archivos YAML.

Raspberry Pi: Raspberry Pi es una serie de computadoras de placa única (SBC) de bajo costo y tamaño reducido.

RESTful: abreviatura de Representational State Transfer, es un estilo arquitectónico utilizado para diseñar servicios web que sean escalables, interoperables y fáciles de mantener.

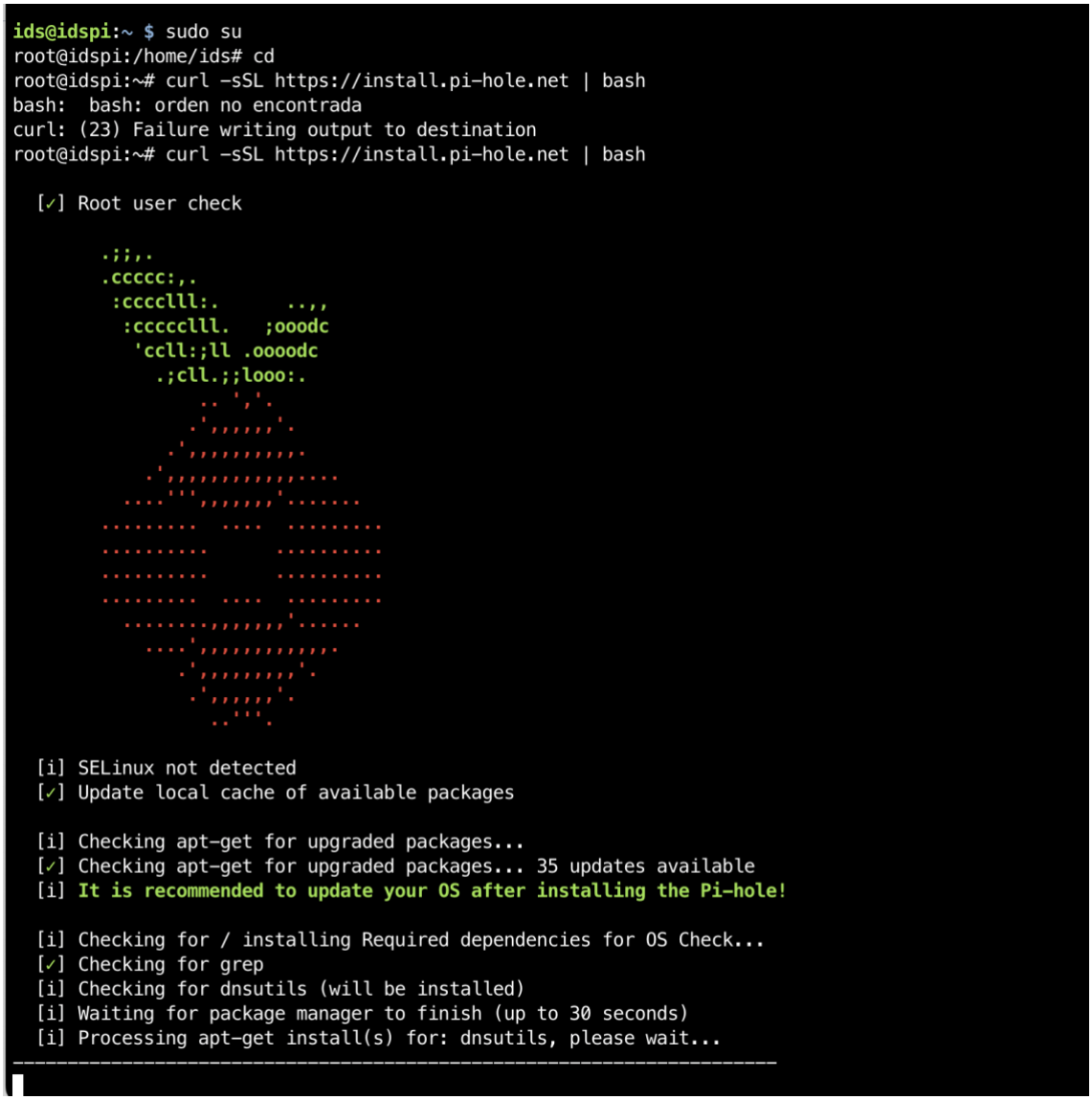
Fleet Server: Forma parte de la solución Elastic Endpoint Security, que permite la gestión y seguridad centralizada de los agentes en diferentes dispositivos.

