Universitat Oberta de Catalunya uoc.edu

“Ventajas e implementación de un sistema IDS/SIEM en el ámbito familiar”

Alumno: José Antonio Salom Martín
Plan de Estudios: “Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones”. (MISTIC)
Trabajo de fin de master.Seguridad empresarial

Director del TFM: Jorge Chinea López
Profesor/a responsable de la asignatura:Victor Garcia Font

Fecha de entrega: 4 de junio de 2019

Agradecimientos:
Quiero dar las gracias a dos personas muy especiales en mi vida.
- A Cristina por el apoyo, la paciencia y el tiempo dedicado a la família mientras realizaba este trabajo.
- A mi padre Jose Salom Ferrer por enseñarme lo que es la fuerza de voluntad para seguir siempre adelante.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Ventajas e implementación de un sistema IDS/ SIEM en el ámbito familiar*
Nombre del autor:	*JOSÉ ANTONIO SALOM MARTÍN*
Nombre del consultor/a:	*Jorge Chinea López*
Nombre del PRA:	*Victor Garcia Font*
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*Ventajas e Implementación de un sistema SIEM para la detección de amenazas*
Idioma del trabajo:	***Castellano***
Palabras clave	*IDS, SIEM, Familia*

Resumen del Trabajo:

La finalidad de este trabajo es valorar las ventajas de implementar un IDS/SIEM en el ámbito familiar y de bajo coste. Para ello se ha implementado en una familia un sistema de detección de intrusos mediante Suricata instalado en una Raspberry-pi. Para poder limitar el tráfico se ha integrado en la Raspberry-pi Pi-hole utilizando servicios DNS y DHCP.Por otro lado se ha implantado un sistema en Windows10 para recuperar los datos del IDS y mostrarlos de forma visual utilizando herramientas de la pila Elastic como son Logstash, ElasticSearch y Kibana. Se ha realizado un estudio de las amenazas en la red a las que están expuestos los menores con la finalidad de diseñar salvaguardas a ésta.

Finalmente, las conclusiones han sido satisfactorias en lo referente a tener un control centralizado del uso de internet en el ámbito familiar. Si bien no es posible aplicar salvaguardas a todas las amenazas detectadas, se considera que mejora sustancialmente la monitorización y la ciberseguridad del hogar.

Abstract:

The purpose of this essay is to assess the advantages of implementing a low cost IDS / SIEM in a family. For this purpose, an intrusion detection system (IDS) using Suricata, installed in a Raspberry-pi, has been implemented in a family. In order to limit the traffic it has been integrated into the Pi-hole Raspberry-pi using DNS and DHCP services. On the other hand, a system has been implemented in Windows10 to recover the IDS data and to display them in a visual way using Elastic stack tools such as Logstash, ElasticSearch and Kibana. A study has been made of the threats in the network to which minors are exposed in order to design safeguards to this.

Finally, it is concluded that the results have been satisfactory in terms of having a centralized control of Internet used in the family, and although it is not possible to apply safeguards to all detected threats, it is considered a substantial improvement in household cybersecurity.

Índice

Índice de ilustraciones

1. Introducción

1.1 Contexto y justificación del Trabajo

Las estadísticas lo dejan claro, los ciberataques crecen cada año y por eso muchas organizaciones están implantando costosos sistemas se seguridad para proteger sus activos.

Entre los sistemas que se están instalando son los sistemas de detección de intrusos (IDS) y sistemas de gestión de eventos e información de seguridad (SIEM). Éstos permite monitorizar el tráfico que circula por la red y avisan en caso de detectar algún evento sospechoso.

Tomando como ejemplo un símil en la seguridad física, un IDS/SIEM lo podríamos comparar con los sistemas de video vigilancia que hace años se instalan en las organizaciones, con la finalidad de detectar cualquier intruso sospechoso de realizar algún delito o sabotaje. En el caso de los sistemas IDS/SIEM los delincuentes a detectar actúan a través de una red de ordenadores. Es por ello que los IDS se dedican a capturar todo el tráfico que circula por la red buscando algún evento sospechoso. Por otro lado, dada la gran cantidad de información a analizar, los SIEM se encargan de mostrar la información de forma más visual.

Pero además del aumento de los ciberataques, los objetivos de los cibercriminales han ido ampliándose. Hoy en día pueden ser objeto de un ciberataque tanto la información de una gran empresa, como los equipos conectado a internet de una familia o los miembros que la componen.

Este trabajo de fin de master se centra en el ámbito familiar, donde no existen sistemas de seguridad preventiva como los IDS/SIEM. Los sistemas de seguridad que encontramos instalados en este contexto no suele pasar de algún antivirus en algún ordenador.

Teniendo en cuenta la cantidad de dispositivos que tenemos conectados a internet en nuestras casas y que la instalación de un antivirus no da una protección total a las ciberamenazas de hoy en día, se pretende valorar si es útil y viable disponer de un IDS/ SIEM en el ámbito domestico.

Otro objetivo de este proyecto es valorar si estos sistemas pueden ayudar los padres a tener mejor información sobre uso que se hace de internet por parte de los miembros de la familia.

Estos sistemas nos podrían informar si se está accediendo a información no recomendable para la edad de nuestros hijos, o cuanto tiempo están con los dispositivos conectados. Esto puede ser muy interesante para conocer el grado de tecnoadicción que

existe en la familia, siendo esta una de las mayores preocupaciones que tienen los padres de hoy en día.

Por otro lado, no debemos olvidar que los cibercriminales pueden tener como objetivo algún miembro de la familia con la finalidad de realizar algún tipo de extorsión, o ser víctimas de grooming, sexting, o ataques de phishing.

El último reto del proyecto es que sea de bajo coste y por lo tanto asequible al bolsillo familiar. Es por ello que se estudiará la posibilidad de instalación de un IDS en una Raspberry-pi con un presupuesto no supere los 100€.

1.2 Objetivos del Trabajo

Los principales objetivos de este trabajo de fin de master son los siguientes:

Objetivos de investigación:

- Investigación sobre como podríamos usar el IDS/SIEM para funciones de control parental.
- Investigación sobre las posibilidades que ofrecen los sistemas de detección de intrusos para dispositivos de bajo coste.
- Investigación y estudio de Suricata IDS.
- Investigación y estudio de sistemas SIEM que puedan integrarse con el IDS en contexto de este trabajo.

Objetivos de implantación:

- Instalación y configuración de dispositivos necesarios y una correcta topología de red.
- Aprender a instalar y configurar Suricata IDS.
- Aprender a configurar reglas y realizar pruebas de detección.
- Aprender instalar y configurar un sistema de monitorización para Suricata IDS, de forma que su uso sea sencillo y visual.
- Puesta en marcha del producto final y realización de pruebas prácticas para control parental.

Objetivos de Entrega:

- Desarrollar las entregas parciales y enviarlas en tiempo y forma.
- Desarrollar la memoria final del trabajo.
- Preparar un video presentación.

Como se puede observar, el trabajo de tiene una componente de búsqueda de información e investigación sobre el funcionamiento de herramientas IDS, SIEM y control parental, pero además pretende implementar un producto funcional que cumpla con su objetivo.

Así pues, el sistema implantado a su vez debería cumplir los siguientes objetivos:

- **Control parental:** Mediante el análisis del tráfico de red podrá avisarnos si algún dispositivo está conectándose a páginas de contenido inapropiado o controlar el tiempo que los dispositivos están conectados a internet.
- **Bajo coste:** Utilizar dispositivos de bajo coste y software libre para evitar costes elevados en la implementación de la solución.
- **Información visual**: Como está orientado a la familia, la información debe ser visual y sencilla de interpretar.
- **Mejorar la ciberseguridad de la familia:** Mediante el sistema de detección de intrusos se pretende mejorar de la seguridad de la red local de la familia vigilando el tráfico de red y detectando cualquier anomalía.

1.3 Enfoque y método seguido

El enfoque de este proyecto, al contextualizarlo en al ámbito familiar, consistirá en investigar si es viable y útil el uso de tecnología IDS/SIEM, cuya finalidad suele ser detectar tráfico peligroso en una red, añadiéndole funciones de control parental.

El proyecto tiene dos partes diferenciadas.

- Parte teórica de investigación sobre las herramientas a utilizar y posibilidad de utilizarlas para el control parental, cuyos resultados se presentarán en la siguiente entrega (PEC 2)
- Parte práctica de instalación y configuración de todos los componentes del sistema que se pretende finalizar para la penúltima entrega (PEC 3) antes de la elaboración de la memoria.

Dado que no hay mucho tiempo para el desarrollo de todos los objetivos, en la planificación de tareas presentada, se puede observar como coexisten tareas del bloque teórico y del bloque práctico para ir avanzando en la implementación en la medida de lo posible.

En el hito de la presentación de la PEC2, una vez realizada la investigación y posibilidades de los diferentes componentes, se valorará si la consecución de todos los objetivos son reales, ajustando la planificación de forma que, si bien algún objetivo se considera inviable ponerlo en práctica, siempre nos queden los objetivos de aprendizaje.

Por otro lado se ha tenido en cuenta las vacaciones de semana santa y el 1 de mayo, ya que son fechas que se suele salir con la familia y en las que no se puede avanzar al mismo ritmo.

1.4 Planificación del Trabajo

Cod	Actividad	Inicio	Fin	Duración
1	**Planificación**			
1.1	Establecer problema a resolver y contexto	20/02/19	22/02/19	3
1.2	Definición de objetivos	20/02/19	22/02/19	3
1.3	Definir propuesta metodología	23/02/19	26/02/19	4
1.4	**Elaborar cronograma de trabajo**			
1.4.1	Definición de tareas	27/02/19	04/03/19	6
1.4.2	Calcular tiempos para entregas	01/03/19	05/03/19	5
1.4.3	Identificar riesgos	04/03/19	05/03/19	2
1.5	Definir costes y recursos necesarios	23/02/19	26/02/19	4
1.6	**Entrega del plan de trabajo**	**05/03/19**	**05/03/19**	**Hito**
2	**Investigación**			
2.1	**Investigación sobre IDS**			
2.1.1	Estudio de Suricata-IDS	06/03/19	10/03/19	5
2.1.2	Estudio de dispositivos necesarios	11/03/19	13/03/19	3
2.1.3	Investigar Raspberry-pi y configuración necesaria	14/03/19	17/03/19	4
2.2	**Investigación sobre sistemas SIEM**			
2.2.1	Estudio de diferentes sistemas SIEM	18/03/19	20/03/19	3
2.2.2	Selección de sistema SIEM	21/03/19	24/03/19	4
2.3	**Investigación sobre de control parental**	**25/03/19**	**31/03/19**	**7**
2.3.1	Estudio de necesidades de control parental			
2.3.2	Estudio de posibilidades de un IDS par control parental			
2.4	**Entrega del PEC2**	**02/04/19**	**02/04/19**	**Hito**
2.4.1	Recopilar información sobre la investigación	13/03/19	27/03/19	14
2.4.2	Redactar de la PEC2	27/03/19	02/04/19	6,5
3	**Implantación**			
3.1	**Adquisición de recursos**	**14/03/19**	**15/03/19**	**2**
3.2	**Implantación y configuración de IDS en Raspberry-pi**			
3.2.1	Instalación y configuración de Raspberry-pi	18/03/19	24/03/19	7
3.2.2	Instalación de dispositivos de red	25/03/19	28/03/19	4
3.2.3	Instalación configuración de Suricata-IDS	29/03/19	03/04/19	5,5
3.2.4	Configuración de reglas de Suricata-IDS	03/04/19	07/04/19	4,5
3.2.5	Pruebas de detección del IDS	06/04/19	07/04/19	2
3.3	**Implantación y configuración de SIEM**			
3.3.1	Instalación y configuración de SIEM seleccionado	08/04/19	13/04/19	6
3.3.2	Tareas de Integración de IDS con el SIEM	14/04/19	18/04/19	5
3.3.3	Pruebas de monitorización	17/04/19	19/04/19	3
3.4	**Implantación de controles parentales**			
3.4.1	Control de trafico Web	21/04/19	22/04/19	2
3.4.2	Control de dispositivos conectados	23/04/19	25/04/19	3
3.4.3	Control de tiempo conectado	26/04/19	28/04/19	3
3.5	**Entrega de la PEC3**	**30/04/19**	**30/04/19**	**Hito**
3.5.1	Recopilar información sobre la implantación	06/04/19	26/04/19	21
3.5.2	Redacción del PEC3	27/04/19	30/04/19	4
4	**Presentación**			
4.1	**Presentación y defensa del TFM**	**04/06/19**	**04/06/19**	**Hito**
4.1.1	Conclusiones sobre el trabajo realizado	02/05/19	05/05/19	4
4.1.2	Elaboración de la memoria final	06/05/19	26/05/19	21
4.1.3	Elaboración del video presentación de la memoria	27/05/19	03/06/19	8

Tabla 1: Planificación

1.5. La planificación temporal detallada de estas tareas y sus dependencias.

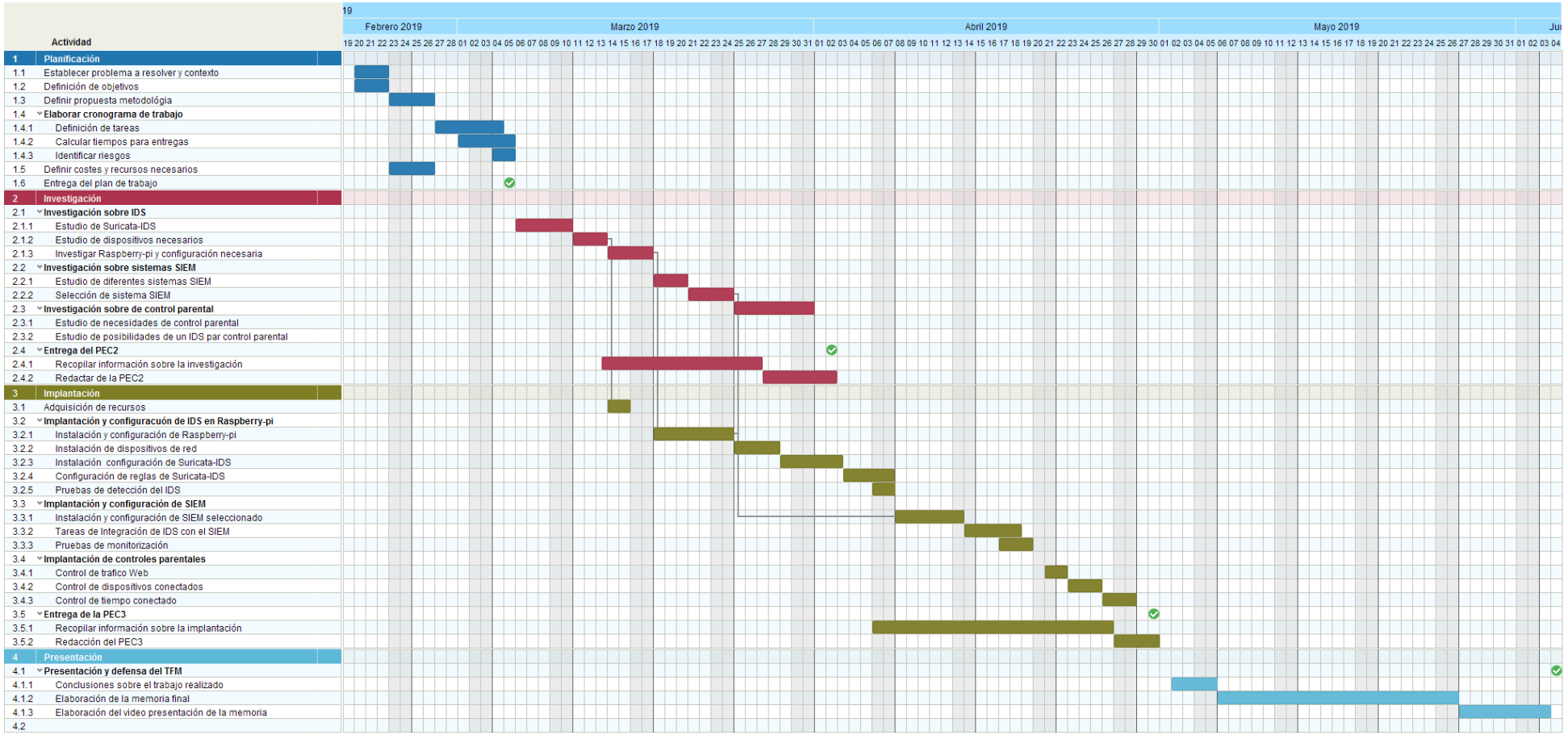


Tabla 2: Diagrama gantt

1.6. Análisis de riesgos

A continuación se enumeran una serie de riesgos que pueden hacer proyecto fracase o que que la planificación no pueda ajustarse a los tiempos marcado.

R1. Objetivo demasiado ambicioso.

Definición del riesgo:

Existe el riesgo de no acabar el trabajo en el tiempo planificado por que el objetivo del proyecto es bastante ambicioso para el periodo de elaboración del trabajo de fin de master, tocando temas que podrían utilizarse para la elaboración de un TFM por cada uno de ellos, como por ejemplo:

- Investigación, configuración y pruebas de un sistema de detección de intrusos.
- Investigación, configuración y pruebas de un sistemas de motorización de información y eventos de seguridad.
- Estudio e implementación de medidas de control parental para un uso seguro y responsable de internet.
- Instalación y puesta en marcha una solución de seguridad.

Mitigación del riesgo:

Se tiene que ser cauteloso en no perderse en cada tema a investigar y acotar el tiempo dedicado.

En la investigación no invertir excesivo tiempo en valorar en profundidad la herramientas disponibles y centrarse en las que mejor se adaptan al proyecto por su compatibilidad y apoyo de la comunidad en la red.

Si hace falta, se tendrá que reducir el nivel de cada objetivo o incluso abandonar la puesta en marcha de alguno y poner foco el aprendizaje y la finalización del trabajo.

R2. Problemas o incompatibilidades de integración de las diferentes herramientas.

Definición del riesgo:

Hasta que no se realice una investigación más detallada, no están definidas las herramientas a utilizar. Esto puede suponer encontrarnos con incompatibilidades entre las diferentes herramientas seleccionada que generen una pérdida de tiempo que acabe penalizando la planificación realizada.

Mitigación del riesgo:

Asegurarse en la fase investigación para la selección de herramientas sobre la compatibilidad con el resto del sistema, antes de entrar en un estudio en detalle de configuración de una determinada herramienta.

R3. Problemas a la hora de definir las medidas de control parental

Definición del riesgo:

Un objetivo principal es poder usar el sistema para control parental que mediante el uso de un IDS/SIEM se deberia poder realizar:

- La monitorización de acceso a contenido inadecuado por parte de los dispositivos conectados a internet.
- Control del tiempo que cada dispositivo está conectado a internet con la finalidad de dar herramientas a los padres para prevenir la tecno-adicción en el uso de internet.

Si al final se concluye que el uso de IDS/SIEM no es la tecnología más adecuada para cumplir alguno de estos objetivos, se puede haber invertido mucho tiempo en pruebas que afecten a la planificación del proyecto.

Mitigación del riesgo:

Una vez se haya investigado y valorado las posibilidades de control parental, en el caso de que se detecte una elevada complejidad en la implementación con una desviación en la planificación del proyecto, se deberá decidir no implementarlo.

R4. Problemas en la implementación física del producto.

Definición del riesgo:

Una vez seleccionadas las herramientas y realizado el estudio sobre su uso y configuración, pueden surgir con problemas inesperados en el momento de la implementación práctica sobre los dispositivos que causen una gran inversión de tiempo no planificado.

Mitigación del riesgo:

No se debería perder más tiempo del necesario en las pruebas y el manejo de los dispositivos, aunque estemos disfrutando en ello.

Si es necesario mucho más tiempo, se tendrá que decidir no se implementa algún requerimiento en la práctica, valorando la formación y aprendizaje realizado.

R5. Problemas de capacidad de hardware.

Definición del riesgo:

El proyecto pretende implementar un IDS/SIEM usando una Raspberry-pi que tiene un coste muy bajo y poca capacidad de proceso. Es parte del proyecto valorar si esto es posible, así como analizar las posibilidades de almacenamiento. Por lo tanto, entra dentro de las posibilidades de riesgo que el proyecto no sea viable.

Mitigación del riesgo:

En este caso no podemos hacer nada para mitigar el riesgo. Si al final el proyecto no funciona en la práctica, habrá servido de experiencia para el aprendizaje.

R6. Falta de tiempo para redactar la memoria.

Definición del riesgo:

Con la intención de conseguir cada uno de los objetivos de proyecto, puede que no se llegue a redactar la memoria a tiempo o que su elaboración cueste más de lo que se había planificado.

Mitigación del riesgo:

Durante el desarrollo de proyecto se deberá ir documentando todos los avances para que en la medida de lo posible se facilite la elaboración de la memoria final.

1.7. Estado del arte.

Aunque este apartado se ampliará en la fase de investigación, se han realizado una serie de búsquedas por internet para analizar las posibles herramientas que están a nuestra disposición para implementar el sistema.

Como el coste es una variable importante a tener en cuenta en este proyecto, se han buscado herramientas de software libre costes por licencia.

Herramientas y dispositivos disponibles:

Dispositivos hardware de bajo coste.

Dispositivos económico para IDS:

Raspberry-pi 3. El dispositivo seleccionado para la instalación del IDS es una Raspberry-pi 3 (1.2 GHz Quad-core ARM Cortex-A53, 1GB RAM, USB 2.0 y bajo consumo eléctrico)

Existen otros dispositivos similares (Odroid-H2,NanoPi NEO4, ODroid Xu4), pero se considera adecuada la selección de la Raspberry-pi por su bajo coste y gran popularidad.

Switch económico con port-mirroring:

Para enviarle todo el tráfico al IDS se necesita un Switch que utilice port-mirroring. Se ha buscado en internet un dispositivo adecuado al proyecto y con un coste asequible. Se ha seleccionado TP-Link TL-SG105E, con un coste de 24,9€, tiene cinco puertos a 1Gb y port-mirroring, cumpliendo así con los requisitos necesarios.

Sistema de detección de intrusos (IDS)

Los IDS de código abierto que destacan en la actualidad son Snort y Suricata. Snort es el más veterano (1998), pero Suricata destaca por ser compatible con la reglas de Snort y ser multihilo.

Snort.

Es un sistema de detección de intrusos de red, de software libre que puede ser instalado en sistemas Unix/Linux y Windows y fue creado por Martin Roesch en 1998. Tiene un lenguaje de regl0as muy extendido que se utiliza para definir los eventos y las alertas.

Suricata

Es otro conocido IDS también de software libre desarrollado por la Open Information Security Foundation. Utiliza el mismo lenguaje de reglas que Snort. Cabe destacar que Suricata puede funcionar en multi-hilo mientras que Snort no.

La selección de Suricata como IDS para el proyecto, viene dada por la motivación de aprendizaje sobre otro IDS distinto a Snort, que fue estudiado en la asignatura de seguridad de redes.

Sistemas de monitorización de eventos e información de seguridad (SIEM)

En este apartado se exponen una serie de herramientas que pueden ser utilizadas para implementar un SIEM. Se encuentran herramientas con finalidades diferentes (gestión de log, análisis y monitorización gráfica...) que deben interaccionar entre sí. El conocimiento para seleccionar qué herramientas son las mas adecuadas tiene aun bastante incertidumbre y hasta finalizar la fase de investigación no se tendrá una opinión clara al respecto.

De momento se enumeran algunas que se han encontrado para valorar:

- Zabbix
- Pandora
- Kibana
- Splunk
- Elastic
- Beast

HARDWARE

Raspberry Pi®

IDS

SNORT®

SURICATA

SIEM

ZABBIX

elastic

kibana

splunk>

logstash

beats

1.8. Recursos necesarios y presupuesto del proyecto.

Para llevar a cabo el proyecto, además de la inversión de tiempo ya planificada en los puntos anteriores, se necesitan algunos recursos materiales con un coste asociado:

Recursos	Uso	Coste
Ordenador personal.	Ordenador donde redactar la memoria. Instalar herramientas en laboratorio virtual. Instalar el SIEM	600€
Conexión a internet	Salida a internet que será monitorizada.	25€/mes
Switch con port mirroring	Switch que envíe el tráfico al IDS para su análisis.	24,99€
Dispositivo IDS	Raspberry-pi donde instalar el IDS	40€
Una familia	Una familia que cuidar cuando navegan por internet.	No tiene precio.