Anexo 1

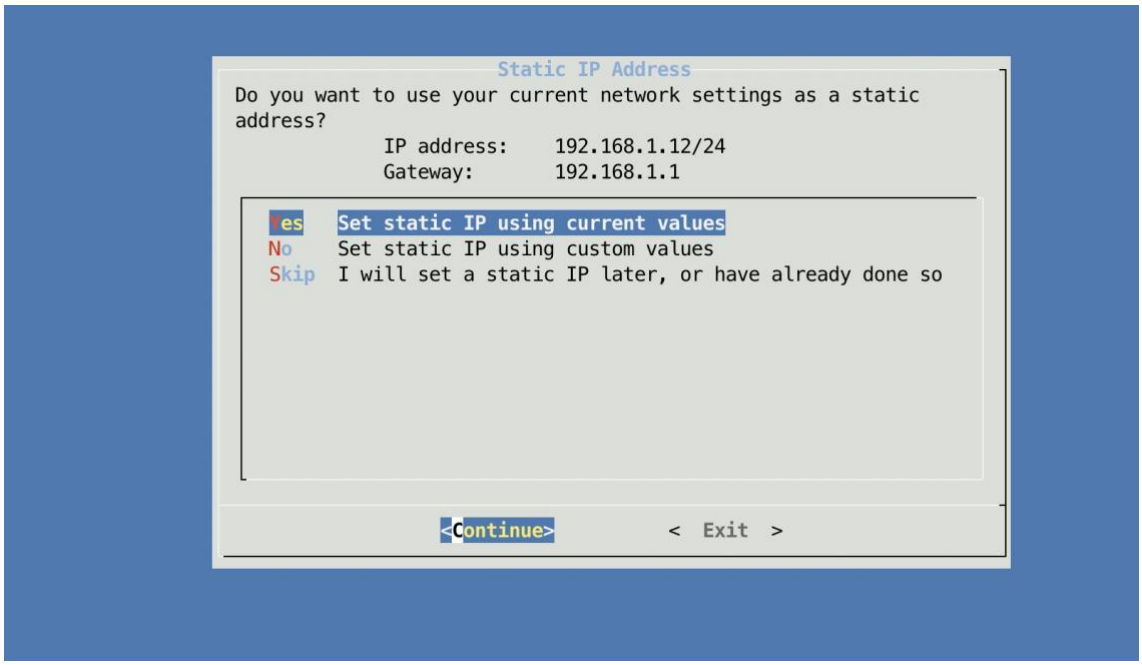
A continuación, se procede a mostrar una guía para instalar Pihole en una raspberry pi

Paso 1. Descargar y lanzar el script de instalación

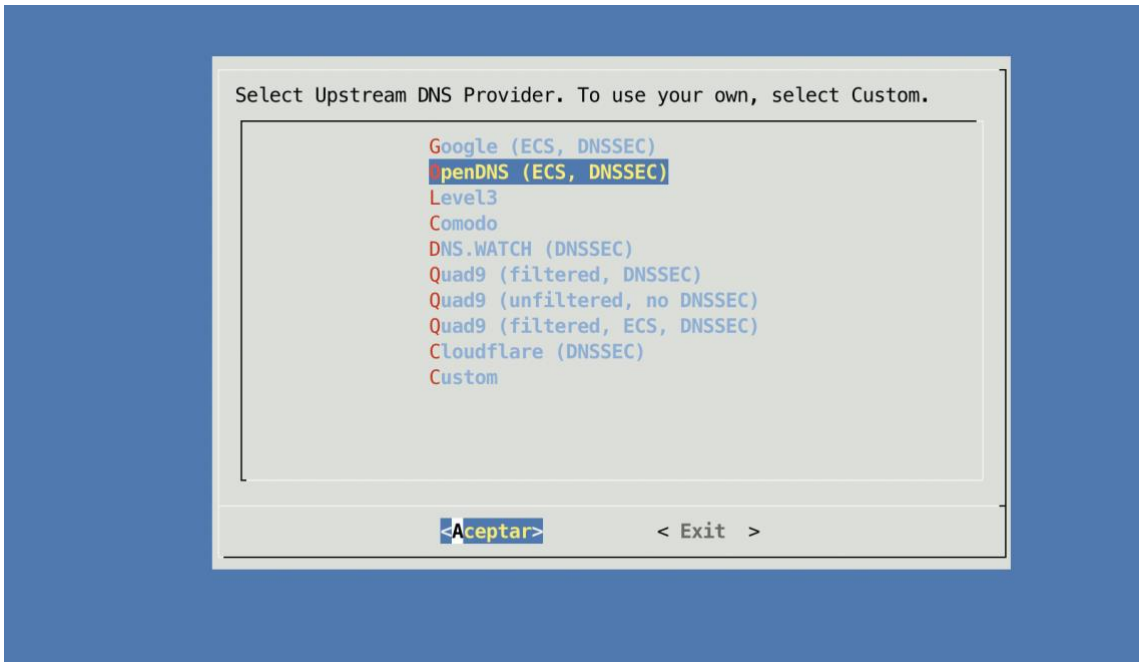
```
curl -sSL https://install.pi-hole.net | bash
```



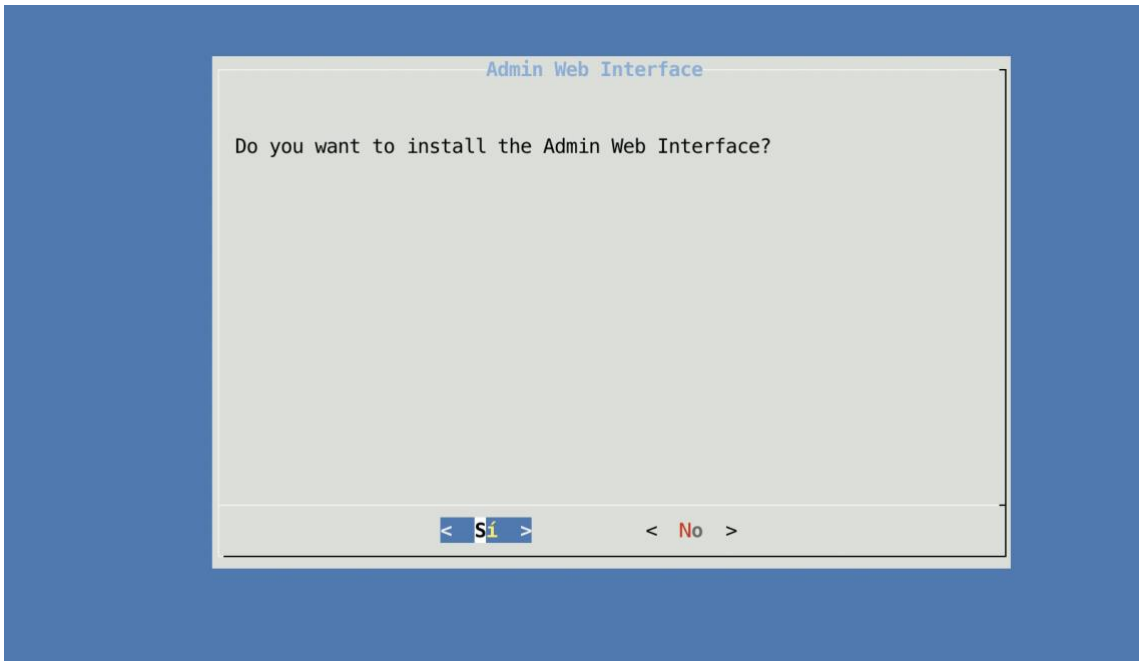
Paso 2. Seleccionar la IP estática de la red