2. Fase de investigación

2.1. Captura de datos.

2.1.1. Definición de IDS

Dentro del ámbito de la seguridad, un Sistema de Detección de Intrusos (IDS) es un sistema de seguridad que nos permite monitorizar un entorno con la finalidad de detectar eventos que puede ser inapropiados o dañinos.

A modo de ejemplo, un sistema de detección de intrusos puede ser un sistema de cámaras que monitorizan las dependencias de una organización con la finalidad de detectar intrusos que susceptibles de realizar un robo o sabotaje.

2.1.2. IDS/IPS/NIDS de sistemas informáticos.

Si nos centramos en el ámbito de los sistemas informáticos, un IDS captura y monitoriza datos que se comunican por la red y o en los mismos equipos. Los IDSs que analizan el tráfico de red, se denominan NIDS (Network Intrusion Detection System).

Por otro lado también existe el concepto de IPS. Básicamente son IDS que además de monitorizar lo que ocurre en su entrono, pueden tomar acciones sobre el evento con la finalidad de neutralizarlo de forma automática. Esto tiene sus ventajas e inconvenientes, ya que por un lado la actuación es mucho más rápida al no tener que esperar a que alguien realice un análisis de la la información para actuar, pero por otro lado existe el riesgo de lanzar acciones equivocadas generadas por falsos positivos, lo que puede afectar al funcionamiento normal de un sistema, cosa que no es nada conveniente por ejemplo en un sistema industrial.

2.1.3. Interface en modo promiscuo

Básicamente, un NIDS funciona analizando los paquetes que circulan por la red en busca de coincidencias definidas previamente mediante unas reglas que identifican comportamientos sospechosos en la red.

Tenemos que tener en cuenta que en una red ethernet, los dispositivos de red solo aceptan los paquetes que van dirigidos al propio dispositivo o a la dirección de broadcast de la red (es decir a todos los dispositivos).

Para poder recibir todo el tráfico de red, el IDS debe establecer la interface de red que hará de sniffer en modo promiscuo. De esta forma el IDS podrá capturar todos los paquetes de red que tenga a su alcance, aunque no sea el destinatario del paquete.

2.1.4. Función de Port-mirroring

Hace no tantos años, se utilizaban hubs o concentradores para crear una red ethernet. En cada boca del hub se conectaba un dispositivo de red, y los paquetes de red pasaban por

el mismo canal de comunicación. Como se ha comentado en el punto anterior, cada dispositivo capturaba los paquetes que le iban dirigidos a él o los llamados broadcast. En este escenario, con poner la interface en modo promiscuo era suficiente para poder realizar una captura de todos los paquetes que pasan por la Ethernet, pero con la posterior utilización de switches este funcionamiento ha cambiado.

A diferencia de un hub, un switch detecta qué dispositivo está conectado en cada boca. Esto le permite reenviar los paquetes de red directamente al dispositivo al que va dirigido, realizando una especie de conmutación interna del tráfico de red hacia las distintas bocas, mejorando así el ancho de banda de la red y evitando colisiones entre paquetes. Además, este funcionamiento del switch, permite aumentar la seguridad de la red, ya que si alguien conecta un dispositivo en modo promiscuo a una boca del switch, solo recibirá los paquetes de red para los que es el destinatario. En cambio, ésto es un inconveniente a la hora de conectar un IDS al switch ya que no recibirá todo el tráfico de la red. Es por esta razón que se necesita utilizar un switch que con funcionalidad de port- mirroring.

Los switches con la funcionalidad port-mirroring, se pueden configurar para que todo el tráfico que pase por una boca, lo reenvíe a otra. De esta forma, si todo el tráfico que va y viene de internet lo reenviamos a la boca donde tenemos conectada la sonda IDS, le estaremos pasando todo el tráfico de acceso a internet para su monitorización y análisis.

2.1.5. Selección de un IDS sobre una Raspberry-pi [1]

Para cumplir con los objetivos del proyecto debemos de seleccionar un IDS de software libre que pueda funcionar correctamente en una Raspberry-pi.

Existen varios IDSs de software libre que podrían utilizarse, pero se quiere ampliar el estudio de reglas de tipo Snort, con lo que nos centramos en dos IDS:

Snort.[2]

Con largo recorrido, desde 1998, Snort es una de las mejores opciones como IDS de software libre ya que está ampliamente extendido y tiene una importante comunidad que lo van enriqueciendo continuamente.

Puede actuar en nuestra red como IDS y como IPS. Funciona escuchando el tráfico de red y contrastándolo con una serie de reglas que le definen los patrones a buscar pudiendo detectar de esta forma anomalías o incidentes de seguridad.

Se pueden utilizar una base de datos de reglas que proporciona la comunidad de Snort o utilizar reglas propias adaptadas a las necesidades particulares. Existen gran cantidad de repositorios de reglas de Snort que se han creado a lo largo de varias décadas. Además se dispone de una extensa documentación sobre Snort.

[1] https://resources.infosecinstitute.com/open-source-ids-snort-suricata/#gref
[2] https://www.snort.org/

Snort no es multihilo, con lo que solo utilizará un núcleo de la CPU independientemente de los que se tengan disponibles. Esto, junto con el aumento de amenazas, la complejidad cada vez mayor de las reglas y el aumento del tráfico de red, hacen que la arquitectura de Snort encuentre algunas dificultades. La versión 3 de Snort pretende solucionar estos problemas, pero de momento el producto aun está en versión Alpha. Por otro lado, Suricata nace sobre 2009 para intentar superar algunos de estos desafíos.

Suricata.[3]

Suricata es un IDS de software libre desarrollado por OSIF (Open Information Security Fundation), pero también puede funcionar como IPS actuando sobre el trafico de red y analizando los ficheros capturados.

Suricata sigue el concepto básico de Snort, pero aprovecha el hardware moderno e incluye soporte para el lenguaje de script LUA que puede utilizarse tanto en la salida como en la definición de reglas ayudando a la detección de amenazas más complejas.

Suricata utiliza un sistema de reglas que es compatible con Snort, así como los formatos de salida, lo que le facilita la migración de un sistema a otro. El hecho de ser compatible con las reglas de Snort, es un buen punto de partida para Suricata, ya que las reglas dan el conocimiento necesario a los IDS para detectar anomalías.

Por otro lado, se puede configurar el sistema de entradas y salidas en diferentes formatos como JSON, lo que facilita la integración con sistemas SIEM.

Otra de las características interesantes de Suricata, es la extracción de archivos. Esto permite por ejemplo extraer todos los archivos seleccionados por una regla para un posterior análisis.

Finalmente cabe destacar que Suricata es multihilo, lo que le permite repartir los procesos en los diferentes núcleos del sistema. Esta es una de las características que le da clara una ventaja sobre Snort.

Elección final del IDS.

Para la implementación de proyecto se decide utilizar Suricata por las siguientes razones:

- Compatibilidad de las reglas con Snort, la cuales ya se han utilizado en otras asignaturas del master.
- La característica de multi-hilo puede ayudar a mejorar el rendimiento del sistema al aprovechar mejor las características de una Raspberry-pi, la cual dispone de un procesador ARM de cuatro núcleos.
- En general Suricata evoluciona y mejora Snort.

[3] https://suricata-ids.org/

2.2. Topología de red

La topología de red a utilizar para implantar un IDS/IPS puede ser muy variada. En lo que se refiere a este proyecto y en el ámbito de una red doméstica, vamos que plantearnos dos escenarios.

Cada escenario va en función de decidir si el IDS va ha realizar tareas de IPS o no. Esto influye de forma importante en la topología.

Si la sonda va a actuar como IPS debería estar situada en un punto de red que pueda actuar sobre el tráfico de forma que puede denegar a borrar paquetes de red. En este caso podemos decir que la sonda hace funciones de Gateway.

En el caso de solo actuar como IDS, la sonda se puede situar off-line. En este caso habría que utilizar técnicas como port-mirroring para enviarle el tráfico a analizar.

En los puntos siguientes se valorarán los dos modelos de topología de red planteados, así como distintos componentes de red necesarios para su implementación:

2.2.1. IDS/IPS on-line. Haciendo funciones de gateway.

En este caso, el IDS suele realiza funciones de Gateway. Esto significa que se interpone entre los dispositivos de la red y el acceso a internet de forma que en el caso de detectar tráfico de red no adecuado pueda eliminarlo en el momento parando así la comunicación.

Podemos montar esta topología de red únicamente con los siguientes dispositivos de red.

- Router acceso a internet.
- Raspberry-pi con sonda IDS configurada a la vez como AP.

El diagrama de red sería el siguiente:

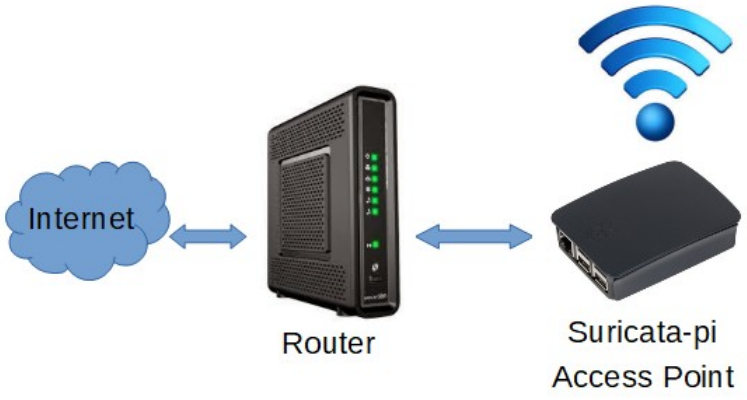


Ilustración 1: Topología de red. Raspberry-pi in-line

Ventajas e inconvenientes:

Ventajas:

- Posibilidad de realizar funciones de IPS, actuando sobre un posible ataque y eliminando paquetes si es necesario.
- En el caso de control parental, permitiría el bloqueo a páginas con contenido no adecuado.
- Puede actuar de cortafuegos.
- Ante una amenaza puede actuar en el momento.
- No necesitamos un switch con port-mirroring
- Como la Raspberry tiene dos interfaces de red (una ethernet y otra wifi), se podría que configurar como punto de acceso, reduciendo así en número de dispositivos de red.

Desventajas:

- Puede crear cuellos de botella, sobre todo si el dispositivo no es muy potente como es el caso de una Raspberry pi.
- El montar la Raspberry-pi como punto de acceso, le asigna una funcionalidad más al aparato que puede penalizar en el rendimiento global del sistema.

2.2.2. IDS off-line. Monitorizando con port-mirroring

Si solo se van a realizar tareas de monitorización, la sonda IDS no tiene por que estar ubicada como punto intermedio del tráfico que sale a internet, sino que la podemos dejarla a un lado y pasarle todo el tráfico con la función de port-mirroring de un switch preparado para ello.

Podemos montar esta topología de red con los siguientes dispositivos de red.

- Router acceso a internet.
- Raspberry-pi como AP
- Switch con port-mirroring
- Punto de acceso wifi

El diagrama de red sería el siguiente:

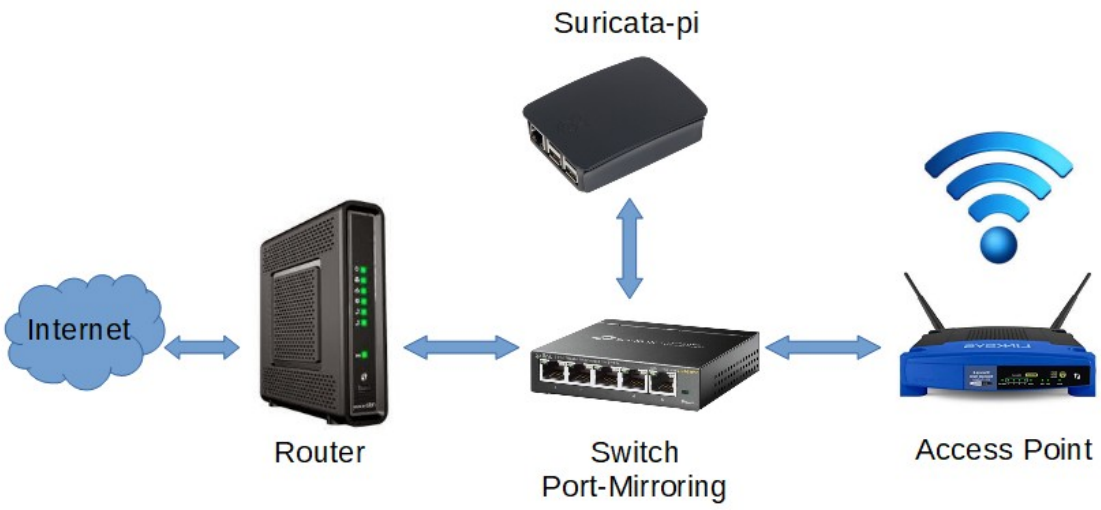


Ilustración 2: Topoligía de red. Raspberry-pi off-line

Ventajas e inconveniente:

Ventajas.

- No afecta al rendimiento de acceso a internet, ya que evitamos que todo el tráfico tenga que atravesar el IDS.
- Es más factible de implementar en dispositivos poco potentes como es una Raspberry pi

Desventajas.

- No podremos realizar funciones de IPS sobre el tráfico en el caso de detectar trafico malicioso.
- A nivel de control parental, no podremos restringir acceso a páginas con contenido inadecuado.

2.2.3. Elección de una topología de red para el proyecto.

En este proyecto se pretende implementar un sistema que permita la monitorización. La elección se basa en el siguiente razonamiento sobre los pros y contras vistos anteriormente:

- Se parte de la utilización de una Raspberry pi como sonda IDS. Teniendo en cuenta que no es un dispositivo muy potente, si además se le asignan funciones de Gateway haciendo que todo el tráfico tenga que pasar por el dispositivo, es muy probable que se genere un cuello de botella y disminuya la velocidad de acceso a internet. No obstante se realizará una PoC (prueba de concepto) para comprobarlo ya que si no fuera por este inconveniente sería una topología interesante a implementar al necesitarse menos dispositivos de red para ponerla en marcha.
- Al no poder utilizar funciones de IPS, se pierde la capacidad de bloquear contenidos en tareas de control parental. A pesar de ello, se considera que las funciones de monitorización ya son útiles de por sí ya que les permitirá a los padres saber qué uso de se está haciendo de internet. En el caso de detectar alguna información sobre la que se tenga que actuar, se deja esta labor de mediación y concienciación para los propios padres con sus hijos.

2.3. Posibilidades de Suricata IDS en una Raspberry-pi.

2.3.1. Instalación de Suricata en una Raspberry-pi.

Existe numerosa documentación en internet que explican como instalar Suricata en una Raspberry-pi, pero siempre es necesario realizar pruebas del rendimiento que demuestren que el dispositivo es capaz de actuar como Sonda IDS.

Este proyecto se centra en el ámbito familiar, donde se supone que la cantidad de tráfico a analizar es mucho menor que en un entorno empresarial. Por lo tanto, uno de los objetivos del proyecto será demostrar que una Raspberry-pi es un dispositivo que aun siendo de bajo coste, puede funcionar como sonda IDS en el ámbito familiar.

2.3.2. Características de Suricata.

Adentrándonos en las posibilidades que nos ofrece Suricata[4], podemos comprobar su guía de usuario, algunas funcionalidades que pueden ser interesantes para los objetivos del proyecto:

Suricata puede funcionar como IDS/IPS/MSM.

Suricata puede realizar labores de IDS para detección de intrusos y amenazas, pero también puede actuar como MSM (Network Security Monitoring) monitorizando las peticiones DNS. Esto puede ser útil para vigilar la navegación por internet que se realiza desde la red, y de esta forma poder realizar un adecuado control parental al respecto.

Reglas compatibles con Snort

Para la detección de amenazas de seguridad se utilizará las reglas que hay desarrolladas por la comunidad y son compatibles con Snort.

Por otro lado se puede utilizar reglas específicas para el control parental. Se estudiará más adelante las necesidades al respecto y la posibilidad de definir reglas para este propósito.

Multihilo.

A pesar de lo pequeño del dispositivo, una Raspberry-pi posee un procesador con cuatro núcleos. Suricata permite ajustar su configuración para repartir los proceso en diferentes núcleos y así mejorar el rendimiento.

Seguimiento de navegación http

Como características de monitorización de redes, Suricata permite guardar todas las peticiones HTTP entrantes y salientes en un fichero log en formato Log Apache. Esto puede ser interesante a nivel de control parental.

[4] https://suricata.readthedocs.io/en/suricata-4.1.3/index.html

Captura de ficheros.

Otra característica que puede ser interesante es la posibilidad de descargar los ficheros binarios del tráfico de red como podrían ser fotos, videos, documentos... Si bien a primera vista parece una característica interesante, debe valorarse bien si utilizarla o no, ya que posiblemente aumente las necesidades de almacenamiento que tengamos que utilizar.

Control de peticiones DNS.

Todas las llamadas DNS también son monitorizadas por Suricata y almacenadas en unos logs. Esto puede sernos útil para tener un control de las llamadas DNS que realizan nuestros dispositivos.

Eve JSON Output

Permite la salida de todo tipo de eventos en formato JSON que facilita la integración con herramientas de terceros.

2.3.3 Definición de reglas para detección de eventos.

Para el funcionamiento de un IDS como Suricata, se utilizan una declaración de reglas que le especifica qué eventos tiene que detectar sobre el tráfico de red monitorizado.

Suricata y Snort utilizan el mismo formato para la declaración de reglas de detección.

Hay paquetes de reglas ya definidas para detectar posibles ataques o malware, pero también es posible definir reglas propias de detección. En el caso de este proyecto se pretende definir alguna reglas útil para el control parental.

Veamos cual es la estructura de una reglas de un IDS Snort/Suricata.

Estructura de las reglas del IDS:

La reglas se dividen en tres partes:

1. Acción. Que determina que hacer cuando la regla detecte un paquete. Las opciones puede varias, como por ejemplo ***Alert***, para indicar una alerta, u otras como ***drop*** (para borrar el paquete en caso se usar funciones de IPS) , **log**, **pass**...

2. Cabecera. Donde se especifica el origen y destino la la comunicación sobre la que se realizará una acción.

La estructura de la cabecera tiene los siguientes campos:

- **Protocolo**: Establece el protocolo de comunicación (TCP, UDP, ICMP…)
- **Red origen:** Define una IP o una red origen. Se pueden utilizar las variables $EXTERNAL_NET o $HOME_NET definidas en el fichero de configuración del IDS.
- **Puerto origen:** Define el puerto de origen.

- **Dirección**: Especifica la dirección de la comunicación con los siguientes símbolos (< , <> , >).
- **Red destino:** Define la red de destino.
- **Puerto destino:** Se define el puerto de destino y puede utilizarse para filtrar paquetes de una determinada aplicación conociendo el puerto que utiliza.

3. Opciones de la regla.[5]

La segunda parte de la regla son las opciones. Se definen entre paréntesis y cada opción está separada por punto y coma (;), luego de cada opción se puede definir la clave y su valor separado por dos puntos (:).

Se ha realizado un estudio de las opciones que pueden ser útiles para el proyecto.

Claves Meta.

Son claves que no tienen efecto sobre la inspección que realiza Suricata, pero pueden ayudar a clasificar las alertas para tratarlas en los informes finales.

msg: Nos sirve para dar un mensaje sobre la alerta presentada. Puede ser muy útil para luego poder manejar y clasificar la información en los logs.

sid y rev: Asignan un código y versión a la regla.

classtype: Clasifica la regla en algún grupo. En este proyecto las que definamos las clasificaremos como “parental-control”. La clasificación la podemos encontrar en el fichero de configuración *classification.config*

Claves ICMP.

Se refieren al protocolo ICMP permitiendo especificar códigos de respuesta concretos.

itype e **icode**: pueden ayudarnos a filtrar mensajes concretos de ICMP para en el caso de realizar un ping a los dispositivos, detectar las respuestas y crear una alarma. La idea sería luego tratar estas alarmas para calcular tiempos de activación del los dispositivos. Realizaremos pruebas en la fase de implementación para comprobar si la idea es factible o es mejor utilizar otros métodos.

Claves Playload

Estas claves son muy útiles para la detección, por que se utilizan para inspeccionar el contenido del paquete o stream de red.

Content: Es una de las claves más importantes. Con ella podemos especificar un valor que queremos detectar en el contenido del paquete.

Sobre la clave **content**, se le pueden aplicar otras claves (modificadores de content), las cuales declaran después de content y tiene la finalidad de ajustar mejor el patrón de búsqueda.

[5] https://suricata.readthedocs.io/en/suricata-4.1.3/rules/index.html

Algunos de estos modificadores de **content** que podemos encontrar en la guía son **depth, startswith, offset, distance, within** ente otros, o **pcre** que permite el uso de expresiones regulares Perl.

Claves http.

Añadido a las posibilidades de los modificadores de búsqueda de content, las claves para http ofrecen opciones especificas para el protocolo e aplicación HTTP. Esto puede ser útil si se pretende utilizar reglas de control a la navegación utilizando el protocolo HTTP tanto a los flujos de datos de **request** como **response.**

Algunas claves de este tipo que pueden ser útiles son **http_method, http_host, http_user_agent**

Claves File

Son algunas claves para detectar propiedades de ficheros como nombre, extensión, tamaño.

Algunos ejemplos son **filename, fileext, filemagic, filesize** o **filestore** que permite guardar el fichero es detectado por la regla.

Claves de Reputación IP

La funcionalidad de Suricata de reputación de IPs se puede utilizar con la clave **iprep** donde podremos indicarle si se activa la regla según los parámetros que le indiquemos determinando por ejemplo la categoría y los puntos de reputación que tiene.

iprep:<side a chechear>,<categoría>,<operador (<, =, >)>,<puntos de reputación >

Algunos ejemplos de reglas:

A modo de ejemplo y para que pueda verse la sintaxis de la declaración de una regla, se muestran a continuación varios ejemplos.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any
(msg:"Detección de PING";
content:"Scarlett"; nocase; classtype:policy-violation; sid:1; rev:1;)
```

Para detectar utilización de un ping desde fuera de mi red a algún equipo de la red local.

```
alert http any any -> any any
(msg:"Buscando palabra prohibida";
content:"sex"; nocase; classtype:parental-control; sid:1; rev:1;)
```

Detecta la utilización de la palabra “sex” en la navegación HTTP.

2.3.4. Herramientas de apoyo.

Como se ha comentado en puntos anteriores, utilizar una topología de red donde tuviéramos que una Raspberry-pi in-line a la salida de internet, crearía un cuello de botella reduciendo así la calidad del servicio de acceso a internet.

Esto quiere decir que Suricata no puede hacer tareas de IPS, con lo que no puede usar la acción **Drop** sobre aquellos paquetes de red identificados como peligrosos.

No obstante se pretende valorar la posibilidad de establecer restricciones utilizando junto con el IDS otras herramientas como por ejemplo:

Servidor DHCP.

Si en lugar de ser el router de acceso a internet quien asigna las direcciones IP de nuestros dispositivos, hacemos que lo haga las Raspberry-pi, conseguiríamos asignar una IP fija a cada dispositivo en función de su mac. Esto mejora la seguridad del sistema y nos ayuda a tener una clara identificación de cada dispositivo, considerándolo una opción interesante a incluir en el proyecto.

Se comprobará si afectará al rendimiento del sistema, pero siendo la asignación de IP una tarea que se realiza en el inicio de conexión de un dispositivo, se espera que no sea así.

Un inconveniente que existe al incluir esta funcionalidad es la dependencia que se crea con la Raspberry-pi, teniendo siempre que estar en funcionamiento para que ofrecer este servicio de DHCP cada vez que un dispositivo solicita una dirección IP.

Servidor DNS.

Instalar un servidor DNS en la Raspberry pi, es otra posibilidad que se puede valorar.

Esto permitiría controlar el servicio de acceso a internet ya que este es muy dependiente de las llamadas DNS. No hay que descartar una posible caída de rendimiento que habrá que valorar.

Además es posible que utilizando el servicio DNS junto con un sistemas de reputación de IP o dominios, se pueda denegar el acceso a ciertas páginas de contenido no adecuado para menores. Otra posibilidad a valorar es denegar el servicio DNS a un determinado dispositivos en un determinado horario.

Utilidades de apoyo.

Es posible que se necesite el uso de otros programas para el control de dispositivos conectados a la red como pueden ser **fping** o **nmap.**

2.4. Análisis y visualización de datos

En la primera parte se ha analizado lo que son los sistemas IDS y como se pueden utilizar para capturar todo tipo de información del tráfico de red. Pero estos datos

capturados se almacenan en ficheros tipo Log difíciles de leer y analizar incluso para el personal técnico.

Es necesario un sistema que permita transformar estos datos en información útil para personas que no están familiarizadas con los sistemas de información, como por ejemplo unos padres de familia.

Por esta razón, junto a la implantación de un IDS, suele ser necesaria la instalación de un SIEM.

El significado del termino SIEM es una combinación de las siglas SIM y SEM.[6]

- SEM. (Security Event Management) Se encarga de la monitorización y correlación de eventos de seguridad en tiempo real.
- SIM. (Security Information Management). Se enfoca en el análisis y monitorización de datos almacenados durante un periodo de tiempo.

Por lo tanto, implantar un SIEM significa tener un sistema que permita gestionar la información de la seguridad y eventos, ofreciendo la posibilidad de recopilar, analizar y presentar esta información, transformando así los datos recopilados por la sonda IDS,en información fácilmente accesible por el usuario que acabe siendo conocimiento útil para la toma decisiones.

En este proyecto, la implantación de un SIEM se considera necesaria para realizar una monitorización del uso de internet de una manea visual al usuario.

Esta información sobre las páginas o los servidores a los que se accede, se va guardando en los diferentes Logs que proporciona Suricata, así como cualquier alarma detectadas por las reglas del IDS. Por otro lado, podría ser necesario analizar datos de otros sistemas para estudiar los tiempos de conexión a internet de los distintos dispositivos, o cualquier otra información que pueda ser interesante para los objetivos del proyecto.

2.4.1. Tratamiento de datos. La gestión de Logs

Como se ha comentado, una tarea importante a realizar será recopilar datos de una serie de ficheros Logs donde el IDS u otros sistemas van realizando apuntes de todo lo que van detectando. Por lo tanto el SIEM debe ofrecemos herramientas que faciliten esta labor.

Estas herramientas deben de ayudarnos a resolver una serie de problemas asociados al manejo de ficheros Logs, como por ejemplo:

Diferentes formatos de LOG.

Podemos encontrar diferentes tipos de Logs, cada uno con un formato y campos distintos. Necesitaremos por lo tanto una herramienta que unifique los formato y filtre los datos que se quieren analizar.

[6] https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi_diseno_configuracion_ips_ids_siem_en_sci.pdf

Localización de log en diferentes lugares.

Estos ficheros de Log pueden estar ubicados en diferentes rutas o sistemas. Es posible que se necesite leer datos de diferentes fuentes y centralizarlos en un lugar para su posterior análisis.

Hay que conocer los datos que contienen lo logs

Para mostrar la información de forma amigable, hay que transformarla y adaptarla. Será necesario por lo tanto conocer bien el contenido que cada Log a tratar, para poder adaptarlo a las necesidades.

Capacidad de almacenamiento.

Una Raspberry-pi no posee mucha capacidad de almacenamiento. Es por ello que para evitar que los Logs aumenten de tamaño y acaben bloqueado el sistema, se deben de tomas alguna medidas preventivas:

- Definir una política de rotación y compresión de Log que eviten un consumo de espacio excesivo.
- Valorar la posibilidad de almacenar o enviar los Logs a una máquina externa.
- Aumentar la capacidad de almacenamiento de la Raspberry-pi, añadiéndole algún disco externo o unidad USB .

2.4.2. Sistemas SIEM a nuestro alcance.

Hoy en día tenemos nuestro alcance muchas herramientas que nos permiten implementar un SIEM. Nos centraremos en las que sean se software libre ya que podemos encontrar sistemas de gran calidad, sin necesidad de invertir en licencias de software. Por otro lado, debemos elegir un sistema flexible y que tenga un buen soporte de la comunidad.

Se ha investigado por internet algunas herramientas con la finalidad de analizar eventos de seguridad de nuestro IDS, como son Snorby, Sguil, Splunk, Zabbix y la pila ELK.

Finalmente se decide utilizar la pila ELK de la empresa Elastic como sistema principal de motorización de datos.

Las razones de esta elección son las siguientes:

- Es un sistema que no solo permite analizar datos del IDS Suricata, sino que puede utilizarse para recoger datos de todo tipo de logs y tratarlos.
- Permite una monitorización de datos sencilla y amigable, permitiendo diseñar un dashboard a medida de las necesidades.
- Puede instalarse en una Raspberry-pi. Se valorará más adelante si esto puede afectar en una caída importante sobre rendimiento global del sistema.
- Goza de gran popularidad en la comunidad de internet.

- Existe una motivación de formación personal sobre esta herramienta.

2.4.3. Posibilidades de implementación de SIEM con una Raspberry-pi

Teniendo en cuenta las posibilidades de proceso y almacenamiento de una Raspberry-pi, se debe valorar si es factible la instalación del SIEM en la propia Raspberry-pi o es conveniente instalarlo en otro equipo con más capacidad de proceso y posibilidades de almacenamiento.

Se realizará una PoC (Prueba de Concepto) sobre la instalación y rendimiento del sistema SIEM en la propia Raspberry-pi en la fase de implementación, de forma que se pueda decidir si es necesario desplegar el SIEM en una tercera maquina o no.

En el caso de utilizar otro equipo se tendrá que tener en cuenta las siguientes cuestiones:

- Definir un canal de comunicación de los ficheros Log que se generan, para su posterior análisis.
- Es deseable que este tercer equipo (que puede ser un PC), teniendo en cuenta que no estará siempre en marcha, sino solo se utilizará el sistema para consultas.
- Si utilizamos un PC para montar el SIEM, se debería valorar la posibilidad de instalarlo en Windows, ya que es más amigable y sencillo en entornos domésticos para usuario no experimentados en Linux o en maquinar virtuales. Al final se debe buscar siempre sencillez de uso.

Se ha realizado una valoración de pros y contras de los dos posibles escenarios:

1. Instalación de un SIEM en la propia Raspberry-pi.

Ventajas:

- Se consigue tener un producto todo en uno.
- Permite una mejor distribución y configuración.

Desventajas:

- Puede generar problemas de rendimiento en el IDS.
- Sería necesario añadir un dispositivo de almacenamiento a la Raspberry-pi.

2. Instalación de un SIEM en otro equipo.

Ventajas:

- No genera problemas de rendimiento sobre el IDS.

Desventajas:

- Es necesario un sistema de comunicación de ficheros Log entre sistemas.
- El equipo no debería estar siempre en marcha. Solo cuando se necesita hacer consultas.

- Se instalaría en Windows ya que es el SO más utilizado en el ámbito familiar.

2.4.4. La pila ELK.

La pila ELK es la solución es la solución elegida para implementar el SIEM. Es un proyecto software libre de la empresa Elastic[7], y permite realizar un completo análisis de Logs con la finalidad de transformar los datos recopilados en información visual y útil.

Realmente está compuesta por varias herramientas independientes que interaccionan entre sí.

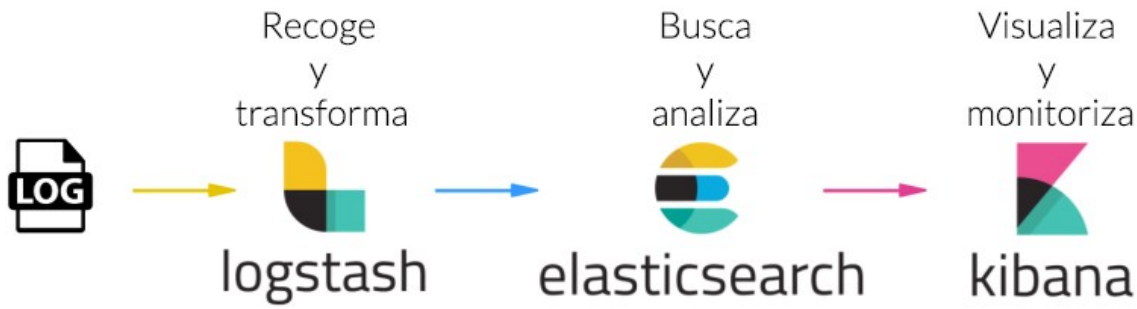


Ilustración 3: Proceso herramientas Elastic

La pila de productos de Elastic se compone principalmente de tres herramientas:

Logtash.[8]

Es la Herramienta encargada de manejar los ficheros Log. Recopilando , parseando y enriqueciendo datos si es necesario, para finalmente poder enviarlos a Elasticsearch.

Debe permitir resolver los problemas del manejo de ficheros Log comentados anteriormente, definiendo canales de datos que recolectan datos estructurados y desestructurados generados por varios sistemas y formatos.

Para realizar esto, Logstash utiliza tres tipos de plugins:

Input plugin.

Se utilizan para indicar de donde se van a recoger la entrada de datos. Hay una variedad importante que nos ofrece la flexibilidad para alimentar el sistema desde diferentes entradas como pueden ser, ficheros, syslog del sistema, beats (componentes instalados en sistemas para obtener datos), salidas de ficheros ejecutables.

[7] https://www.elastic.co/es/
[8] https://www.elastic.co/es/products/logstash

En el proyecto tendremos que seleccionar los ficheros Log que nos ofrece Suricata tanto para obtener datos de alertas, como navegación HTTP o peticiones DNS.

Filter plugins.

Estos plugins son los encargados de tratar los datos recopilados, para adaptarlos a nuestras necesidades o unificarlos.

Output plugins.

Son los que permiten definir a donde enviamos finalmente los datos tratados. Normalmente en nuestro caso se utilizará el envío de los datos a ElasticSearch, pero pueden enviarse a ficheros csv, ficheros de texto, servidores en la nube o a otros sistemas de monitorización como Zabbix.

ElasticSearch.[9]

Tal y como indican en la página oficial, ElasticSearch es el corazón de ELK, actuando como un motor de búsqueda y análisis RESTful distribuido, capaz de resolver de forma rápida todo tipo de consultas sobre gran cantidad de datos almacenados.

Se suele situar en medio de las otras dos herramientas, alimentándose de datos ya tratados por LogStash y ofreciendo información a Kibana para su visualización.

Su base de datos no relacional, almacenan e indexan los datos para luego realizar búsquedas que permiten la combinación de datos estructurados, no estructurados, así como utilizar agregaciones para estudiar tendencias y patrones.

La indexación de datos hace que las búsquedas sean más rápidas. Se realizarán las pruebas oportunas para valorar si es posible instalarlo en una Raspberry-pi o si necesitamos un tercer equipo para su ejecución.

Kibana[10]

Es herramienta de visualización, que se integra con ElasticSearch y nos permite diseñar el front-end a modo de Dash-board con diferentes tipos de gráficos, para mostrar la información de una forma amigable al usuario.

[9] https://www.elastic.co/es/products/elasticsearch
[10] https://www.elastic.co/es/products/kibana

En el interface de Kibana se diferencian tres apartados.

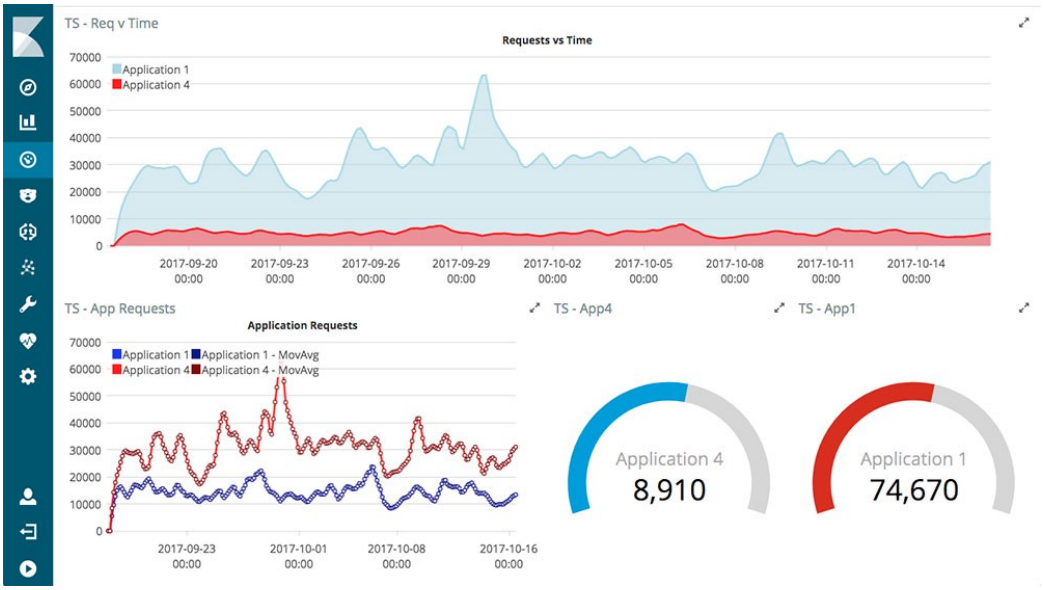


Ilustración 4: Ejemplo de interface KIBANA

- **Discover**: Donde se indican los datos a recoger. Seleccionando origen de datos, filtrando campos...
- **Visualizations**. Donde define y configura las visualizaciones de datos, seleccionando el tipo de gráficos, la composición de valores en los ejes X e Y...
- **Dashboart**. Es donde se diseñan los tableros de información, seleccionando las distintas visualizaciones de datos (gráficos, tablas…) que se han definido previamente.

2.5. Peligros en la red y la mediación parental.

Nuestros hijos usan de forma natural la tecnología desde edades muy tempranas utilizando una gran cantidad de dispositivos que tienen a su alcance como móviles, tables, ordenadores, consolas de videojuegos. Hoy en día cualquiera de estos dispositivos está conectado a internet ofreciendo acceso a información y experiencias en todo momento y en cualquier lugar.

Todo esto es positivo y ayuda comunicarse, aprender, crear… pero también tiene sus peligros y los padres primero deben conocerlos y luego ejercer su función de protectores de la misma forma que ocurre en la vida real.

El problema es que muchos padres de hoy se sienten superados por los avances tecnológicos y sus hijos navegan por internet completamente solos a edades muy tempranas, y sin haber tenido una mínima educación que les permitan tener las precauciones adecuadas.

Esto comporta una serie de peligros para el menor como por ejemplo:

- Sextorsión
- Grooming
- Sexting
- Acceso a contenido no adecuado a la edad.
- Ciberbullying
- Contacto con desconocidos
- Tecnoadicción.

Este proyecto tiene el objetivo de ayudar a los padres a realizar un adecuado control parental, utilizando las dos estrategias de mediación parental que se definen en la "Guía para el uso seguro y responsable de internet por los menores"[11] de la OSI (Oficina se Seguridad del Internauta):

- **La mediación activa:** supervisión acompañamiento y orientación. Se basa básicamente en prestar atención a lo que hace tu hijo cuando está conectado a la red y concienciarse sobre el uso adecuado de las TIC.
- **La mediación restrictiva:** establecer reglas sobre el uso de internet. Se basa principalmente en establecer normas de uso y límites sobre el uso de la red para evitar los riesgos.

Se pretende con ello que el sistema implantado sea una herramienta de ayuda al control parental que deben realizar los padres en el acompañamiento a sus hijos ayudándoles a

[11] https://www.is4k.es/sites/default/files/contenidos/herramientas/is4k_guia_mediacion_parental_internet.pdf

conocer qué uso real se está haciendo del acceso a internet en sus familias, para de esta forma poder dialogar y concienciar adecuadamente a sus hijos.

2.5.1. La monitorización de la red con un IDS para el control parental

Como se ha comentado anteriormente las herramientas IDS están diseñadas para detectar amenazas de seguridad. En este proyecto se pretende utilizar para que además de esta tarea, se realice una labor de control parental.

Actualmente existe muchos sistemas de control parental que permiten controlar el uso de nuestros hijos de las nuevas tecnologías, pero la mayoría consisten en instalar aplicaciones en los distintos dispositivos. Hay que tener en cuenta que hoy en día, en cualquier hogar se pueden encontrar varios dispositivos conectado a internet y con sistemas operativos distintos, lo que complica el tener un sistema de control parental instalado en cada dispositivo.

La herramienta de control parental propuesta en este proyecto, no necesita instalar aplicaciones en los distintos dispositivos que tenemos en las casas. En lugar de esto, se utilizará la sonda IDS que vigila la salida a internet de forma centralizada.

El sistema de control parental propuesto se compone del dispositivo Raspberry-pi con el IDS Suricata instalado, pero también puede hacer uso de cualquier herramienta añadida que nos sea útil para los objetivos del proyecto, como pueden ser la utilización de servidores DNS o DHCP.

Para valorar las posibilidades de la herramienta de control parental que se propone en el proyecto se plantea realizar un pequeño análisis de riesgos sobre las amenazas de internet para los menores comentadas anteriormente.

2.5.2. Amenazas a las que se enfrentan los menores en la red.

En el siguiente punto se enumeran las diferentes amenazas a las que están expuestos los menores cuando utilizan internet. Sobre estás amenazas se calculará el riesgo dependiendo de la posibilidad y gravedad de que ocurra y finalmente se planteará qué salvaguardia podría ofrecernos el sistema propuesto para reducir el riesgo.

Enumeración de amenazas a las que se expone un menor con el uso de las nuevas tecnologías:

Amenaza 1. Acceso a contenidos inapropiado.

Con el acceso a internet, nuestro hijos tiene acceso a una cantidad de información. Esto es muy positivo a la hora de adquirir nuevos conocimientos pero también puede se perjudicial para su crecimiento al acceder a contenidos inapropiados para su edad (gestos obscenos, sexo explícito, erotismo, violencia, crueldad, odio, prácticas ilegales, drogadicción, etc).

Los padres pierden el control sobre el contenido al que acceden sus hijos, ya que es difícil estar continuamente a su lado mientras utilizan cualquier dispositivo conectado a internet.

Amenaza 2. Tecno-adicción.

Es uno de los problemas que más está preocupando a los padres, no tanto por la gravedad, sino por que es muy común entre los menores de hoy en día.

Muchos menores (y algunos adultos), no tienen control del tiempo que se pasan conectados, y pueden llegarse a generar dependencia del uso. Aquí se incluye el uso de todo tipo de dispositivos, desde móviles, ordenadores y hasta videoconsolas.

Es un problema preocupante y en aumento, de hecho OMS reconoció que la adicción a los videojuegos es un desorden de salud mental e incluyó esta problemática en la Clasificación Internacional de Enfermedades. [12]

Amenaza 3. Descarga de malware

Los menores en su navegación por internet pueden fácilmente, si no tenemos protegido los equipos adecuadamente, descargar e instalar involuntariamente algún tipo de malware que puede comprometer la seguridad de los equipos.

Amenaza 4. Contacto con desconocidos.

Al igual que no deseamos que nuestros menores hablen con desconocidos cuando están jugando en un parque, se debe de evitar que lo hagan cuando navegan por internet ya que esto puede llevarlos otros peligros más graves. Hay que tener en cuenta que en cualquier juego o aplicación en principio inofensivo, puede tener una función de chat que le permita el contacto con extraños.

Amenaza 5. Ciberbullying

Se podría decir que es similar al acoso escolar que podía sufrir un menor de sus compañeros hace veinte años atrás en el colegio, pero ahora con uso de las nuevas tecnologías y su difusión a través de internet, el daño que se produce y la capacidad de propagación se ha disparado. Se trata de un acoso difícil de detectar por los padres y profesores, por lo que la principal arma que se tiene contra esta amenaza es la concienciación de los propios chavales.

Amenaza 6. Grooming

Se trata de un acoso sexual de un adulto hacia un menor utilizando las nuevas tecnologías. El acosador sexual establece lazos de amistad con la víctima utilizando personalidad falsa durante un tiempo hasta que consigue algún vídeo o imagen comprometida del menor. A partir de ese momento comienza el ciber-chantaje. La

[12] https://icd.who.int/browse11/l-m/en#/http%3a%2f%2fid.who.int%2ficd%2fentity%2f1448597234

víctima cede al chantaje de enviar más imágenes comprometida por miedo, entrando así en un círculo difícil de romper y de fatales consecuencias psicológicas para la víctima.

Amenaza 7 . Sexting.

El sexting consiste en la publicación de contenidos de tipo sexual, producidos por el propio remitente utilizando el móvil o webcams. Muchas veces ocurre entre las propias parejas de jóvenes, que una vez terminan con la relación, pierden el control del contenido publicado. Puede derivar en otros peligros como la sextorsión, grooming o pornografía infantil.

Propuestas de salvaguarda del sistema propuesto en el proyecto.

Para poder hacer frente a las amenazas, el sistema propuesto en este proyecto pretende ofrecer una serie de salvaguardas que ayuden a mitigar los riesgos:

Salvaguarda 1. Monitorizar contenido http.

Sería interesante poder almacenar información de los acceso a internet realizados por cada dispositivos para poder analizarlos cuando queramos.

Suricata tiene la posibilidad de registrar todo el contenido HTTP en el el fichero http.log con formato Apachelog. Esto puede ser útil para poder filtrar contenido a páginas HTTP.

Salvaguarda 2. Monitorizar llamadas DNS.

Suricata puede almacenar en el fichero dns.log todas las peticiones DNS. Esto permite poder monitorizar más tarde todas las llamadas. Sería interesante poder identificar las aplicaciones utilizadas según las peticiones DNS realizadas.

Salvaguarda 3. Alertas cuando se acceda a páginas de mala reputación.

Puede ser útil que el sistema nos generara algún tipo de alarma si se detecta alguna solicitud hacia una página clasificada como inapropiada para niños. Se puede investigar de implementar reglas que se disparen si se intentan acceder a estas páginas. Se estudiará esta posibilidad en la fase de implementación.

Salvaguarda 4. Alertas cuando se detecte contenido de palabras prohibidas.

Suricata puede analizar el contenido de los paquetes de red, con lo que se podrían crear unas reglas que lanzarán una alarma si se detecta el contenido de alguna palabras “prohibida” que deba llamarnos la atención..

Salvaguarda 5. Eliminar acceso a páginas de mala reputación filtrando con DNS.

Ya que no podemos utilizar la Raspberry-pi como IPS para evitar cuellos de botella en el acceso a internet, se puede estudiar establecer un sistema de restricción de acceso si

hacemos que la Raspberry-pi actúe como servidor DNS y de esta forma restringir el acceso a ciertas páginas mediante el uso de listas negras de reputación.

Salvaguarda 6. Eliminar contenido de publicidad con filtrado de DNS.

Utilizando la técnica de usar la Raspberry-pi como servidor DNS, es posible que también se consiga evitar recibir el contenido de publicidad innecesaria.

Salvaguarda 7. Identificar dispositivos conectados asignado IP con DHCP.

Para tener un control de nuestros dispositivos, puede ser interesante implementar un sistema que siempre asignara la misma IP a los dispositivos de nuestra red. La mejor forma de hacerlo sin tener que ir configurando cada dispositivo es instalando un pequeño servidor DHCP en la Raspberry-pi, de forma que dependiendo de la MAC, se asigne una IP determinada a cada dispositivos.

Salvaguarda 8. Registrar el tiempo de actividad de los dispositivos.

Hay que estudiar la mejor solución para conocer el tiempo se actividad o conexión de los equipos para ser conscientes del uso real de internet, detectar excesos de uso que puedan desembocar en una tecno-adicción y prevenirla hablando con los hijos, marcando así unas pautas de uso responsable y adecuado.

En la fase de implementación se buscará la mejor solución que para poner en marcha esta salvaguarda. Pueden haber varias soluciones como puede ser el control mediante pings periódicos a cada dispositivo, poner una alarma en Suricata que detecte algún protocolo de conexión inicial de cada equipo, o un filtro del uso de llamadas DNS en los log de Suricata.

Salvaguarda 9. Restringir accedo a internet según horario a ciertos dispositivos.

Sería interesante que el sistema pudiera establecer horarios de uso de internet para los distintos dispositivos. Se podría estudiar la posibilidad de no dar servidor DNS a ciertos dispositivos según un horario establecido, valorando las posibilidades del servidor DNS para establecer ciertos horarios de servicio a ciertas IPs.

2.5.3. Tabla de análisis de riesgos.

Finalmente se muestra una tabla donde se ha valorado el riesgo de las diferentes amenazas detectadas. Para ello se ha valorado el posible daño causado en caso de que se manifieste la amenaza, y una estimación de la posibilidad de que se produzcan, calculando el riesgo en función de estas variables.

En la columna final se han identificado para cada amenaza, la salvaguarda que se podría implementar en este proyecto para minimizar el riesgo.

Amenazas	Daño (1-5)	Posibilidad (1-5)	Riesgo (DxP)	Salvaguardas con Suticata-pi
A1. Contenidos inapropiados	2	4	8	S1. Monitorización HTTP S2. Monitorización DNS S3. Alertas Ips malareputación S4. Alertas palabras prohibidas S5. Filtrado IPs por DNS S6. Filtrado publicidad por DNS
A2.Tecnoadicción	3	4	12	S7. Identificar dispositivos S8. Control tiempo de conexión S9. Restricción horaria
A3.Malware	2	3	6	S3. Alertas Ips malareputación S5. Filtrado IPs por DNS
A4.Contacto con desconocidos	2	3	6	S4. Alertas palabras prohibidas S8. Control tiempo de conexión S9. Restricción horaria
A5.Ciberbullying	4	2	8	S4. Alertas palabras prohibidas
A6.Grooming	5	1	5	S4. Alertas palabras prohibidas S8. Control tiempo de conexión S9. Restricción horaria
A7.Sexting	4	2	8	S4. Alertas palabras prohibidas S8. Control tiempo de conexión S9. Restricción horaria

3. Fase de implementación

3.1. Un IDS en una Raspberry-pi.

En este primer punto se va a describir toda la implantación se software realizada sobre una Raspberry-pi, desde la instalación del sistema operativo a la instalación de la sonda IDS Suricata así como el servicio DHCP y DNS con Pi-hole.

En el punto 3.1.2 también se explicará como preparar la infraestructura de red con la configuración de un Switch con funciones de Port-mirroring para que la sonda IDS reciba todo el tráfico de red.

3.1.1. Instalación de Raspberry-pi.

Una vez tenemos una Raspberry-pi hay que instalar el sistema operativo en una tarjeta SD. Para ello será suficiente con una tarjeta de 16Gb, pero es recomendable que sea de clase 10 para no penalizar en velocidad de acceso.

A continuación se define el proceso de la instalación del sistema operativo Raspbian en la Raspberry-pi:

3.1.1.1. Descargar una imagen del sistema operativo.

Para este proyecto se ha elegido la imagen de Raspbian Stretch Lite. La página oficial la podemos encontrar en la siguiente URL:

https://www.raspberrypi.org/downloads/raspbian/

Para no cargar la Raspberry-pi excesivamente con software innecesarios, se accederá al dispositivo mediante comandos del terminal. No se instalará un entorno de ventanas.

Se debe tener en cuenta que una vez el sistema está funcionando, solo tendrá conectado el cable de corriente y el cable de red.

Es por ello se que se ha elegido la la versión lite que viene con el software mínimo para funcionar y a partir de aquí instalar el software que realmente se sea necesario.

La URL del sistema operativo a descargar es la siguiente:

https://downloads.raspberrypi.org/raspbian_lite_latest.torrent

3.1.1.2. Grabar la imagen del sistema operativo en una tarjeta SD.

Una vez se tiene la imagen del sistema operativo descargada, hay que instalarla en una tarjeta SD que hará las funciones de disco duro en la Raspberry-pi.

Existen varias aplicaciones en internet que se pueden utilizar para realizar esta operación.

En este proyecto se han probado dos:

- Win32 Disk Imager [13]

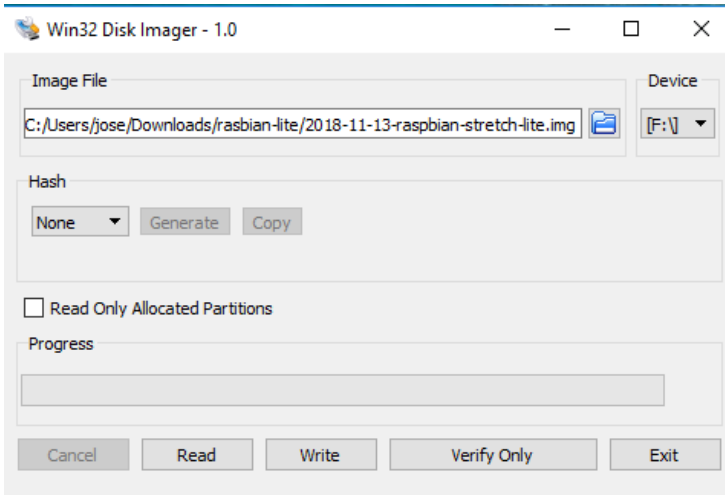


Ilustración 5: Win32 Disk Imager

- Balena Etcher [14]

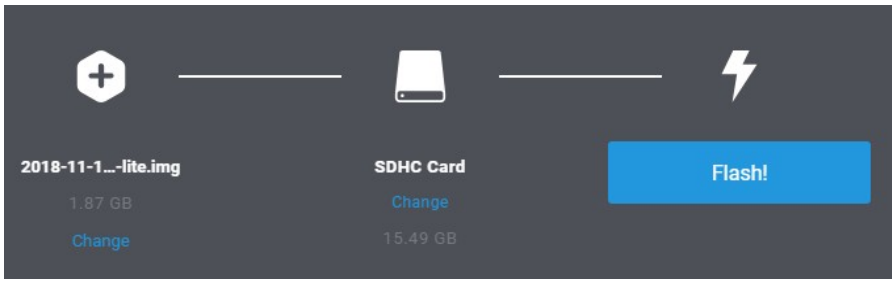


Ilustración 6: Balena Etcher

Cualquiera de las dos son perfectamente válidas. En ellas hay que seleccionar la imagen descargada, la unidad donde se ha montado la tarjeta SD y dar la orden de escribir o flashear la imagen.La operación puede durar unos minutos.

Una vez se tiene la imagen quemada en la tarjeta SD, ya se puede colocar en la Raspberry pi y arrancarla.

Para arrancar la Raspberry-pi, es necesario conectarla a una fuente de alimentación con conector micro-usb (el utilizado por los móviles tipo Android) que proporcione hasta 2Amperios .

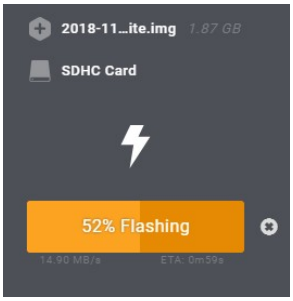


Ilustración 7: Flash SD en Balena

[13] Win32 Disk Imager https://sourceforge.net/projects/win32diskimager/

[14] Balena Etcher https://www.balena.io/etcher/

3.1.1.3. Acceso a la Raspberry-pi

En el primer arranque de la Raspberry-pi, se tiene que acceder con un teclado conectado al puerto USB y un monitor conectado a la salida HDMI.

Para acceder al sistema operativo se utiliza el usuario y clave por defecto:

```
Usuario: pi
Contraseña: Raspberry
```

Conviene cambiar la contraseña por defecto usando el comando:

```
$ passwd
```

3.1.1.4. Conectamos con la red wifi.

Una vez se ha accedido al sistema y se ha cambiado la contraseña, la primera tarea a realizar es conectarla a la red wifi para tener conexión a internet y de este modo poder actualizar e instalar programas.

Para configurar la Wlan desde el terminal se han de seguir el siguiente proceso:

Antes que nada se debe conocer la SIDD de la red Wifi a la que queremos conectarnos y la contraseña de acceso.

Editamos el fichero /etc/wpa_supplicant/wpa_supplicant.conf

Para ello, se puede utilizar el comando:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

En el fichero, se añaden los datos para conectarnos a nuestra red wifi:

```
network={
        ssid="nombre-de-tu-wifi"
        psk="password-de-tu-wifi"
        key_mgmt=WPA-PSK
```

Por lo que el fichero quedaría de la siguiente forma:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=ES

network={
        ssid="nombre-de-tu-wifi"
        psk="password-de-tu-wifi"
        key_mgmt=WPA-PSK
```

Una vez modificado el fichero, hay que reiniciar la Raspberry-pi con el comando:

```
sudo shutdown -r now
```

Cuando el sistema haya arrancado con *ifconfig wlan0* podemos comprobar que ya tenemos asignada una IP en nuestra red Wifi y tiene acceso a internet.

3.1.1.5. Actualizar el sistema operativo.

La siguiente tarea consiste en actualizar el sistema con las siguientes instrucciones:

```
sudo apt-get update
sudo apt-get upgrade
```

3.1.1.6. Castellanizar el dispositivo.

De momento el teclado de la Raspberry-pi está en ingles, con lo que es un poco costoso su manejo a la hora de escribir caracteres especiales.

En este punto se indica como ha castellanizar la Raspberry-pi para que sea más cómodo su manejo.

Para ello han seguido los pasos del siguiente tutorial:

https://wiki.bandaancha.st/C%C3%B3mo_espa%C3%B1olizar_tu_Raspberry_Pi

Podemos resumir el proceso en las siguientes acciones:

- Cambiar la codificación de idioma.

  #sudo dpkg-reconfigure locales.

  Seleccionar es_ES.UTF-8

- Seleccionar configuración del teclado.

  ```
  #dpkg-reconfigure keyboard-configuration
  ```

  Seleccionamos teclado Spanish

- Elegir la zona horaria:

  ```
  # dpkg-reconfigure tzdata
  ```

  Seleccionamos Europa/Madrid

3.1.1.7. Instalar servicio ssh

Se necesita un acceso remoto que permita conectarse desde otro ordenador a la Raspberry-pi. Hay que tener en cuenta que no va ha tener conectada ni monitor ni teclado.

Para dar este acceso remoto se instalará el servicio SSH.

Para instalar el servicio simplemente ejecutamos el siguiente comando:

```
# apt-get install ssh
# service ssh start
```

Luego se debe establecer que se arranque automático el servicio ssh en el inicio del sistema operativo. Esto lo podemos establecer ejecutando raspi-config y seleccionando las opciones siguientes:

```
#raspi-config
 -> 5 Interfacing Options -> P2 SSH -> Sí
```

3.1.1.8. Modificamos el nombre de la sonda IDS a "suricatapi".

Para ello hay que editar y modificar los los archivos hostname y hostsdel directorio /etc

```
root@raspberrypi:/home/pi# cat /etc/hostname
suricatapi
```

```
root@raspberrypi:/home/pi# cat /etc/hosts
127.0.0.1 localhost
::1    localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
127.0.1.1 suricatapi
```

3.1.1.9. Instalación del servicio FTP

Para poder descargar los ficheros Log desde otro equipo, es necesario instalar también el servicio de FTP de la siguiente manera:

```
root@suricatapi:/var/log/suricata# apt-get install ftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libfile-copy-recursive-perl openbsd-inetd update-inetd
Se instalarán los siguientes paquetes NUEVOS:
  ftpd libfile-copy-recursive-perl openbsd-inetd update-inetd
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 0 no
actualizados.
Se necesita descargar 124 kB de archivos.
Se utilizarán 357 kB de espacio de disco adicional después de esta
operación.
```

3.1.2. Configuración de la topología de red

En este apartado se valoran las diferentes topologías de red que se podrían utilizar sobre todo en función de utilizar Suricata en modo IDP o solo en modo IDS.

3.1.2.1. Topologia de red a configurar.

Se han realizado pruebas de utilizar la Raspberry pi como Gateway de forma que pueda actuar también como IPS.

Para ello se han seguido el tutorial que se muestra en la siguiente página web:

http://msrobotics.net/index.php/laboratorio-pi/227-configura-Raspberry-pi-como-punto-de-acceso-wifi

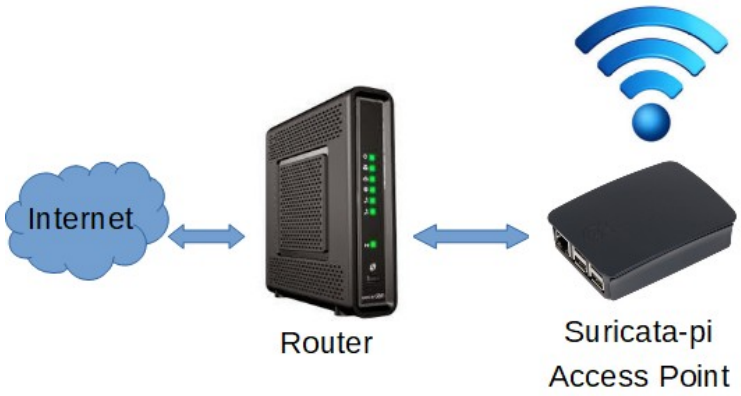


Ilustración 8: Raspberry-pi como AP

Conclusiones: Los resultados has sido una bajada considerable del rendimiento de red. La causa de esto es que todo el tráfico debe pasar por la interface Wlan de la Raspberry-pi y ser encaminado por la interface eth0 hacia el router de salida a internet, lo que genera un cuello de botella y su consecuente perdida de calidad del servicio de acceso a internet.

Esto implica que por necesidades rendimiento se decide utilizar Suricata solo con funciones de IDS que permita la monitorización del tráfico y la detección de eventos, pero sin capacidad para eliminar paquetes.

La topología final de dispositivos de red que se van a configurar sitúa a la sonda IDS (Raspberry-pi) no en medio del tráfico de red haciendo cuello de botella, sino a un lado. Para que todo funcione se debe utilizar un Switch que le pase todo el tráfico a la sonda con la función de Port-mirroing. Realmente es una topología más compleja ya que es necesaria la utilización más dispositivos de red a configurar, pero por otro lado, tiene la ventaja de no afectar al rendimiento de acceso a internet.

En la siguiente imagen muestra los componentes de red implicados así como las IPs asignadas:

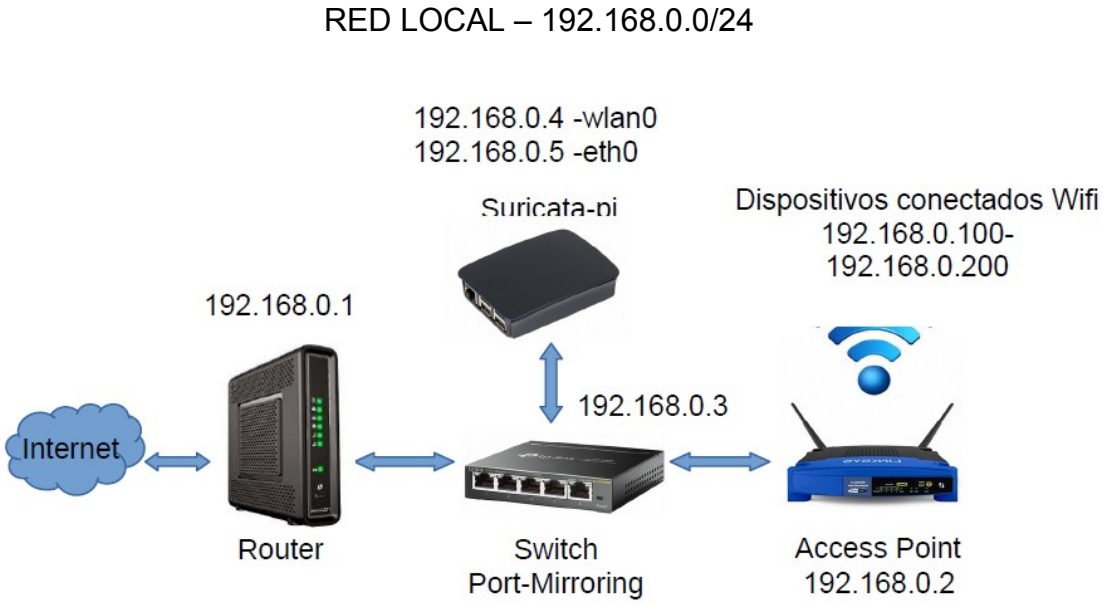


Ilustración 9: Dispositivos de red configurados

En los siguientes puntos se explica como configurar cada componente de red.

3.1.2.2. Configuración de red de la Raspberry-pi.

La Raspberry-pi tiene dos interfaces de red, una por cable RJ-45 y la otra vía Wifi.

Para el proyecto nos interesa establecer IPs estáticas a estas interfaces de red y a los dispositivos que componen la topología de red que se va a preparar.

- wlan0: 192.168.0.4
- eth0: 192.168.0.5

Para configurar eth0 editamos el fichero */etc/dhcpcd.conf* y descomentamos el ejemplo de IP configuration de eth0. En *ip_address* pondremos la IP que queremos asignar

```
# Example static IP configuration:
interface eth0
static ip_address=192.168.0.5/24
static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.0.1
static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

Para la red Wifi realizaremos algo similar añadiendo lo siguiente al fichero.

```
# Configuracion WLAN0
interface wlan0
static ip_address=192.168.0.4/24
static routers=192.168.0.1
static domain_name_servers=192.168.0.1
```

Ahora ya se puede conectar la Raspberry-pi al switch y reiniciar. Para que se levante la interface de red eth0, debe estar conectada a la red por cable a un switch de la red.

```
$ sudo reboot
```

```
root@suricatapi:/home/pi# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
                inet  192.168.0.5    netmask  255.255.255.0    broadcast
192.168.0.255
root@suricatapi:/home/pi# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
                inet  192.168.0.4    netmask  255.255.255.0    broadcast
192.168.0.255
```

3.1.2.3. Configuración del Switch con port-mirroring.

Con el objetivo que la Raspberry-pi reciba todos los paquetes de red que salen a internet y pueda actuar como IDS, se deben cumplir dos condiciones:

A. Configurar la boca de red en modo promiscuo.

Esto hará que la interface de red (eth0) acepte todos los paquetes de red, que tanto si son dirigidos a esta interface como si no.

B. Utilizar un Switch con port-mirroring.

Los switch tiene la capacidad de detectar que dispositivos están conectado a cada boca del switch y enviar a cada boca los paquetes de red que van dirigidos a cada dispositivo.

Es por ello que se necesita un switch con capacidad de Port-mirroring. Esta funcionalidad permite reenviar todos los paquetes que entran por una boca de switch (a la salida a internet), a otra donde estará conectada la raspberry-pi.

En este proyecto se ha utilizado el **Switch TL-sg105e** por contemplar esta funcionalidad y su bajo coste.

Ilustración 10: Switch TL-sg105e

Para la configuración del Switch se ha seguido el siguiente proceso:

1. Descargar e instalar utilidad Unmanaged Pro Configuration Utility

https://www.tp-link.com/us/download/TL-SG105E_V1.html#Easy_Smart_Configuration_Utility

2. La herramienta detecta el switch permitiendo entrar con la contraseña de fábrica y configurar el dispositivo según nuestras necesidades.

3. Cambio de contraseña

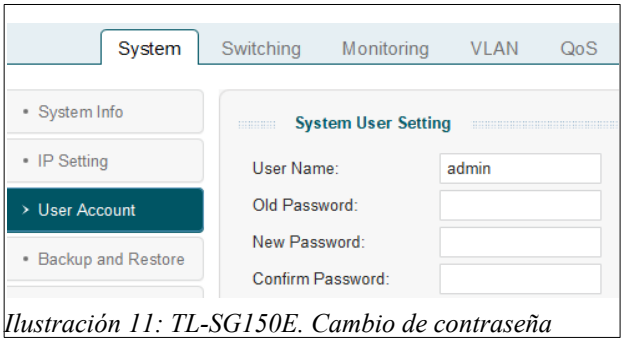


Ilustración 11: TL-SG150E. Cambio de contraseña

4. Se cambia la IP estática.

Ilustración 12: TL-SG150E. Asignación de IP

5. Establecer la función de port-mirroring.

Para ello primero se definen en que boca se va a conectar cada dispositivo:

Puerto	Dispositivo	Observación
Puerto 1	Router Wifi- Linksys	Todo el tráfico que llega a este puerto lo reenviaremos al Puerto2
Puerto 2	Raspberry-pi con sonda IDS	Recibe todo el tráfico del Puerto1
Puerto 5	Router de salida a Internet	Salida a internet

Según esta planificación de puertos configuramos el Port-mirroring en el Switch para que todo el tráfico de internet que pasa por el puerto 1 lo reenvíe a la sonda IDS que está conectada al puerto2.

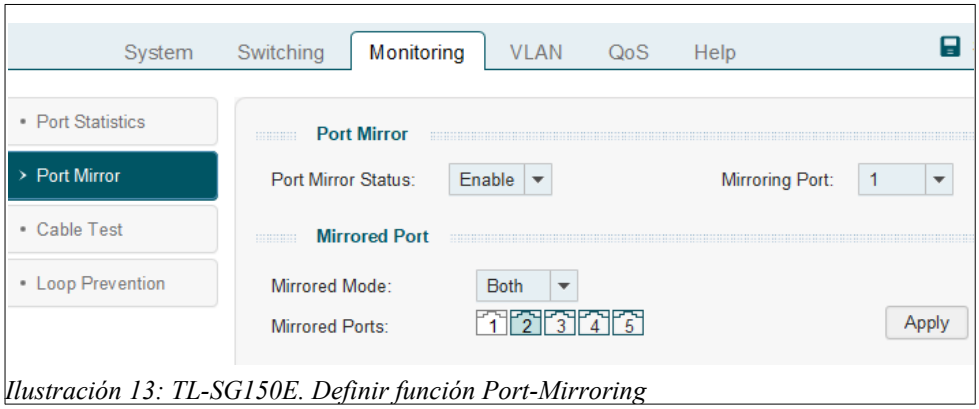


Ilustración 13: TL-SG150E. Definir función Port-Mirroring

3.1.2.4. Configuración del Access Point Wifi.

Para la conexión de los dispositivos a internet por wifi, se ha utilizado un router LINKSYS.

Ilustración 14: Router Linksys WRT54GL

Solución del problema sobre la identifican los dispositivos.

Planteamiento del problema encontrado:

En una primera configuración del router wifi, se le asignó una IP que estaba en el rango 192.168.1.0/24 y por lo tanto su función era encaminar el tráfico a la red 192.168.0.0/24 donde se encontraba el router de internet y la IDS.

Esto suponía que en los paquetes que llegaban al IDS no se identificaba las IPs de origen de los dispositivos, ya que todo se enmascaraba con la ip del router Wifi.

Para resolver esto se tuvo que configurar el router Wifi para que funcionara en modo Router *en lugar de modo* **Gateway** y asignar a todos los dispositivos IPs de la misma red 192.168.0.0/24. Esto se realiza en el apartado de Advanced Routing del Setup.

Ilustración 15: Linksys WRT54GL en modo Router

Configurar la IP 192.168.0.2 como IP estática del Router wifi:

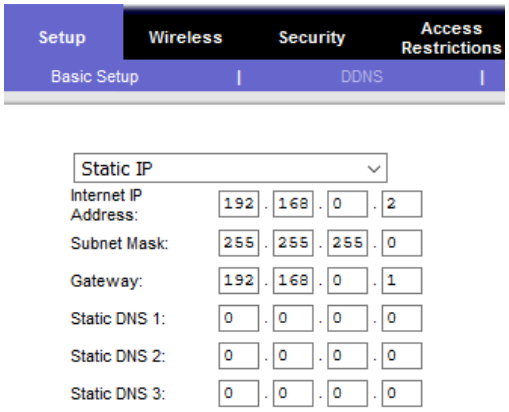


Ilustración 16: Linksys. Confiruración de IP

3.1.3. Instalación de IDS Suricata en Raspberry-pi

En los siguientes puntos se exponen los pasos seguidos para la instalación de software Suricata en la Raspberry-pi, así como algunas nociones sobre la configuración de esta.

El proceso de instalación del IDS Suricata ha sido el siguiente:

3.1.3.1. Actualización del sistemas

Antes de descargar los paquetes a instalar conviene tener el sistema actualizado.