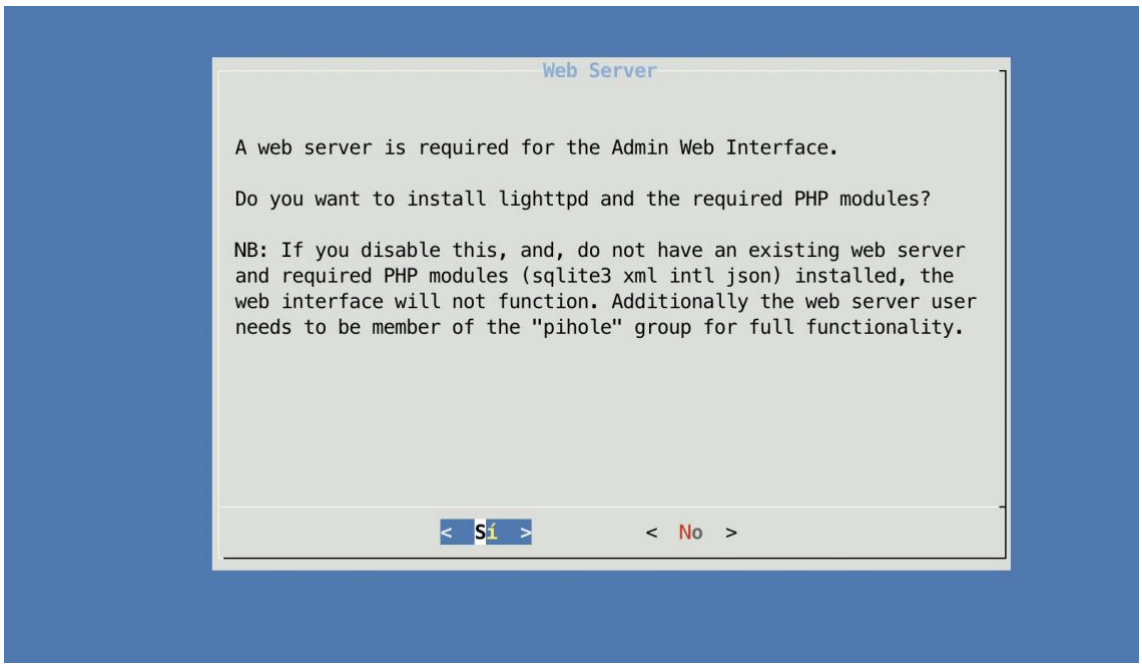


Paso 3. Seleccionar el proveedor de DNS

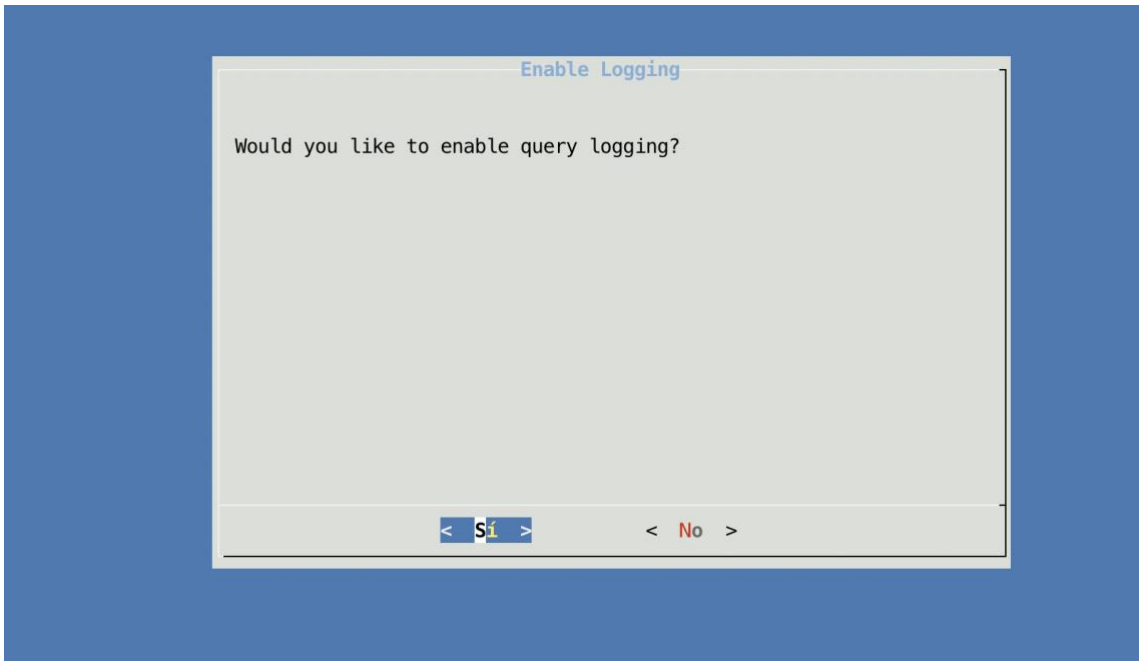


Paso 4. Activar la UI de administración