```
# apt-get update
# apt-get upgrade
```

3.1.3.2. Instalar los paquetes de Suricata del repositorio de raspbian

La instalación de Suricata se realiza con un simple comando apt-get.

```
# apt-get install suricata
Leyendo lista de paquetes... Hecho
...
```

3.1.3.3. Comprobamos que está el servicio en ejecución.

Con el comando service podemos comprobar que el servicio está bien instalado. Si da algún error se podrá observar en el fichero */var/log/suricata/suricata.log*

```
#  service suricata status
● suricata.service - Suricata IDS/IDP daemon
     Loaded: loaded (/lib/systemd/system/suricata.service; disabled;
vendor preset: enabled)
    Active: active (running) since Fri 2019-04-19 13:12:27 CEST; 57s
ago
     Docs: man:suricata(8)
           man:suricatasc(8)
            https://redmine.openinfosecfoundation.org/projects/suricata
/wiki
 Main PID: 7258 (Suricata-Main)
   CGroup: /system.slice/suricata.service
                        └─7258 /usr/bin/suricata -D --af-packet -c
/etc/suricata/suricata.yaml --pidfile /var/run/suricata.p

abr 19 13:12:27 suricatapi systemd[1]: Starting Suricata IDS/IDP
daemon...
abr 19 13:12:27 suricatapi suricata[7257]: 19/4/2019 -- 13:12:27 -
<Notice> - This is Suricata version 3.2.1 RE
abr 19 13:12:27 suricatapi systemd[1]: Started Suricata IDS/IDP
daemon.
```

3.1.3.4. Comprobación de Logs de suricata.

Los Logs de suricata se guardan en el directorio /*var/*log/suricata

Comprobamos que el sistema esta recogiendo información.

```
root@suricatapi:/var/log/suricata# ls -l
total 260
-rw-r----- 1 root root 174111 abr 19 13:16 eve.json
-rw-r----- 1 root root      0 abr 19 13:12 fast.log
-rw-r----- 1 root root  79767 abr 19 13:16 stats.log
-rw-r--r-- 1 root root   5003 abr 19 13:12 suricata.log
```

Comprobamos la ip de un dispositivo conectado a la red. En concreto el 192.168.0.32

Se realiza la comprobación de que sí se recogen eventos en el fichero **eve.json** mientras navego por ese dispositivo.

```
# tail -f eve.json | grep 192.168.0.32
{"timestamp":"2019-04-
19T13:37:14.268640+0200","flow_id":1838944789073051,"in_iface":"eth0",
"event_type":"tls","src_ip":"192.168.0.32","src_port":57118,"dest_ip":
"204.16.244.13","dest_port":443,"proto":"TCP","tls":{"subject":"C=US,
unknown=15213, ST=PA, L=Pittsburgh, unknown=5001 Baum Blvd STE 770,
O=Liberated Syndication, OU=PremiumSSL Wildcard,
CN=*.libsyn.com","issuerdn":"C=GB, ST=Greater Manchester, L=Salford,
O=COMODO CA Limited, CN=COMODO ECC Organization Validation Secure
Server
CA","fingerprint":"94:01:38:8d:1b:f9:da:5f:6b:93:2b:80:00:db:66:1b:94:
ca:f3:bf","sni":"cesargarciasaez.libsyn.com","version":"TLS
1.2","notbefore":"2018-07-09T00:00:00","notafter":"2020-08-
05T23:59:59"}}

{"timestamp":"2019-04-
19T13:37:14.970273+0200","flow_id":1340698518015521,"in_iface":"eth0",
"event_type":"dns","src_ip":"192.168.0.32","src_port":52404,"dest_ip":
"62.81.16.164","dest_port":53,"proto":"UDP","dns":
{"type":"query","id":32421,"rrname":"www.sueldo30.com","rrtype":"A","t
x_id":0}}

{"timestamp":"2019-04-
19T13:37:15.040828+0200","flow_id":1340698518015521,"in_iface":"eth0",
"event_type":"dns","src_ip":"62.81.16.164","src_port":53,"dest_ip":"19
2.168.0.32","dest_port":52404,"proto":"UDP","dns":
{"type":"answer","id":32421,"rcode":"SERVFAIL","rrname":"www.sueldo30.
com"}}

{"timestamp":"2019-04-
19T13:37:15.048577+0200","flow_id":2003721209494977,"in_iface":"eth0",
"event_type":"dns","src_ip":"192.168.0.32","src_port":52404,"dest_ip":
"62.81.16.213","dest_port":53,"proto":"UDP","dns":
{"type":"query","id":32421,"rrname":"www.sueldo30.com","rrtype":"A","t
x_id":0}}
```

3.1.3.5. Actualización de reglas

Utilizaremos el comando **suricata-oinkmaster-updater** para realizar esta actualización de reglas del repositorio *https://rules.emergingthreats.net/*. Con esto preparamos nuestro sistema para que lance una alarma cuando detecte alguna amenazas identificada por las reglas.

```
# suricata-oinkmaster-updater
Loading /etc/suricata/suricata-oinkmaster.conf
```

```
Downloading file from https://rules.emergingthreats.net/open/suricata-
3.0/emerging.rules.tar.gz... done.
Archive successfully downloaded, unpacking... done.
Setting up rules structures... done.
Processing downloaded rules... disablesid 0, enablesid 0, modifysid 0,
localsid 0, total rules 27031
```

3.1.3.6. Configuración del rotado de ficheros logs.

Nos interesa manejar la configuración de rotado de los ficheros **even.json** para controlar la transferencia de estos ficheros a ELK.

En la versión 3,2,1 que viene instalada en repositorio de la Raspberry-pi se puede utilizar logrotate para organizar la gestión de logs.

Logrotate es una herramienta que nos permitirá configurar el rotado, la compresión de logs del sistema.

El fichero de configuración es /etc/logrotate.conf

Luego en el directorio /etc/logrotate.d podemos encontrar configuraciones particulares para cada servicio.

Para Suricata existe un fichero con este mismo nombre con la configuración predeterminada, tal y como indica en el manual[15].

Se modifica el fichero /etc/logrotate.d/suricata configurando el sistema de rotación de ficheros, para el fichero **eve.json** con una rotación diaria.

```
/var/log/suricata/*.json
{
    daily
    rotate 15
    missingok
    nocompress
    dateext
    dateformat -%d%m%Y-%H:%M
    create
    sharedscripts
    postrotate
            /bin/kill -HUP $(cat /var/run/suricata.pid)
    endscript
}
```

Una vez que hayamos terminado con la configuración en un archivo dado, podemos chequearla de la siguiente manera:

```
# logrotate -d /etc/logrotate.conf
```

[15] Manual Suricata:*https://suricata.readthedocs.io/en/suricata-3.2.1/configuration/log-rotation.html?highlight=logrotate*

```
rotating pattern: /var/log/suricata/*.json
 after 1 days (15 rotations)
empty log files are rotated, old logs are removed
considering log /var/log/suricata/eve.json
  Now: 2019-04-19 17:33
  Last rotated at 2019-04-19 17:07
  log does not need rotating (log has been already rotated)
not running postrotate script, since no logs were rotated
```

Podemos Forzar la ejecución con la siguiente instrucción:

```
# logrotate -vf /etc/logrotate.conf
```

3.1.3.7. Iniciar el servicio de Suricata con el arranque del sistema.

Para ello consultamos si esta establecido, y comprobamos que es *disabled.*

```
# systemctl is-enabled suricata
disabled
```

Se habilita inicio del servicio al arranque del sistema con la instrucción "**systemctl enable nombredelservicio"**

```
# systemctl enable suricata
Synchronizing state of suricata.service with SysV service script
with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable suricata
# systemctl is-enabled suricata
enabled
```

Se reinicia el sistema y comprobamos que está todo correctamente arrancado.

3.1.3.8. Configuración de Suricata.

Una vez tenemos el IDS Suricata instalado en la Raspberry-pi podemos realizar algunos ajustes en su configuración para adaptarlo a nuestras necesidades.

Los ficheros de configuración de Suricata están en ***/etc/suricata***.

Fichero de configuración principal *suricata.yaml*

En este fichero es donde se definen las variables de configuración.

A continuación de destacan algunas de las principales variables que se han modificado:

```
 HOME_NET: "[192.168.0.0/24]"                # Define la red local
default-rule-path: /etc/suricata/rules             # Directorio de las
reglas
rule-files:
- parental-control.rules                  # Fichero de reglas de
                                Control parental
..

default-log-dir: /var/log/suricata/             # Directorio de Logs

  - eve-log:                         # Configuración eve.json
      enabled: yes
      filetype: regular
      filename: eve.json
     types:
        - alert:                        # Tipos de registros
            http: yes                  # enable dumping of http fields
            tls: yes                   # enable dumping of tls fields
            ssh: yes                   # enable dumping of ssh fields
            smtp: yes                  # enable dumping of smtp fields
            dnp3: yes                  # enable dumping of DNP3 fields
```

Definición de reglas personalizadas.

Se crea un fichero llamado **parental-control.rules** para definir reglas personalizadas orientadas a la detección de eventos interesantes para el control parental que luego puedan se monitorizados.

Con estas reglas se puede definir formas de detectar el uso de una determinada aplicación como puede ser Whataapp. Para ello hay que analizar los puertos o servidores que utiliza la aplicación y según esto disparar las alertas para posteriormente ser analizadas.

Ejemplo: Para lanzar una alarma cuando se utilice uno de los puertos utilizados por WhatsApp (el 5222) se podría definir esta regla.