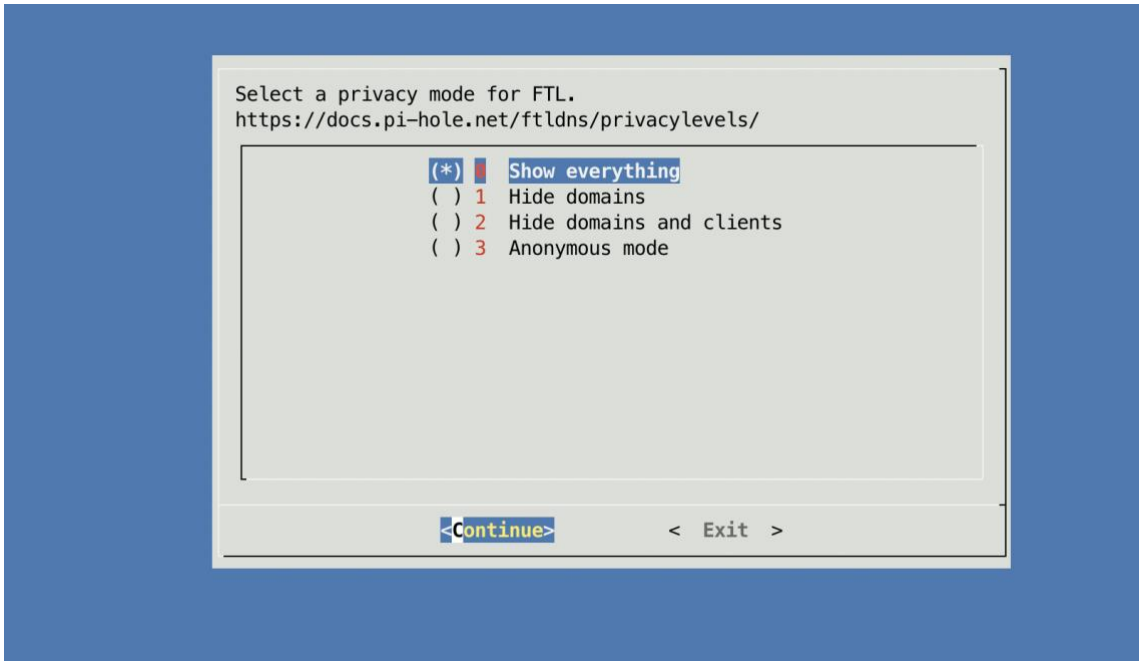




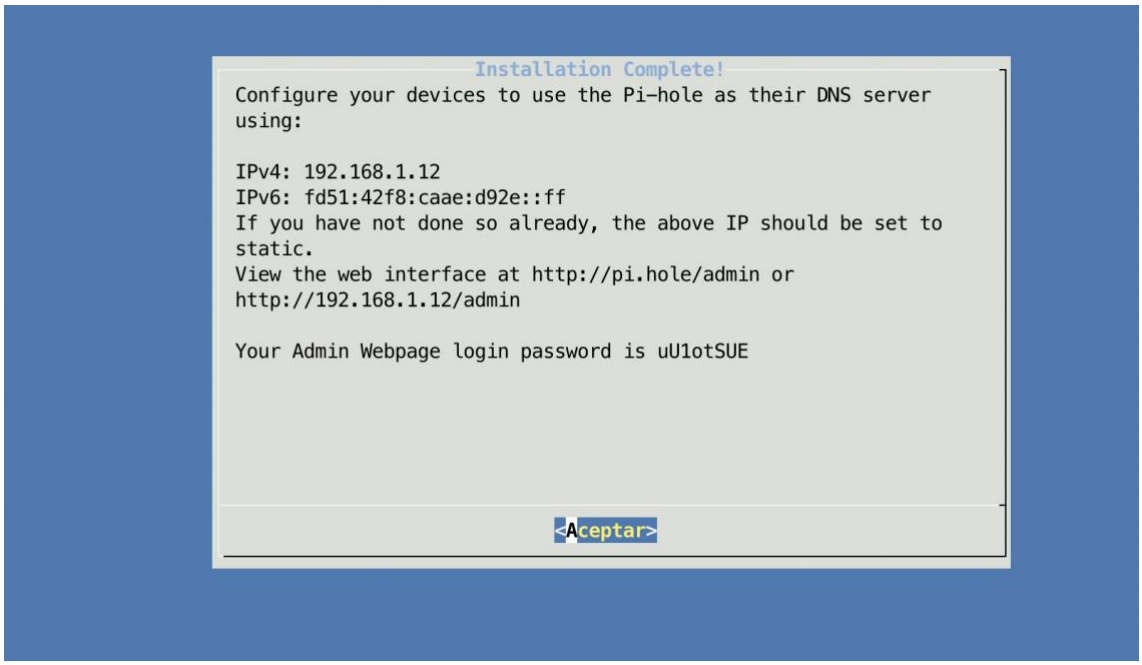
Paso 4. Activar el registro de log



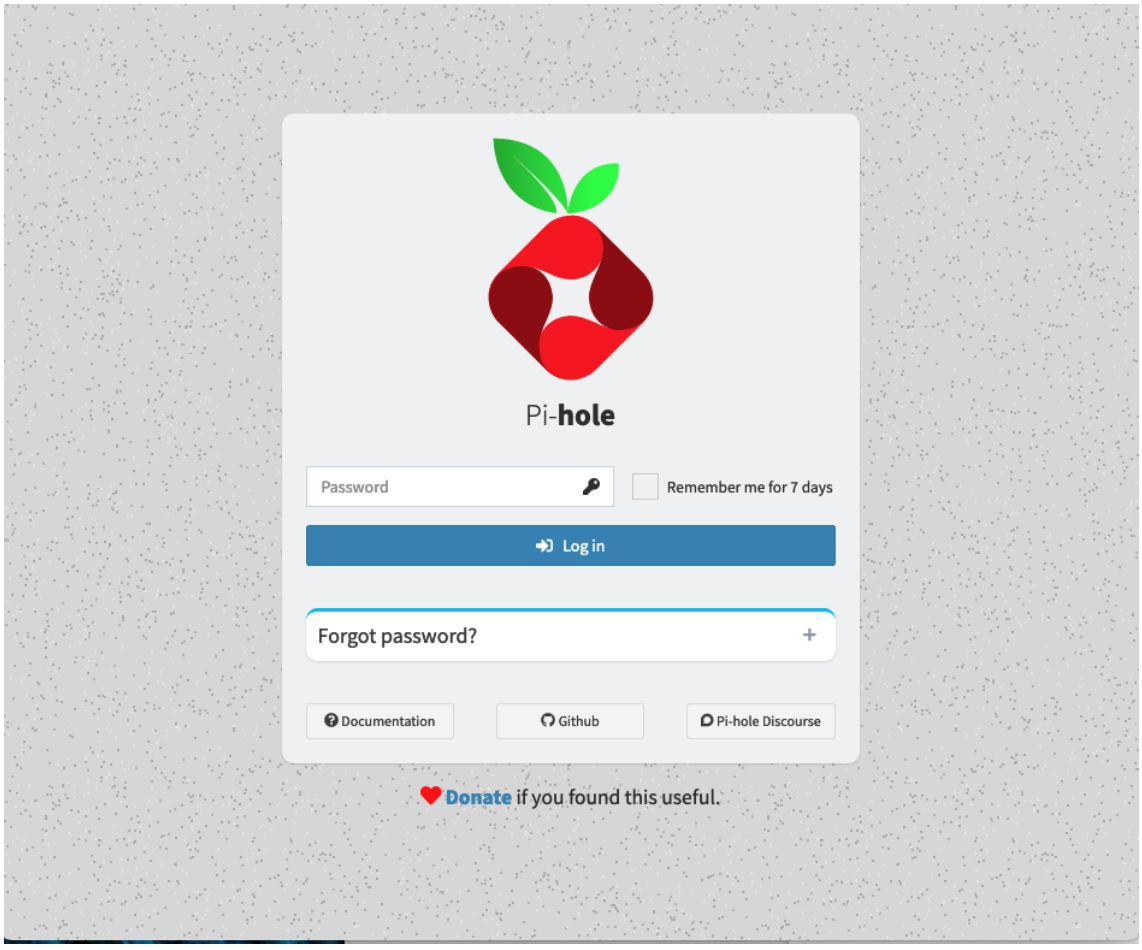
Paso 5. Seleccionar el nivel de privacidad de las estadísticas, por defecto no se quiere ocultar ninguna información.