```
alert tcp any any -> any 5222 (msg:"CONEXION WhatsApp!!";
flow:established,to_server; sid:1000001; rev:1;)
```

3.1.4. Instalación de Pi-hole.

Ante la imposibilidad de utilizar Suricata como IPS, se ha investigado otras posibilidades que permitan alcanzar los objetivos respecto a la denegación de acceso a sitios web peligroso o con contenido no apropiado para menores.

La solución propuesta es controlar el servidor de nombres de dominio (DNS) no suministrando la IP de los nombres de dominios que estén clasificados como peligrosos. Esto no afectaría a la velocidad ya que solo interviene en el momento de solicitar la IP asociada a un nombre de dominio.

Para implementar esta solución se propone utilizar Pi-hole[16], el cual se ha incluido como parte de este proyecto para cumplir con los objetivos marcados, ofreciendo la funcionalidad de bloquear el acceso a dominios incluidos en listas negras.

Además del control de servicio DNS, en Pi-hole también se activará el servicio de DHCP, de manera que se tenga un control por MAC de la asignación de direcciones IP en la red local, lo que permitirá identificar de forma clara los dispositivos conectados, así como detectar con facilidad si se ha conectado algún dispositivo intruso que no tenemos inventariado en nuestra red.

En la web del proyecto se destacan de Pi-hole las siguientes características :

- Fácil de instalar.
- Bloquea contenido sin configurar los navegadores de los clientes.
- Mejora el rendimiento de navegación al utilizar una cache de DNS.
- Es ligero para poder funcionar en una Raspberry-pi.
- Utiliza una interface gráfica que facilita la consultas.
- Se puede activar el servicio DHCP para controlar mejor a los dispositivos conectados a la red.
- Bloquea anuncios tanto en IP4 como en IP6
- Es software libre.

3.1.4.1. Proceso de instalación de Pi-hole.

El proceso de instalación es muy sencillo.

Antes de proceder a la instalación, se ejecuta un asistente que va informando y va realizando una serie de preguntas:

[16] Ref: https://github.com/pi-hole/pi-hole

- Descarga y ejecutamos el asistente para la instalación.

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

- Nos indica que se van ha tener que utilizar IP estáticas.

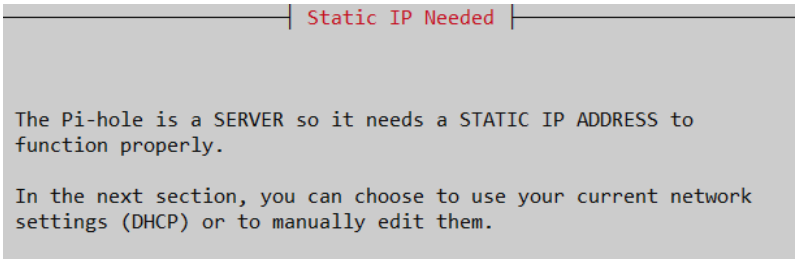


Ilustración 17: Instalar Pi-hole. IP estatica

- Seleccionamos la interface Eth0 para ofrecer el servicio.

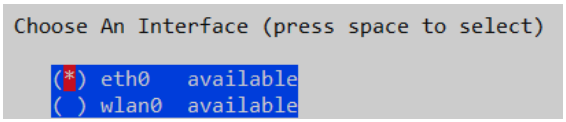


Ilustración 18: Instalar Pi-hole. Seleción de interface

- Elegimos OpenDNS como DNS por defecto donde redireccionar las consultas.

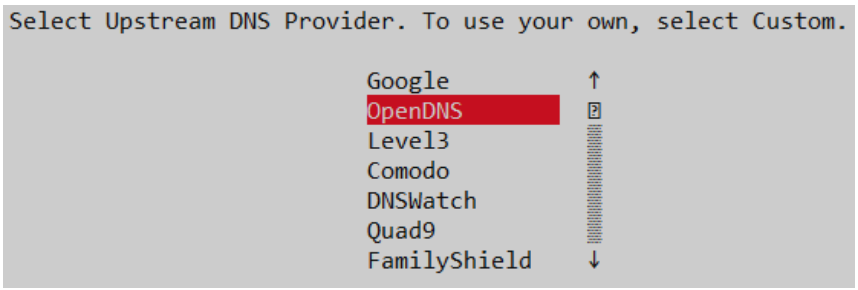


Ilustración 19: Instalar Pi-hole. Selección DNS

- Seleccionamos las listas de anuncios y malwares que consultará Pi-hole para realizar los bloqueos.

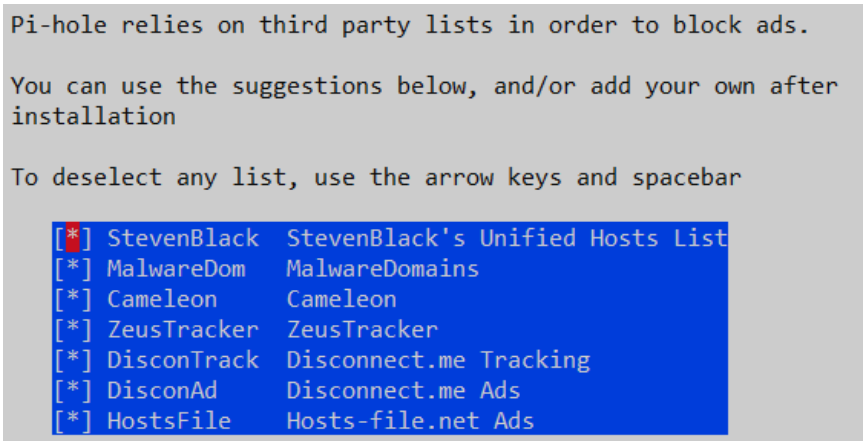


Ilustración 20: Instalar Pi-hole. Seleeción de listas

- Seleccionamos si actuará sobre Ip4 e Ip6.

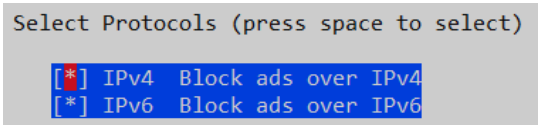


Ilustración 21: Instalar Pi-hole. Selección protocolos

- Nos informa de la IP que tenemos asignada en Eth0

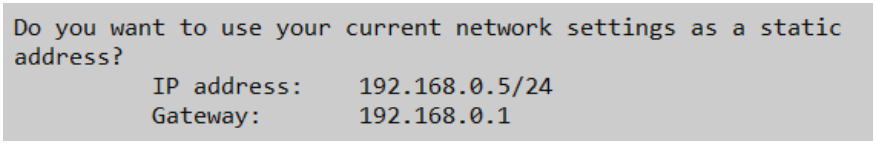


Ilustración 22: Instalar Pi-hole. Confirmación de IP

- Pregunta si se desea instalar la interface web.

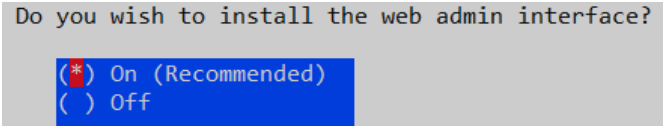


Ilustración 23: Instalar Pi-hole. Elegir instalar interface Web

- Pregunta si se desea instalar el servidor web ligero (lighttpd).

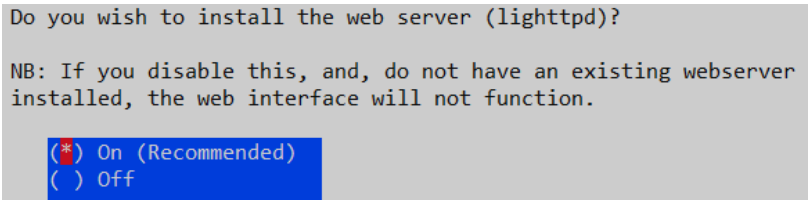


Ilustración 24: Instalar Pi-hole. Elegir servidor web

- Pregunta si se quiere tener log de las consultas DNS.

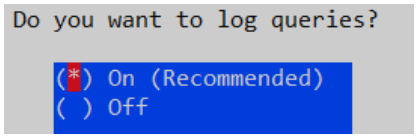


Ilustración 25: Instalar Pi-hole. Log de DNS

- Permite seleccionar el modo de privacidad de las consultas.

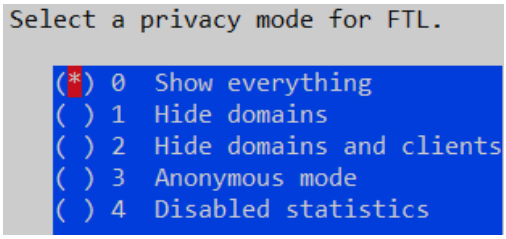


Ilustración 26: Instalar Pi-hole. Elegir nivel de privacidad

- Informa que la instalación está completa y da la contraseña de acceso.

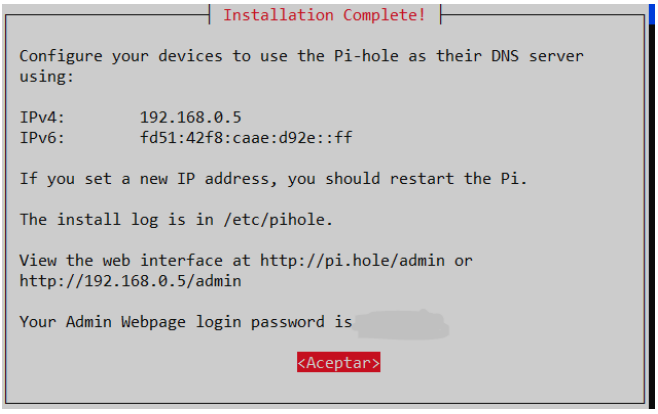


Ilustración 27: Instalar Pi-hole. Instalación completa

3.1.4.2. Establecer Pi-hole como servidor DHCP.

En este proyecto nos interesa que nuestra Raspberry-pi sea el servidor DHCP y esto se puede hacer en la configuración de Pi-hole.

Una vez Pi-hole sea nuestro servidor DHCP se tendrá el control de asignar IP concretas a ciertos dispositivos según su MAC y de está forma tenerlos más controlados.

Para configurar a Pi-hole como el servidor de DHCP de la red local, desde la interface web, hay que entrar en Configuración DHCP Setting y activar el servicio.

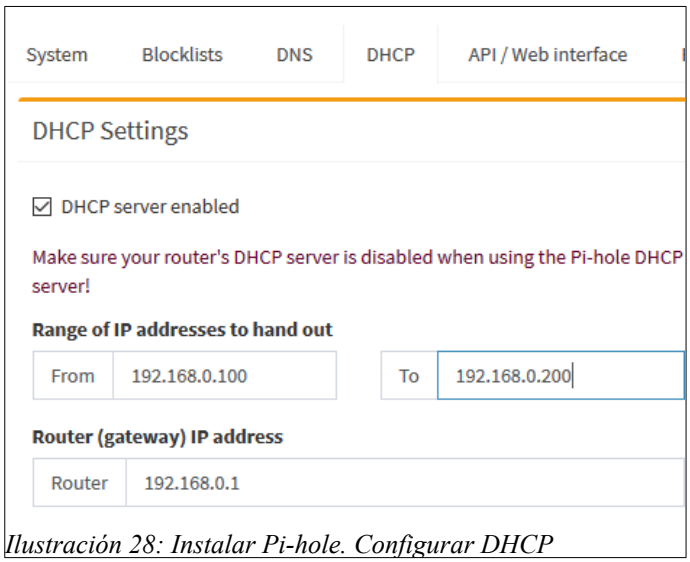
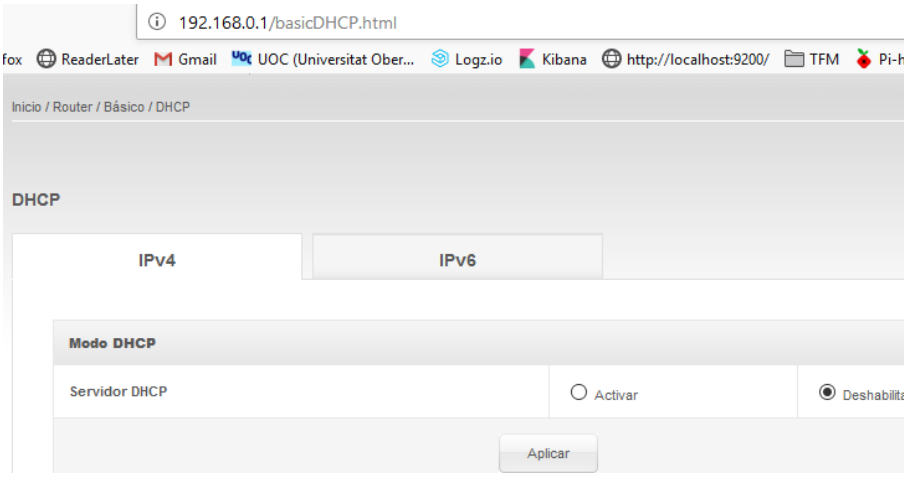


Ilustración 28: Instalar Pi-hole. Configurar DHCP

Nos avisa, que se debe desactivar el servicio de DHCP del router de internet para evitar conflictos a la hora de asignar las IPs a los dispositivos.

Así pues, accedemos al Router 192.168.0.1 y deshabilitamos su servicio DHCP.



Se hace la prueba de conectarse desde un dispositivo forzando la asignación de una nueva IP y se comprueba que ha sido asignada por la Raspberry-pi.

3.1.4.3. Dashboard de Pi-hole.

Una vez tenemos Pi-hole funcionando y configurado, se puede acceder al Dashboard con la contraseña de acceso.

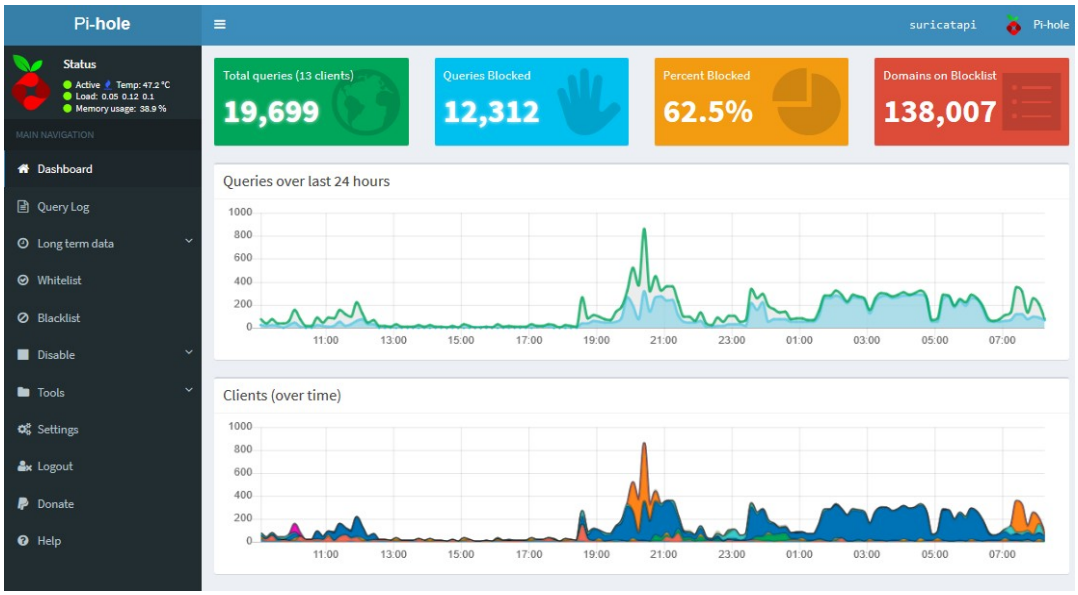


Ilustración 30: Pi-hole. Dashboard principal

En el panel accedemos de una vista a gráficas temporales de accesos así como otras tipo tarta con los principales dominios permitidos y bloqueados.

Por otro lado, es posible desde el menú de la derecha acceder a los logs de acceso y filtrar por dispositivos.

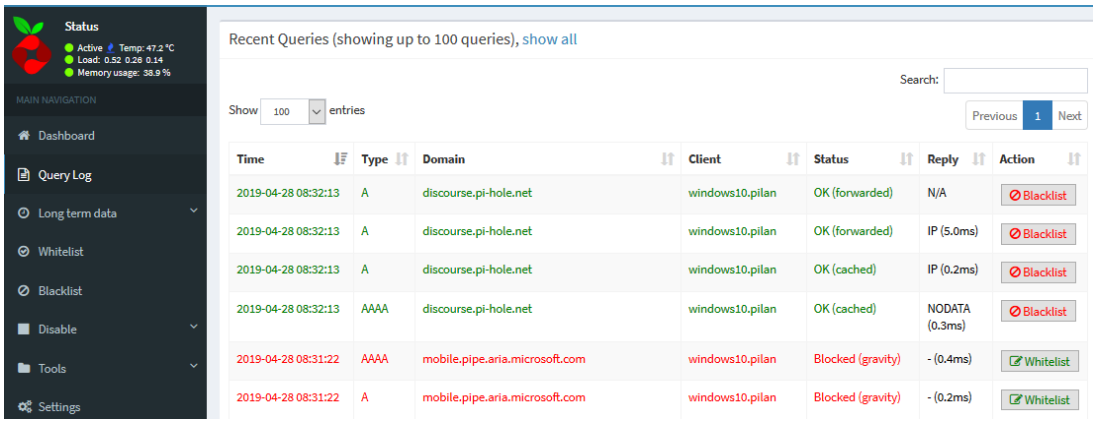


Ilustración 31: Pi-Hole. Query Log

También podemos añadir y parametrizar las listas negras y blancas para ajustar los bloqueos a las necesidades del administrador.

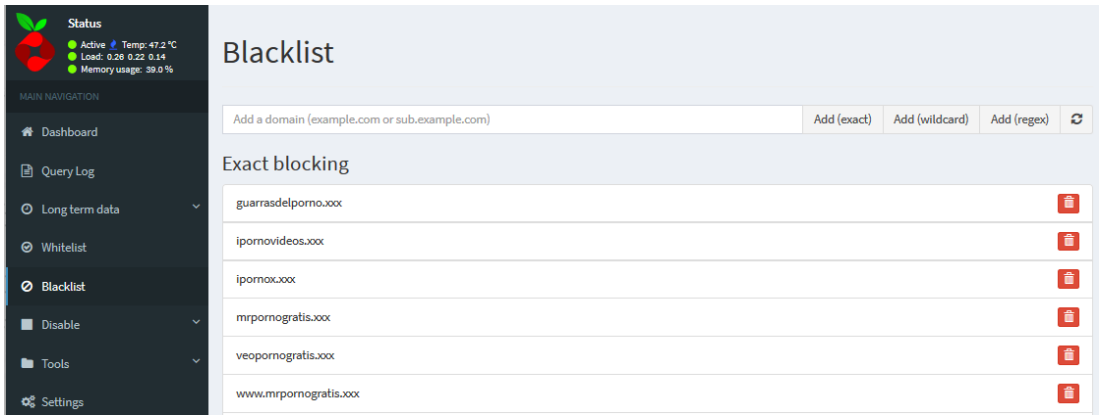


Ilustración 32: Pi-Hole. BlackList

3.2. Implantación de SIEM en el ámbito familiar.

En esta fase del proyecto se va a analizar la posibilidad de implantar un SIEM que permita de una forma visual analizar los eventos detectados por la Sonda IDS que se ha instalado en la Raspberry-pi.

Las herramientas utilizadas para realizar esta implementación son la Pila ELK de Elastic que incluye **Logstash** para el manejo de log, **Elastic search**[17] como base de datos y motor de búsqueda y **Kibana** como visualizador de datos.

https://www.elastic.co/es

En el apartado de análisis de problemas encontrados en la implantación se explicará por qué se instala la pila ELK en un equipo Windows el cual deberá de recoger los Logs de nuestra Sonda IDS. Algunas de las razones que hacen adoptar esta solución son las siguientes:

- Se considera que instalar toda la pila ELK en la Raspberry pi, puede afectar en el rendimiento del sistema teniendo en cuenta que es un dispositivo poco potente que ya está manejando el IDS de Suricata.
- Se ha probado a instalar solo el componente de manejo de Log en la Raspberry-pi, pero además de los problemas con los paquetes ARM, tendríamos que tener siempre en marcha el servidor ElasticSearch para recoger los datos y no se ha visto operativo.
- En un entorno familiar, Windows es el S.O más extendido con lo que se ha descartado la instalación en un sistema GNU/Linux a pesar de ser una plataforma de instalación más natural para estas herramientas.
- Se considera que el equipo Windows no va ha estar siempre en marcha como un servidor. Por lo tanto la funcionalidad de sincronización de LOGs que se pretende implantar es que cuando arranque el Windows recoja de la Sonda IDS los Logs pendientes y los incorpore a la base de datos de Elastic Serach. Para ello habrá que preparar algún tipo de Script que realice esta tarea. Se utilizará el protocolo FTP para la transferencia de ficheros, pero también se han realizado pruebas con otros sistemas como SAMBA.

[17] https://www.elastic.co/es

3.2.1. Instalación de la Pila Elastic.

A continuación se expone el proceso seguido durante la instalación de los componentes principales de la pila Elastic: Elasticsearch, Logstash y Kibana. Todos ellos instalados en un equipo doméstico con Windows 10 home edition.

3.2.1.1. Instalación de Elastic Search en Windows con un paquete MSI

Para la instalación de ElasticSearch en Windows se han seguido los siguientes puntos:

- Bajamos el paquete .msi para Elasticsearch de la siguiente URL.

  https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.0.0.msi

- Ejecutamos el fichero descargado y seguimos los pasos.
- Configuración de directorios de instalación.

  Preparamos los directorios a partir de la raíz C:\ELK donde instalaremos todos los productos de la pila.

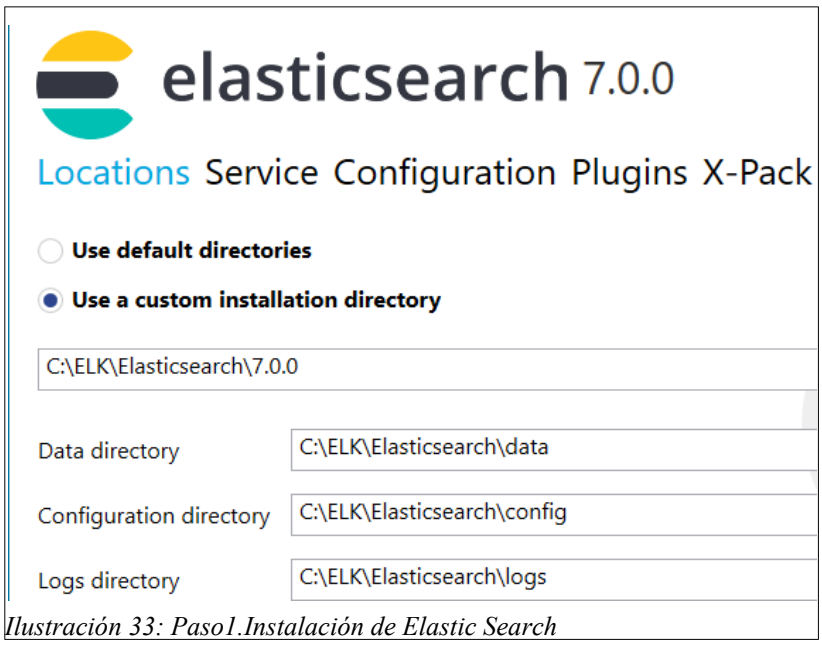


Ilustración 33: Paso1.Instalación de Elastic Search

- Seleccionamos las instalación como un servicio para que este arranque cuando arranque el sistema operativo.

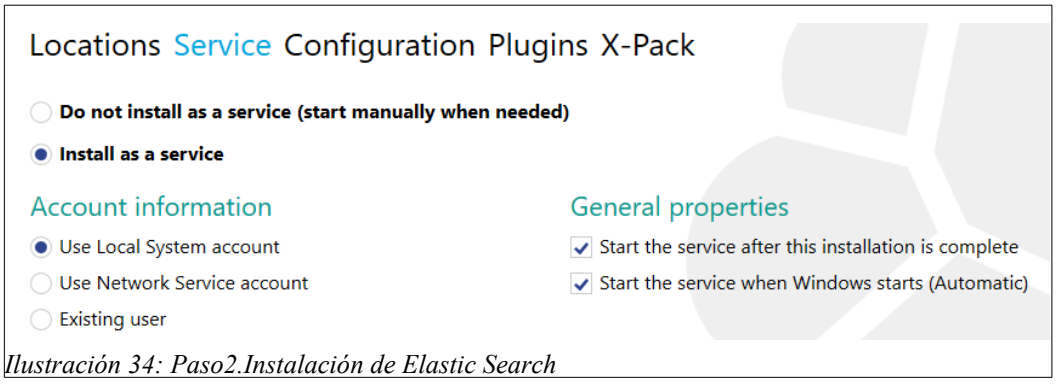


Ilustración 34: Paso2.Instalación de Elastic Search

- Permite configurar algunas características como identificadores, valores de me memoria, puertos de red. En este proyecto se dejan los valores por defecto.

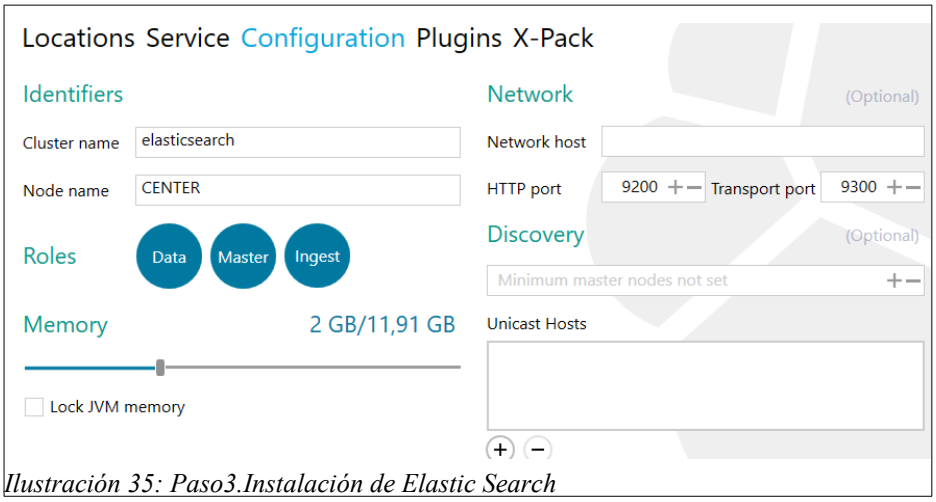


Ilustración 35: Paso3.Instalación de Elastic Search

- Permite seleccionar plugins a añadir a la instalación. De momento no se selecciona ninguno.

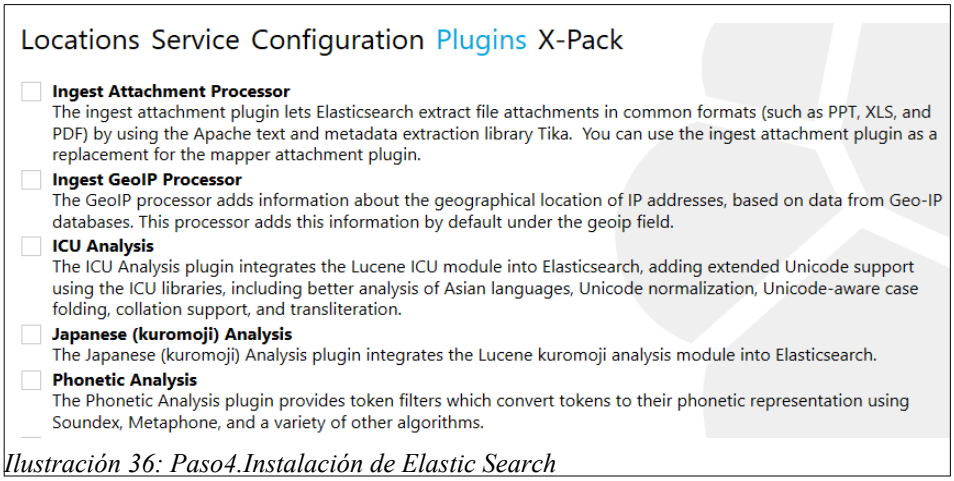


Ilustración 36: Paso4.Instalación de Elastic Search

- Se selecciona el tipo de licencia básica.

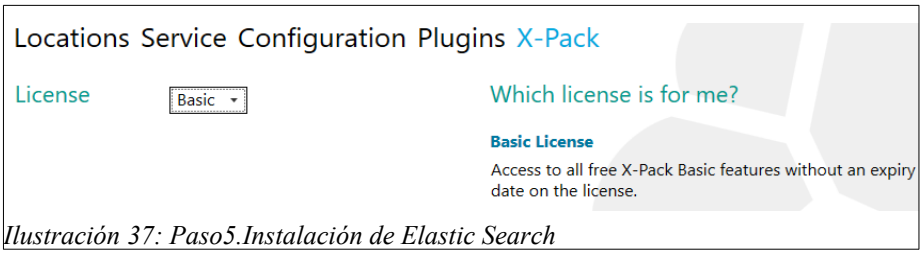


Ilustración 37: Paso5.Instalación de Elastic Search

- Se inicia la instalación hasta que da un aviso de instalación correcta.

Ilustración 38: Instalación de Elastic Search finalidada

- Se comprueba que el servicio de Elastic Search esta en marcha con el navegador.

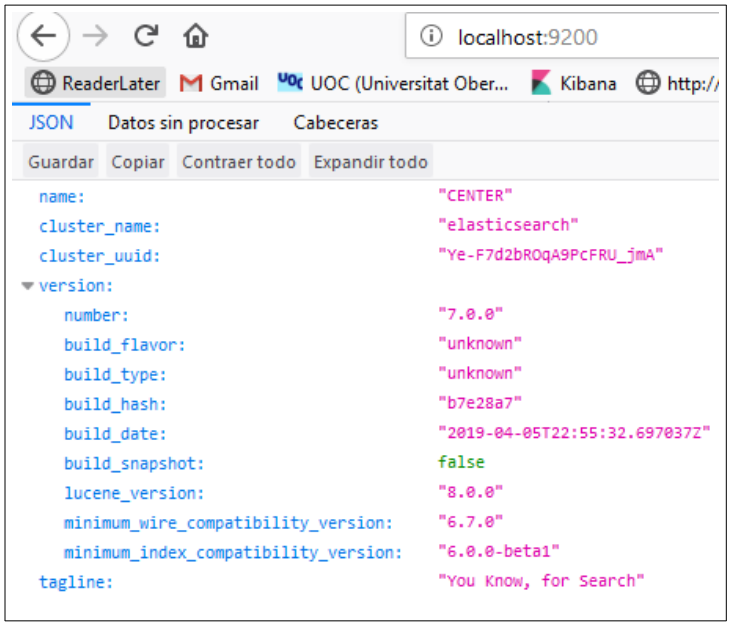


Ilustración 39: Servidor Elastic Search

3.2.1.2.Instalación de Kibana

En el proceso de la instalación de Kibana en Windows, se han seguido los siguientes puntos:

- Descargar la última versión de Kibana para Windows de la siguiente URL.

  https://artifacts.elastic.co/downloads/kibana/kibana-7.0.0-windows-x86_64.zip

- Descomprimir el fichero en la unidad que hemos preparado para instalar todas las herramientas ELK (C:\ELK\kibana).

- Abrir el fichero de configuración de Kibana (config/kibana.yml)para indicar que apunte a nuestro nodo de Elastic Search.

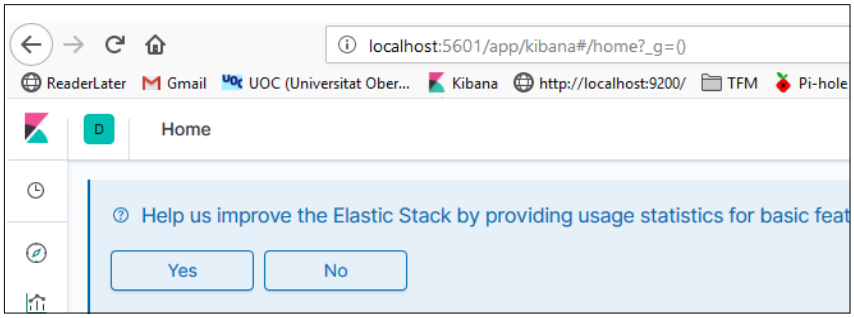
  Esto establece en la variable `elasticsearch.hosts`

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

- Para ejecutar Kibana se debe ejecutar `C:\ELK\kibana\bin\kibana.bat` en Windows.

```
C:\ELK\kibana\bin>kibana.bat
  log   [08:38:40.893] [info][status][plugin:kibana@undefined]
Status changed from uninitialized to green - Ready
...
  log   [08:38:44.881] [info][migrations] Finished in 174ms.
  log   [08:38:44.885] [info][listening] Server running at
http://localhost:5601
  log   [08:38:44.939] [info][status][plugin:spaces@7.0.0]
Status changed from yellow to green - Ready
```

- En el navegador se comprueba que se tiene acceso a Kibana con la URL:



3.2.1.3. Instalación de Logstash

Para realizar la instalación de **Logstash** en Windows, se deben seguir los siguiente pasos:

- Descargar la última versión de Kibana para Windows de la siguiente URL.

  https://artifacts.elastic.co/downloads/logstash/logstash-7.0.0.zip

- Descomprimir el fichero en la unidad que hemos preparado para instalar todas las herramientas ELK (C:\ELK\logstash).

- Preparar un fichero de configuración para la carga de datos.

  Para realizar la prueba se prepara de fichero de configuración provisional que cargue datos de un fichero **eve.json** creado por Suricata. Los datos los mostrará por pantalla y los enviará a Elastic search. Más adelante se explicará el formato eve.json, así como el fichero de configuración definitivo.

```
input {
  file {
    path =>  ["c:/elk/ftp/eve.json"]
    codec =>   json
    mode => ["read"]
    type => "SuricataIDPS"
    start_position => "beginning"
    }
}
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "suricata-%{+YYYY.MM.dd}"
    document_type => "suricata"
  }
}
```

- Para probar a ejecutar y ver si carga los datos.

  C:\ELK\logstash>bin\logstash.bat -f eve.conf

3.2.2. Automatización del proceso de transferencia de datos.

Para realizar la carga de datos de la sonda IDS y poder visualizarlos en Kibana se ha de establecer unos procesos de transferencia de ficheros.

Durante el diseño del sistema de carga de datos han aparecido una serie de problemas que han afectado en la planificación realizada y que se explicarán en el apartado de problemas encontrados durante la implementación.

Las solución elegida para el diseño del proceso de carga de datos tiene que cumplir con los siguientes requisitos:

- Los datos se generan en la Sonda IDS donde está instalado Suricata y Pi-hole en sus respectivos ficheros log.
- Los datos tiene que ser capturados por el nodo de ElasticSearch.
- El nodo ElasticSearch está instalado en un equipo Windows que no siempre está encendido.
- No se puede tener montado un sistema de transferencia continua (en modo Tail del file input de Logstash) ya que Elastic Search no está siempre disponible para recibir los datos generados.

Finalmente, la solución adoptada para realizar esta transferencia consiste en:

- Utilizar le modo “Read” del file input en Logstash de ficheros para leer ficheros desde el inicio y luego borrarlos.
- Desde Suricata se capturan solo los ficheros eve.json de los que se generará una copia cada día configurada por el sistema de **rotatelog**
- La transferencia de ficheros se iniciará cada vez que se arranque el sistema Windows realizando una petición de ficheros pendientes ala sonda IDS.
- Para la transferencia se utilizará el protocolo SFTP y la aplicación psftp.exe.

3.2.2.1. Preparación de la rotación de ficheros log de Suricarta.

Se define la rotación diaria para el fichero *eve.json*. Esto se configura en el fichero de configuración:

```
/var/log/suricata/*.json
{
    daily
    rotate 15
    missingok
    nocompress
    dateext
    dateformat -%d-%m-%Y-.log
    create 666 pi pi
    sharedscripts
    postrotate
            /bin/kill -HUP $(cat /var/run/suricata.pid)
    endscript
}
```

/etc/logrotate.d/suricata

Se configura el fichero con las siguientes características:

- Que el proceso de rotado sea diario.
- Se utiliza el parámetro missingok para no enviar la señal de error al SO.
- Que guarden 15 ficheros antes de borrarlo. Se considera que el sistema Windows los arrancaremos con una periodicidad menor a 15 días. Si se considera que el periodo es mayor se debería aumentar este parámetro.

- El fichero no se comprimirá.
- Se formatea el fichero añadiendo la fecha con la finalidad de tener controlado el contenido de los datos de cada fichero.
- Es importante que los fichero pertenezcan del usuario **pi**, y tengan permiso de lectura. En caso contrario no se podrían leer y borrar desde una sesión SFTP del usuario **pi**.
- Se finaliza el proceso de Suricata para que vuelva a crear el nuevo fichero **eve.json**.

Realizamos una prueba y vemos como se van almacenando los ficheros cada día a la espera de ser recogidos por el equipo Windows 10, donde se donde se tiene instalados el nodo ElasticSearch.

```
# pwd
/var/log/suricata
# ls -l
total 184980
-rw-rw-rw- 1 pi   pi    10910097 abr 22 08:44 eve.json
-rw-rw-rw- 1 pi   pi   131348486 abr 22 06:25 eve.json-22042019.log
```

3.2.2.2. Preparación de la carga de datos desde el equipo Windows.

El sistema de automatización de carga de datos desde el equipo Windows se configura mediante la realización de los siguientes pasos:

Configurar el sistema SFTP.

1. Se descarga la aplicación PSFTP.exe[18] , disponible en la siguiente URL :

https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe

2. Se copia la aplicación en el directorio C:\ELK\FTP donde se prepara el sistema de transferencia.

3. Se configura un script **(ftp.txt)** para indicarle a PSFTP mediante instrucciones FTP, los pasos que tiene que seguir para descargarse los ficheros **eve.json-*** y luego borrarlos.

```
lcd c:\ELK\ftp\descarga
cd /var/log/suricata
mget eve.json-*
del eve.json-*
quit
```

ftp.txt

[18] Psftp.exe *https://the.earth.li/~sgtatham/putty/latest/w64/psftp.exe*

4. Se prepara un fichero .bat para ejecutar la descarga de datos con la instrucción

```
c:\elk\ftp\psftp pi@192.168.0.5 -pw contraseña -b c:\elk\ftp\ftp.txt
```

```
echo 'CARGDANDO DATOS DE LA SONDA IDS...'
c:\elk\ftp\psftp.exe  pi@192.168.0.5  -pw  contraseña  -b  c:\elk\ftp\
ftp.txt
copy c:\elk\ftp\descarga\* c:\elk\ftp\historico\
copy c:\elk\ftp\descarga\* c:\elk\ftp\
del /q c:\elk\ftp\descarga\*
```

cargadatos.bat

5. Se prepara en el Programador de Tareas de Windows para que se lance el fichero **cargadatos.bat** al iniciarse el equipo.

- Se crea una nueva tarea en el **Programador de Tareas de Windows**

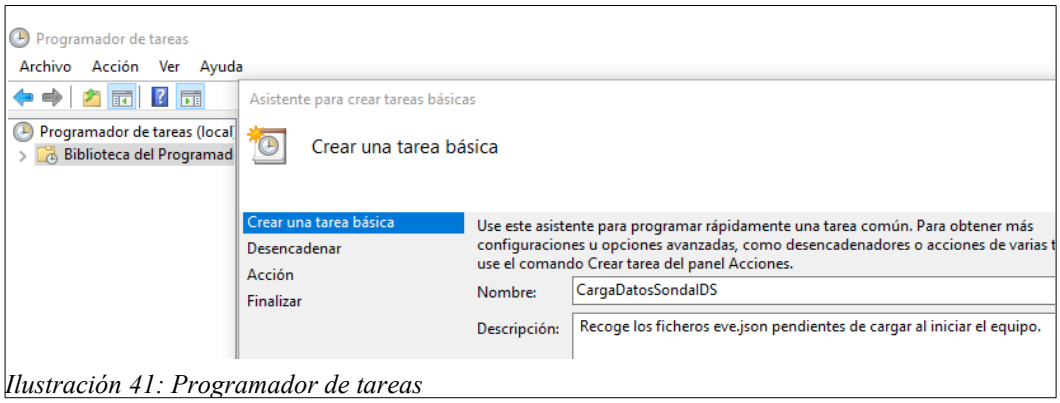


Ilustración 41: Programador de tareas

- Se indica que se lanzará al iniciarse el equipo.

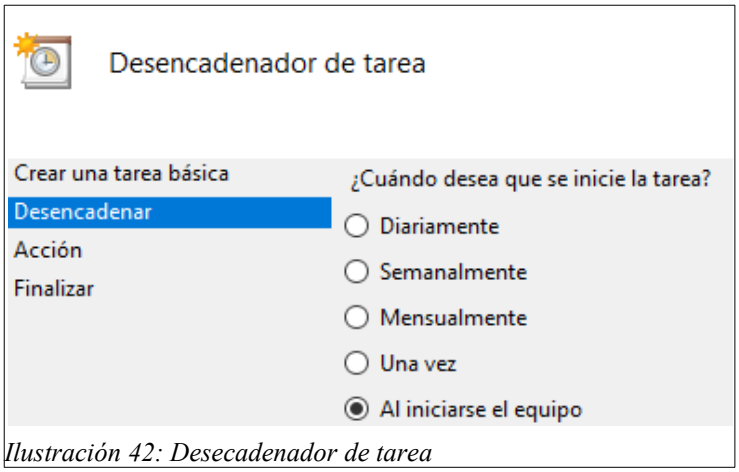


Ilustración 42: Desecadenador de tarea

- Cuando se lance la tarea, se indica que se va a iniciar un programa

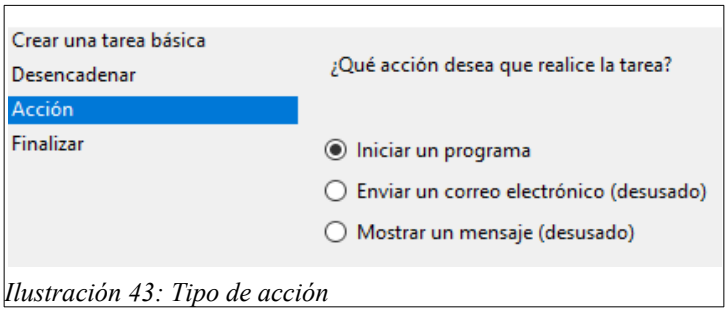


Ilustración 43: Tipo de acción

- Se indica donde se encuentra el programa a lanzar. En este caso será el fichero **bat** preparado anteriormente **cargadatos.bat**

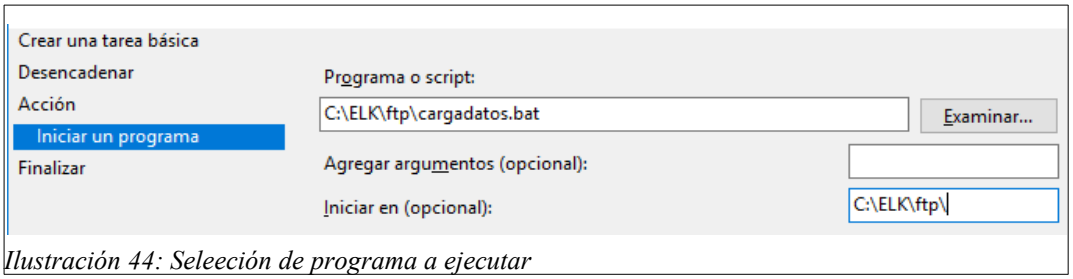


Ilustración 44: Seleeción de programa a ejecutar

- Se realizar un prueba se comprueba que transfiere los ficheros correctamente.

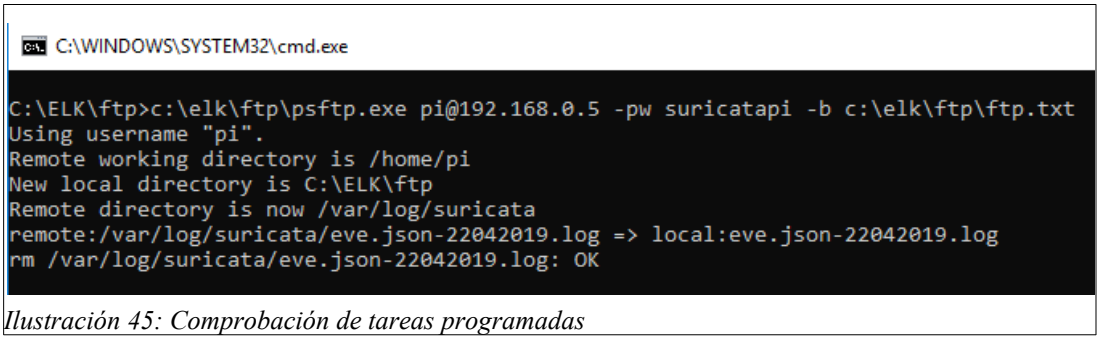


Ilustración 45: Comprobación de tareas programadas

Comprobación del sistema de carga de datos utilizando Logstash.

Se realizan los siguientes pasos para comprobar que la carga de datos funciona correctamente:

1. Ejecutamos Logstash con el fichero de configuración **eve.conf** que se ha preparado para cargar los ficheros eve.json de Suricata.

2. Se comprueba que si lanzamos Logstash este queda a la espera de la llegada de un fichero nuevo.

```
[2019-04-22T09:34:04,548][INFO ][logstash.agent           ]
Successfully started Logstash API endpoint {:port=>9600}
```

3. Cuando se lanza el proceso de carga por SFTP, detecta el fichero pero no lo carga de momento por que está ocupado por el proceso SFTP.

```
[2019-04-22T09:34:34,602][WARN  ][filewatch.readmode.handlers.readfile]
failed        to        open        c:/elk/ftp/eve.json-22042019.log:
java.nio.file.FileSystemException:    c:\elk\ftp\eve.json-22042019.log:
El proceso no tiene acceso al archivo porque est├í siendo utilizado
por otro proceso.
,
["sun.nio.fs.WindowsException.translateToIOException(WindowsException.
java:86)",
"sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.jav
a:97)",
"sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.jav
a:102)"]
```

4. Cuando acaba la transferencia inicia la carga del fichero.

```
...
{
    "@timestamp" => 2019-04-22T07:37:40.450Z,
     "dest_port" => 443,
      "@version" => "1",
     "timestamp" => "2019-04-22T06:25:02.000128+0200",
    "event_type" => "flow",
        "src_ip" => "192.168.0.106",
       "dest_ip" => "216.58.214.170",
          "path" => "c:/elk/ftp/eve.json-22042019.log",
          "type" => "SuricataIDPS",
       "flow_id" => 420622269061059,
          ...

          "host" => "CENTER",
         "proto" => "UDP",
      "src_port" => 49330,
     "app_proto" => "failed"
}
...
```

5. Se comprueba que después de la carga se borra el fichero.

Preparar el arranque automático del sistema.

Para que los procesos arranquen de forma automática además del proceso de **Carga de Datos** visto anteriormente, deben estar iniciados los proceso de **Logstash** y **Kibana.**

Para realizar esto se ha intentado hacer creando mediante Servicios de Windows, pero han surgido una serie de problemas para los que se necesitaría más tiempo para resolverlos.

Para este caso práctico es suficiente con preparar el arranque de estos procesos a través del Programador de Tareas.

Para ello se han organizado en el Programador de Tareas una carpeta llamada ELK donde se han definido tres tareas.

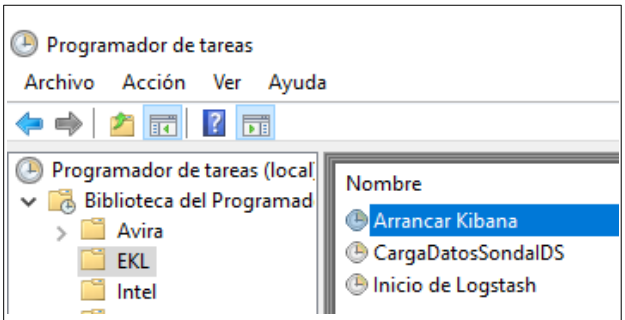


Ilustración 46: Tareas programadas para servicios ELK

- Arrancar Kibana.

  ```
  Inicia C:\ELK\kibana\bin\kibana.bat
  ```

- Inicio de Logstash

  ```
  Inicia C:\ELK\logstash\bin\logstash.bat -f C:\ELK\logstash\
  eve.conf
  ```

- CargaDatosSondaIDS

  Con un retardo de un minuto para que el equipo se pueda conectar primero a la red Wifi.

  ```
  Inicia C:\ELK\ftp\cargadatos.bat
  ```

3.2.3. Lectura de datos por Logstash.

La herramienta Logstash nos va a permitir definir como cargar los datos que obtendremos del IDS y enviarlo a Elasticsearch para que puedan ser monitorizados de forma visual con Kibana.

Para ello veremos en los siguientes puntos como se ha preparado un fichero de configuración que indique a Logstash como debe leer y tratar los datos que genera la sonda IDS, si es necesarios hacer alguna adaptación y como lo enviamos la ElasticSearch.

Para ello, se necesita conocer como se guardan los datos en el fichero **eve.json** que se genera Suricata.

3.2.3.1 Formato del fichero eve.json

Los ficheros **eve.json** que proporciona Suricata, es de donde se guarda la información que se necesita para monitorizarla posteriormente en Kibana.

Para analizar bien los datos primero, hay que conocer la organización y formato de este fichero. Esta información se puede encontrar en el manual de la página de Suricata.[19]

Se puede destacar que es un fichero en formato JSON [20], que permite un fácil tratamiento por aplicaciones como logstash o Kibana.

El fichero **eve.json** tiene una cabecera común y luego según el contenido del campo **event_type** se distingue diferente tipo de información capturada. En este caso nos centraremos en siguientes tipos de registros:

- **alert:** Para alertas generadas por el IDS
- **dns**: Para el tráfico DNS con sus peticiones y respuestas.
- **http**: Para el tráfico Http.

A continuación se muestran un ejemplo de la cabecera y de los tres tipos de eventos:

Cabecera:

```
{
    "timestamp": "2009-11-24T21:27:09.534255",
    "event_type": "alert",
    "src_ip": "192.168.2.7",
    "src_port": 1041,
    "dest_ip": "x.x.250.50",
    "dest_port": 80,
    "proto": "TCP",
    "alert": {
}
```

[19] Ref: https://suricata.readthedocs.io/en/suricata-4.1.3/output/eve/eve-json-format.html
[20] Ref: https://es.wikipedia.org/wiki/JSON

Ejemplo del cuerpo de un evento tipo alerta:

```
"alert": {
  "action": "allowed",
  "gid": 1,
  "signature_id": 1,
  "rev": 1,
  "app_proto": "http",
  "signature": "HTTP body talking about corruption",
  "severity": 3,
  "source": {
    "ip": "192.168.43.32",
    "port": 36292
  },
  "target": {
    "ip": "179.60.192.3",
    "port": 80
  },
```

Ejemplo del cuerpo de un evento tipo HTTP:

```
"http": {
    "hostname": "www.digip.org",
    "url" :"\/jansson\/releases\/jansson-2.6.tar.gz",
    "http_user_agent": "<User-Agent>",
    "http_content_type": "application\/x-gzip"
}
```

Ejemplo del cuerpo de un evento tipo DNS:

```
"dns": {
    "type": "query",
    "id": 16000,
    "rrname": "twitter.com",
    "rrtype":"A"
}
```

3.2.3.2. Carga de un fichero eve.json del IDS con LogStash.

Mediante el sistema de transferencia de ficheros un fichero eve.json se recogerá de la Sonda IDS hasta el equipo Windows donde se está ejecutando un proceso de Logstash.

Este proceso para poder cargar el fichero, debe estar configurado con un fichero, en este caso llamado **eve.conf**, que le indicará como hacerlo.

El fichero de configuración de Logstash, está organizado en tres bloques.

- **Input**: Para definir la entrada de los datos.
- **Filter** : En este bloque se puede tratar los datos para filtrarlos o realizar mapeos.
- **Output**: Define la salida o envío de datos tratados.

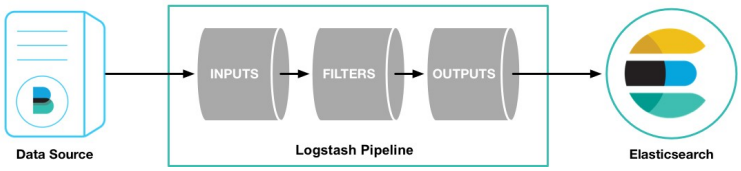


Ilustración 47: Bloques Logstash

En el siguiente enlace se puede leer una guía básica del funcionamiento de Logstash.[21]

El fichero de configuración utilizado para la carga de datos **eve.json** es el siguiente:

```
input {
  file {
    path =>  ["c:/elk/ftp/eve.json-*"]
    codec =>   json
    mode => ["read"]
    type => "SuricataIDPS"
    start_position => "beginning"
    }
}
filter {
   translate {
        field => "[src_ip]"
        destination => "[Equipo]"
        dictionary_path => ["c:/elk/logstash/config/devices.yml"]
        fallback => "Otro"
      }
    if [src_ip]  {
    geoip {
      source => "src_ip"
      target => "geoip"
      add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
      add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
    }
    mutate {
      convert => [ "[geoip][coordinates]", "float" ]
    }
    if ![geoip.ip] {
```

[21] Ref: https://www.elastic.co/es/blog/a-practical-introduction-to-logstash

```
      if [dest_ip]  {
        geoip {
          source => "dest_ip"
          target => "geoip"
          add_field => [ "[geoip][coordinates]", "%{[geoip][longitude]}" ]
          add_field => [ "[geoip][coordinates]", "%{[geoip][latitude]}"  ]
        }
        mutate {
          convert => [ "[geoip][coordinates]", "float" ]
        }
      }
    }
  }
}
output {
  stdout { codec => rubydebug }
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "suricata-%{+YYYY.MM.dd}"
    document_type => "suricata"
  }
}
```

En el fichero de configuración podemos destacar de cada apartado lo siguiente:

Input : Se realiza sobre los ficheros transferidos en modo lectura (no en modo tail) en formato JSON y se leen desde el principio al fin. Una vez leido el fichero será borrado.

Filter: De momento se han utilizado dos plugin :

- **geoip**: Para asignar Geolicalización a las direcciones IPs.
- **translate**: Para realizar un mapeo mediante diccionario entre IPs de dispositivos de la red local y su nombre.
- **drop**: Es utilizado para detectar registros de estadísticas y borrarlos.
- **mutate**: Permite renombrar, mover o reemplazar campos de un evento. En este caso se usa para convertir el tipos de registro a otro formato.

Output: La salida es enviada al nodo de ElasticSearch que hay en localhost y a la consola a modo de testeo.

Para realizar el mapeo de datos con se ha necesitado crear un fichero (**devices.yml)** con las IPs y los nombres de los dispositivos a modo de diccionario.

```
"192.168.0.100": iPhone
"192.168.0.101": Samsung
"192.168.0.102": Kuawai
"192.168.0.103": iPod
...
"192.168.0.105": iPad
"192.168.0.120": TV_LG
```

devices.yml

3.2.4. Monitorización de datos en Kibana

En el siguiente punto se quiere explicar como se pueden mostrar estos datos de una forma gráfica utilizando Kibana. Para ello nada mejor que hacerlo con unos ejemplos prácticos mostrando este proceso desde de recogida de datos con Suricata hasta la monitorización en Kibana.

3.2.4.1 Detección de conexiones de WhatsApp.

En este caso, se va a crear una regla en Suricata para que genere un evento cada vez que un dispositivo se conecte a WhatsApp. Estos eventos serán recogidos por Logstash y enviados a ElasticSearch para poder generar una gráfica de que equipos que se conectan a WhatsApp.

Para definir una regla que detecte este evento, se puede identificar los puertos que utiliza WhatsApp para conectarse desde las aplicaciones clientes. Investigando en Internet se puede averiguar que los puertos de conexión que utiliza la aplicación de WhatsApp son los siguientes:

- TCP 5222
- TCP 5223
- TCP 5228
- TCP 5242

Creación de reglas para la detección de WhasApp.

Se crean 4 reglas para detectar las conexiones a estos puertos:

```
alert tcp any any -> any 5222 (msg:"CONEXION WhatsApp!!";
flow:established,to_server; sid:1000001; rev:1;)
alert tcp any any -> any 5223 (msg:"CONEXION WhatsApp!!";
flow:established,to_server; sid:1000002; rev:1;)
alert tcp any any -> any 5228 (msg:"CONEXION WhatsApp!!";
flow:established,to_server; sid:1000003; rev:1;)
alert tcp any any -> any 5242 (msg:"CONEXION WhatsApp!!";
flow:established,to_server; sid:1000004; rev:1;)
```

/etc/suricata/rules/parental-control.rules

Se reinicia el servicio de suricata:

```
root@suricatapi:/etc/suricata/rules# service suricata restart
```

Se prueba con un dispositivo una conexión con WhatsApp y se comprueba en el fichero fast.log que suricata la generados algunas alarmas.

```
04/23/2019-19:07:04.892782    [**]  [1:1000001:1]  CONEXION  WhatsApp!!
[**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.100:61394
-> 31.13.83.49:5222
04/23/2019-19:07:04.892956    [**]  [1:1000001:1]  CONEXION  WhatsApp!!
[**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.100:61394
-> 31.13.83.49:5222
04/23/2019-19:07:04.893183    [**]  [1:1000001:1]  CONEXION  WhatsApp!!
[**] [Classification: (null)] [Priority: 3] {TCP} 192.168.0.100:61394
-> 31.13.83.49:5222
```

/var/log/suricata/fast.log

En el fichero eve.json también aparecen eventos de tipo Alarma sobre las conexiones de WhatsApp.

```
{"timestamp":"2019-04-
23T19:08:06.470409+0200","flow_id":58812163912678,"in_iface":"eth0","e
vent_type":"alert","src_ip":"192.168.0.100","src_port":61377,"dest_ip"
:"17.252.76.33","dest_port":5223,"proto":"TCP","alert":
{"action":"allowed","gid":1,"signature_id":1000002,"rev":1,"signature"
:"CONEXION WhatsApp!!","category":"","severity":3}}