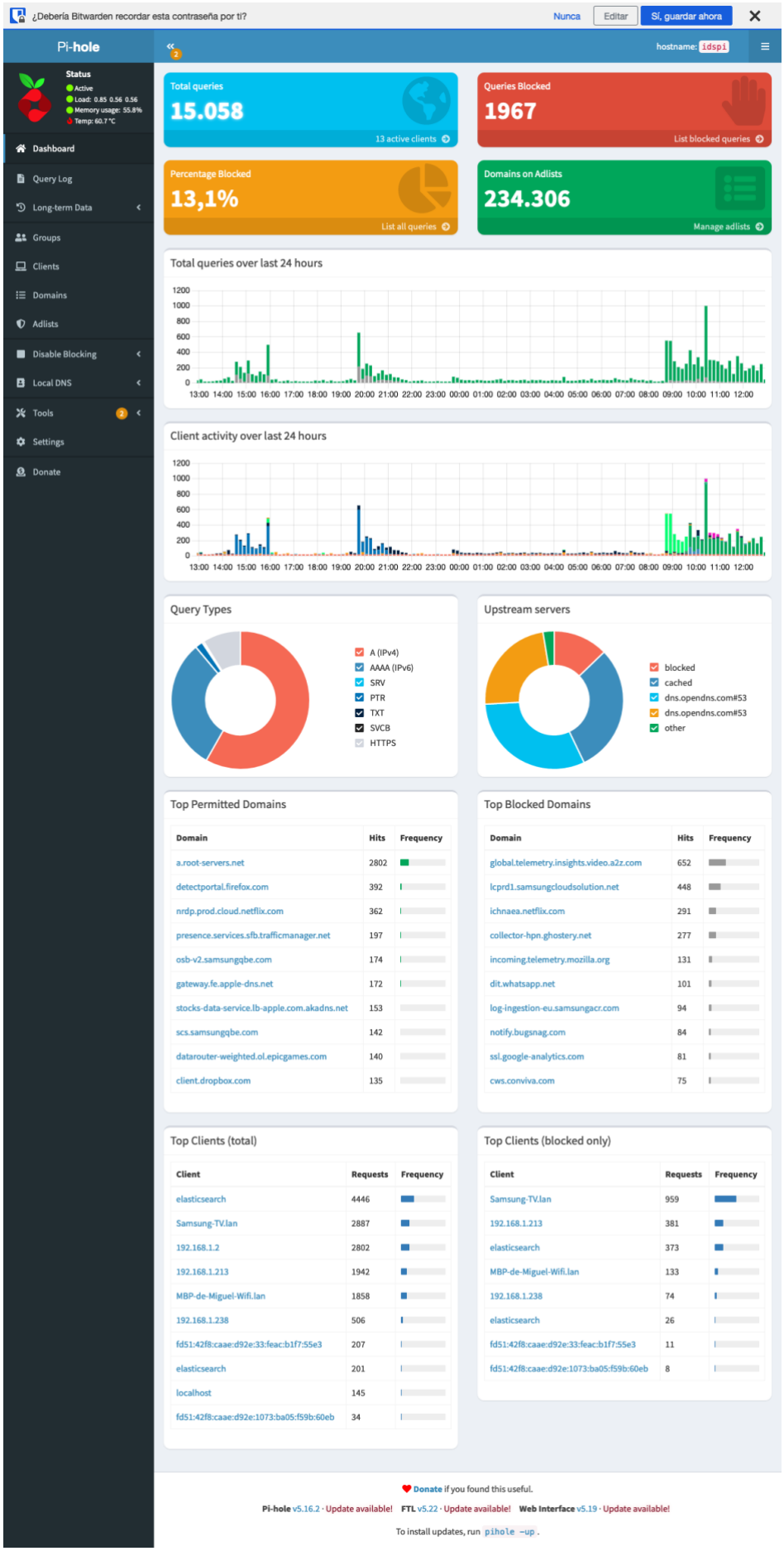


Paso 7. Confirmar



Se accede a PiHole http://pi.hole/admin/login.php




¿Debería Bitwarden recordar esta contraseña por ti?
Nunca
Editar
Sí, guardar ahora
Pi-hole
hostname: idspi
Status
Active
Load: 0.85 0.56 0.56
Memory usage: 55.8%
Temp: 60.7 °C
Dashboard
Query Log
Long-term Data
Groups
Clients
Domains
Adlists
Disable Blocking
Local DNS
Tools
Settings
Donate
Total queries
15.058
13 active clients
Queries Blocked
1967
List blocked queries
Percentage Blocked
13,1%
List all queries
Domains on Adlists
234.306
Manage adlists
Total queries over last 24 hours
Client activity over last 24 hours
Query Types
A (IPv4)
AAAA (IPv6)
SRV
PTR
TXT
SVCB
HTTPS
Upstream servers
blocked
cached
dns.opendns.com#53
dns.opendns.com#53
other
Top Permitted Domains
Domain
Hits
Frequency
a.root-servers.net 2802
detectportal.firefox.com 392
nrdp.prod.cloud.netflix.com 362
presence.services.sfb.trafficmanager.net 197
osb-v2.samsungqbe.com 174
gateway.fe.apple-dns.net 172
stocks-data-service.lb-apple.com.akadns.net 153
scs.samsungqbe.com 142
datarouter-weighted.ol.epicgames.com 140
client.dropbox.com 135
Top Blocked Domains
Domain
Hits
Frequency
global.telemetry.insights.video.a2z.com 652
lcprd1.samsungcloudsolution.net 448
ichnaea.netflix.com 291
collector-hpn.ghostery.net 277
incoming.telemetry.mozilla.org 131
dit.whatsapp.net 101
log-ingestion-eu.samsungacr.com 94
notify.bugsnag.com 84
ssl.google-analytics.com 81
cws.conviva.com 75
Top Clients (total)
Client
Requests
Frequency
elasticsearch 4446
Samsung-TV.lan 2887
192.168.1.2 2802
192.168.1.213 1942
MBP-de-Miguel-Wifi.lan 1858
192.168.1.238 506
fd51:42f8:caae:d92e:33:feac:b1f7:55e3 207
elasticsearch 201
localhost 145
fd51:42f8:caae:d92e:1073:ba05:f59b:60eb 34
Top Clients (blocked only)
Client
Requests
Frequency
Samsung-TV.lan 959
192.168.1.213 381
elasticsearch 373
MBP-de-Miguel-Wifi.lan 133
192.168.1.238 74
elasticsearch 26
fd51:42f8:caae:d92e:33:feac:b1f7:55e3 11
fd51:42f8:caae:d92e:1073:ba05:f59b:60eb 8
Donate if you found this useful.
Pi-hole v5.16.2 · Update available! FTL v5.22 · Update available! Web Interface v5.19 · Update available!
To install updates, run pihole -up .

Para añadir más dominios y bloquear sospechosos, de publicidad, telemetría o tracking se procede añadir más lista en el apartado “Adlists”

Se añade la siguiente lista para bloquear más dominios de publicidad
https://v.firebog.net/hosts/AdguardDNS.txt

Es posible encontrar muchas más en estas dos direcciones

- https://firebog.net/
- https://avoidthehack.com/best-pihole-blocklists