{"timestamp":"2019-04-
23T19:09:16.844162+0200","flow_id":1818891919328213,"in_iface":"eth0",
"event_type":"alert","src_ip":"192.168.0.101","src_port":48301,"dest_i
p":"52.50.6.66","dest_port":5223,"proto":"TCP","alert":
{"action":"allowed","gid":1,"signature_id":1000002,"rev":1,"signature"
:"CONEXION WhatsApp!!","category":"","severity":3}}
```

/var/log/suricata/eve.json

Preparar la recogida de datos en Kibana.

- Entramos en Kibana y en el apartado de Index Management de la configuración comprobamos que tenemos datos de los últimos días.

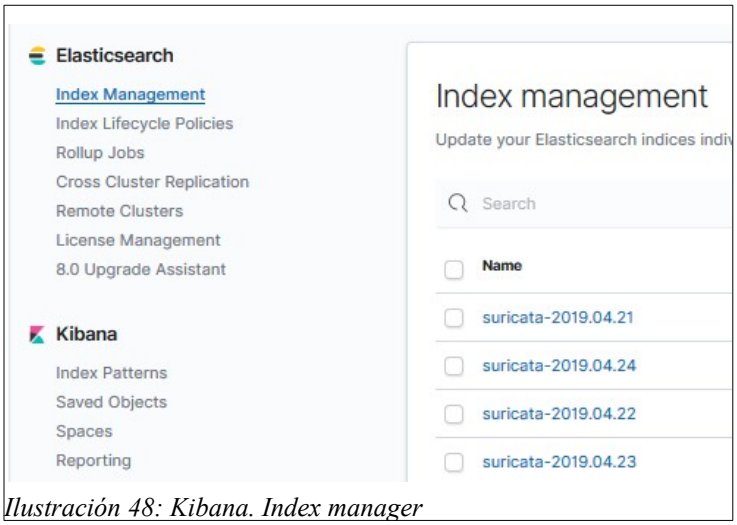


Ilustración 48: Kibana. Index manager

Se prepara una consulta de datos.

- Para ello, en el apartado Discover del menú de Kibana se pueden ver los datos en bruto capturados por Kibana.

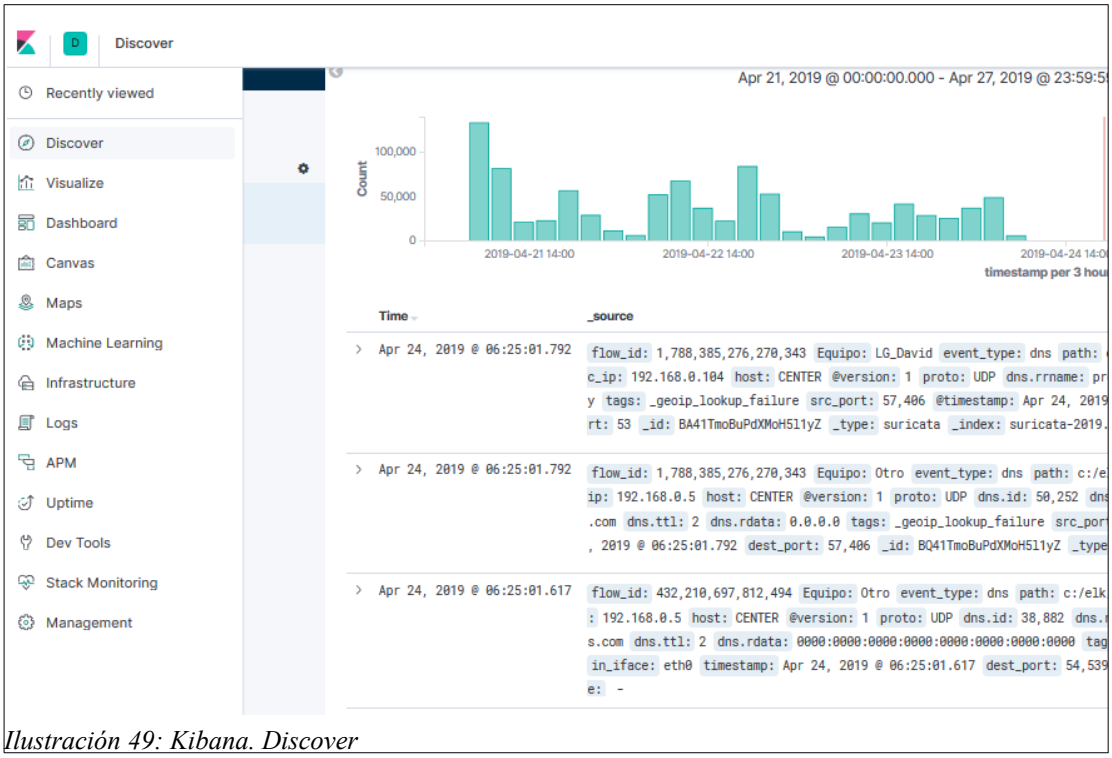


Ilustración 49: Kibana. Discover

- Se prepara una consulta filtrando por Tipo de Evento = **alert** y que el Contenido sea ***"CONEXION Whatsapp!!"*** que es el String que se había configurado en la alarma.

Ilustración 50: Kibana. Filtrar datos

- Se añade a la consulta el campo de Time y Equipo.

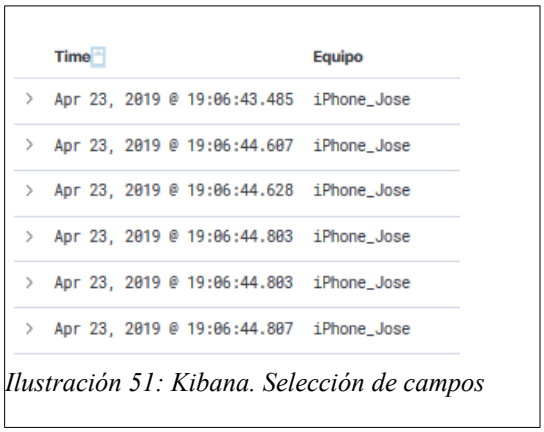


Ilustración 51: Kibana. Selección de campos

- Se guarda la búsqueda para utilizarla posteriormente.

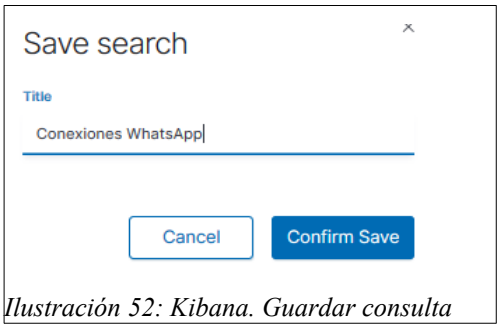


Ilustración 52: Kibana. Guardar consulta

Configurar un gráfico para mostrar los datos de la consulta.

- En el apartado Visualization de Kibana se selecciona la opción de nueva visualización y se elije el tipo de gráfico. En este ejemplo se va a utilizar un gráfico de tarta.

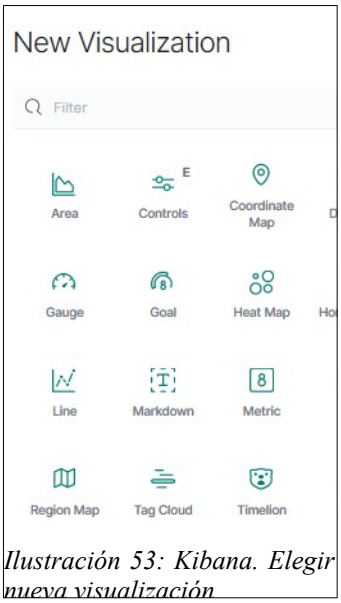


Ilustración 53: Kibana. Elegir nueva visualización

- Seleccionamos el Split por el que queremos que se divida la tarta. En este ejemplo se indica que se divida el queso por el campo Equipo. El gráfico mostrado es el siguiente:

Ilustración 54: Kibana. Gráfico tarta de uso de Whatsapp

3.2.4.2. Monitorización de los servidores más visitados

Los pasos seguidos para este ejemplo son los siguientes:

- Se prepara un filtro donde se elige el tipo de evento DNS y que los registros sean de tipo respuestas, es decir que se ha efectuado la traducción de nombre y por lo tanto Pi-hole ha autorizado su acceso.

Ilustración 55: Kibana. Filtro datos DNS

- Seleccionar los campo de nombre del dominio consultado.

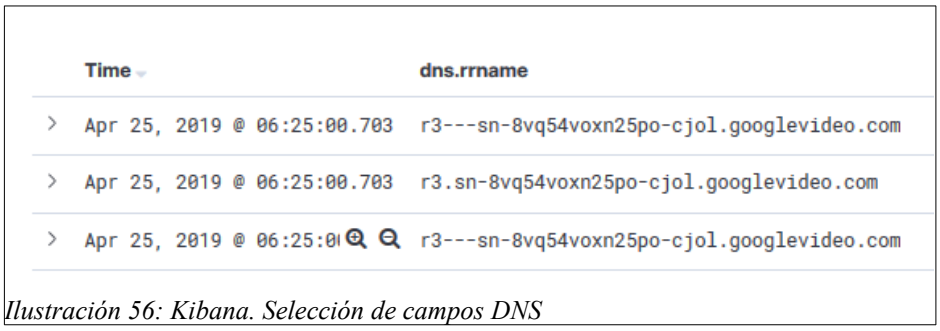


Ilustración 56: Kibana. Selección de campos DNS

- Preparar un gráfico con los dominios más consultados.Este puede ser de tarta o de barras.

Ilustración 57: Kibana. Gráfico de tarta de sitios más visitados

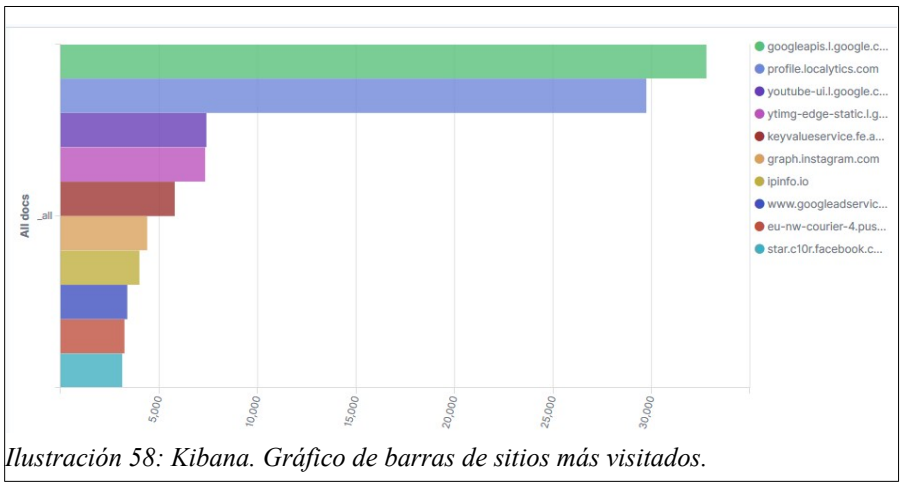


Ilustración 58: Kibana. Gráfico de barras de sitios más visitados.

3.3. Pruebas de monitorización y control parental

Una vez tenemos todo el sistema implantado en este apartado se valorarán las posibilidades del sistema para la monitorización y el control parental.

Para ello se realizarán algunas pruebas de concepto o PoC (del inglés proof of concept) implementando alguna utilidad del sistema como ayuda el control parental, con la finalidad de comprobar su utilidad.

Hay que destacar que con el tiempo invertido en la instalación de los distintos componentes junto a la problemas que han ido apareciendo, han desviado la planificación establecida quedando poco tiempo para realizar más pruebas.

Las pruebas realizadas se presentan como salvaguardas de tres de las amenazas a las que se exponen los menores:

- Uso excesivo de dispositivos conectados a internet.
- Acceso a contenido inadecuado para menores.
- Detección de algún tipo de malware

3.3.1. PoC. Control de dispositivos conectados.

Uno de los objetivos del proyecto es poder monitorizar el tiempo que están los dispositivos conectados. La finalidad de esto es ofrecer datos a los padres para poder mediar con sus hijos en lo que respecta al riesgo de la tecno-adicción y poder llegar a acuerdos consensuados sobre el uso responsable de las TIC.

Con esta finalidad, utilizando un sistema centralizado, se propone la siguientes solución:

Control de respuesta de los dispositivos mediante el uso ping.

El objetivo se ha centrado en los dispositivo móviles. La idea es realizar un ping a cada dispositivo en un intervalo corto de cada 10 minutos con la finalidad de realizar unos registros de las respuestas y de esta forma poder sacar conclusiones.

Estos datos de guardarían en un fichero donde se registraran las respuestas de los ping. El proceso se puede lanzar utilizando el sistema cron guardando un timespam de unix de cada ejecución.

Más adelante se podrá utilizar estos datos para analizar y obtener estadísticas sobre el tiempo de uso de los dispositivos.

Implementación:

Para implementar esta solución se han realizado pruebas con la aplicación FPING, que permite una mayor control que el ping convencional ya que en una misma linea de comando se puede configurar para realizar un ping a varios dispositivos.

La instalación se realiza con apt-get:

```
# apt-get install fping
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  fping
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 8 no
actualizados.
```

Se puede averiguar que dispositivos están disponibles en la red especificando con el parámetro -g la red local.

```
 fping  -g 192.168.0.0/24 2> /dev/null | grep alive
192.168.0.1 is alive
192.168.0.2 is alive
192.168.0.3 is alive
192.168.0.4 is alive
192.168.0.5 is alive
192.168.0.102 is alive
192.168.0.101 is alive
192.168.0.111 is alive
192.168.0.121 is alive
192.168.0.106 is alive
```

Se puede mostrar utilizando la opción -d los nombres de los equipos para tener una lectura más sencilla del resultado:

```
# fping -d -g 192.168.0.100 192.168.0.125 2> /dev/null | grep alive
Samsung is alive
Kuawai is alive
Windows10.pilan is alive
LG is alive
PlayStation.pilan is alive
Huawei.pilan is alive
```

Por otro lado como se pretende centrar el control sobre los móviles de casa y el ipad se acotará las IPs a estos dispositivos.

```
fping -d -g 192.168.0.100 192.168.0.105 2> /dev/null | grep alive
Kuawai is alive
LG is alive
Samsung is alive
```

Finalmente podemos ajustar el comando fping para que añada un timestamp del sistema.

```
fping   -n   -g 192.168.0.100 192.168.0.105 2> /dev/null | grep alive
| perl -nle 'print scalar(localtime), " ", $_'
Sun Apr 28 16:47:00 2019 LG is alive
Sun Apr 28 16:47:00 2019 iPad.pilan is alive
Sun Apr 28 16:47:00 2019 Samsung is alive
Sun Apr 28 16:47:00 2019 Kuawai is alive
```

Una vez se tiene preparado el comando fping, se puede guardar los resultados en un fichero para poder analizarlo en el futuro y ejecutarlo cada 10 minutos con **cron**.

Para ello se creará el fichero ***testping*** con el siguiente contenido.

```
 fping  -n  -g 192.168.0.100 192.168.0.105 2> /dev/null | grep alive
|   perl   -nle   'print   scalar(localtime),   "   ",   $_'   >>
/var/log/suricata/conectados.log
```

El resultado de los pings se irán almacenando en */var/log/suricata/conectados.log*

Se prepara el cron del usuario pi para que ejecute testping cada 10 minutos.

```
# m h  dom mon dow   command
*/10 * * * *    /home/pi/testping
```

Conclusiones:

Las pruebas no son definitivas, ya que se ha comprobado que hay móviles que responden al ping aunque no estén siendo usados en ese momento.

Las conclusiones finales es que el método no es válido para cumplir los objetivos planteados, ya que en las estadísticas aparecen dispositivos continuamente conectados cuando no es así. Por lo tanto se plantea como trabajo futuro, buscar otros métodos para poder monitorizar el uso de Internet de forma centralizada.

3.3.2. PoC. Bloquear acceso contenidos inadecuados.

El siguiente ejemplo muestra como utilizar el sistema para el bloqueo de contenido inapropiado para los menores.

En este caso se va a intentar bloquear el acceso a contenido pornográfico utilizando Pi-hole.

Para ello utilizaremos una lista de dominios con contenido pornográfico y configuraremos Pi-hole para que el servicio DNS no responda a las peticiones sobre estos dominios.[22]

La lista utilizada como ejemplo es la siguiente:

https://raw.githubusercontent.com/chadmayfield/pihole-blocklists/master/lists/pi_blocklist_porn_top1m.list

[22] Ref: https://mangolassi.it/topic/16905/add-porn-blocking-to-your-pi-hole

El proceso realizado es el siguiente:

- Añadir lista de dominios a bloquear en el apartado Blocklist de la configuración de Pi-hole.

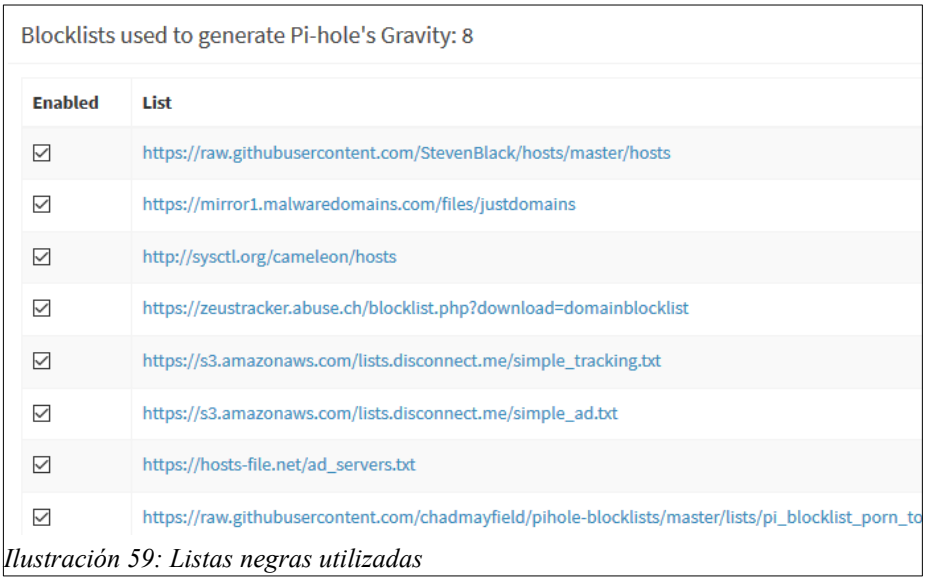


Ilustración 59: Listas negras utilizadas

- Probamos a acceder a un dominio prohibido.

```
>nslookup www.youporn.com
Servidor:  suricatapi
Address:  192.168.0.5
Nombre:  www.youporn.com
Addresses:  ::
          0.0.0.0
```

Se puede comprobar en que devuelve la IP 0.0.0.0, es decir que ha bloqueado su acceso.

- Si accedemos a la página web desde el navegador lo podemos comprobar.

Ilustración 60: Comprobación de acceso restringido.

- En cambio, si se deshabita el servicio de Pi-hole durante un minuto y hacemos la misma prueba se puede comprobar que si que devuelve la dirección IP.

```
nslookup www.youporn.com
Servidor:  suricatapi
Address:  192.168.0.5
Respuesta no autoritativa:
Nombre:  youporn.com
Address:  216.18.168.116
Aliases:  www.youporn.com
```

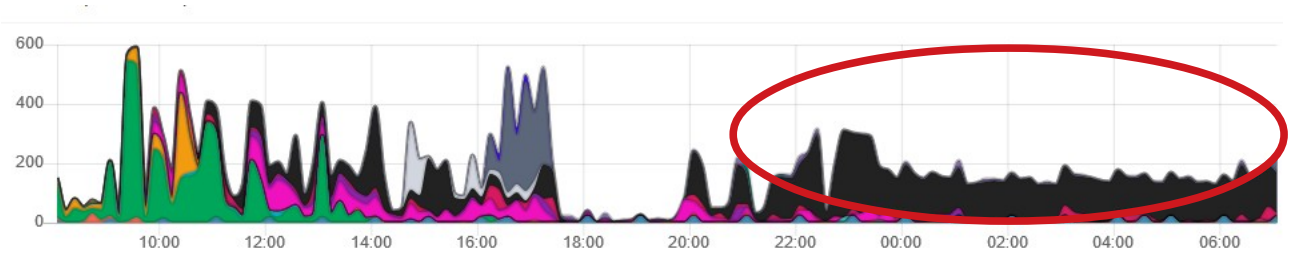
3.3.3. PoC. Detección de actividad sospechosa en un dispositivo.

En la siguiente prueba, se muestra un caso real de como detectar malware en un dispositivo de la red, utilizando el sistema.

Pasos de la investigación de posible malware fueron los siguientes:

1. Se detecta un tráfico excesivo por la noche cuando la familia está durmiendo.

Esto nos pone en alerta ya que puede ser que algún dispositivo tenga algún malware instalado y este enviando o consultando información.



2. Se identifica el dispositivo con IP 192.168.0.104, desde donde se genera el tráfico sospechoso.

3. En el siguiente gráfico, la banda azul pertenece al tráfico bloqueado. Se puede observar que el 85% del tráfico ha sido bloqueado posiblemente al ser llamadas a dominios incluidos en alguna lista negra.

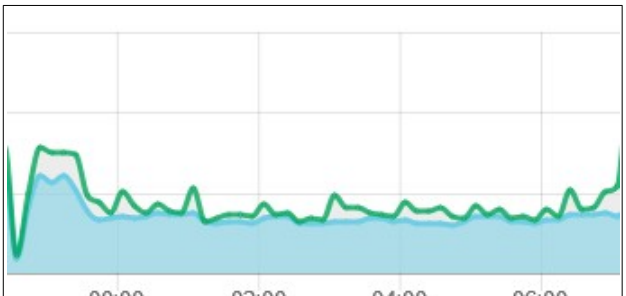


Ilustración 61: Tráfico sospechoso

4. Se analiza qué dominios a los que se han realizado consultas en este periodo y sobre todo cuales han han sido bloqueados. Los dominios son los siguientes:

- ipapi.co : Resuelto con la IP: 104.25.209.99
- ipinfo.io : Resuelto con la IP: 216.239.32.21
- freegeoip.net : Bloqueado por gravity.list
- profile.localytics.com : Bloqueado por gravity.list

5. Se investiga si la sonda IDS tiene alguna alarma en ese periodo de tiempo y se encuentra la siguiente alarma disparada desde el mismo dispositivo.

```
04/21/2019-06:32:10.461574  [**] [1:2020716:3] ET POLICY Possible
External IP Lookup ipinfo.io [**] [Classification: Potential Corporate
Privacy Violation] [Priority: 1] {TCP} 192.168.0.104:49906 ->
216.239.32.21:80
```

Suricata IDS, nos avisa de una amenaza sobre una posible violación de la privacidad.

6. Investigación sobre los dominios sospechosos:

Dominios no bloqueados:

- **https://ipinfo.io**
- **https://ipapi.co/**

Estos dos dominios facilitan la localización de una IP, lo cual puede ser usada por un software malicioso para geolocalizar el dispositivo infectado.

Se puede ver en la página como son una simple llamada a curl, se puede obtener los datos en formato JSON para una fácil procesamiento por una aplicación.

```
$ curl https://ipapi.co/8.8.8.8/json/
{
"ip" : "8.8.8.8"
"city" : "Mountain View"
"region" : "California"
"region_code" : "CA"
"country" : "US"
"country_name" : "United States"
"continent_code" : "NA"
"in_eu" : false
"postal" : "94035"
"latitude" : 37.386
"longitude" : -122.0838
"timezone" : "America/Los_Angeles"
```

```
"utc_offset" : "-0700"
"country_calling_code" : "+1"
"currency" : "USD"
"languages" : "en-US,es-US,haw"
"asn" : AS15169
"org" : "Google LLC"
}
```

Dominios bloqueados:

- **profile.localytics.com:** Parece ser un servicio para crear perfiles con la finalidad de realizar un marketing personalizado.

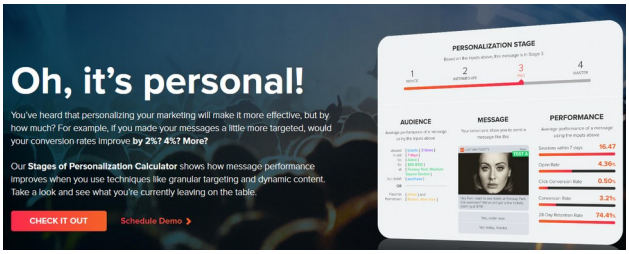


Ilustración 62: Localytics

Según su página web [23] "Localytics proporciona una API REST para almacenar y recuperar información de Perfiles sobre sus usuarios. Si desea exportar de forma masiva o importar perfiles, visite Profile Exports o Profile Imports."

- **freegeoip.net :** Es otro dominio para realizar Geolocalización.

Esta incluido en reglas de amenazas por que es muy utilizado por Malwares, pero también pude ser usado de forma legítima por aplicaciones a las que se les ha dado permiso para ello.

https://www.snort.org/rule_docs/1-46664

https://www.malware-traffic-analysis.net/2017/10/11/index2.html

7. Conclusiones :

Hay algunos programas maliciosos que lo primero que hacen cuando infectan un dispositivo, es verificar y Geolocalizar la dirección IP pública para informar al desarrollador de malware. Por lo que se puede investigar por la red, este comportamiento lo realizan muchos programas maliciosos como Cryptowall 2.0, Kriptovo, ZeroAccess …

Por otro lado, el tráfico también puede ser generado simplemente por una App que se le haya dado permiso de forma inconsciente, para realizar peticiones de Geolocalizción y captura de perfiles de comportamiento.

[23] Localytics https://www.localytics.com/

8. Acciones a realizar.

Instalar una app en el móvil comprometido para averiguar qué app está realizando este tráfico y eliminarla.

Esto lo pueden realizar algunas aplicaciones de tipo Firewall que monitorizan las conexiones de red que realizan las Apps del móvil. Un ejemplo de aplicación en Android para realizar esto podría ser NetGuard:

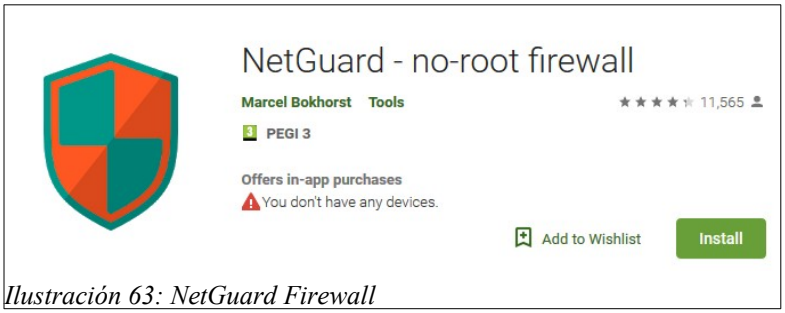


Ilustración 63: NetGuard Firewall

4. Conclusiones finales

A continuación se describen las conclusiones finales del trabajo así como algunas lecciones aprendidas:

- En la puesta en práctica del proyecto ha quedado patente la utilidad y necesidad de este tipo de soluciones en el ámbito familiar.
- Se ha podido comprobar que tanto Suricata IDS como la Pila Elastic son aplicativos muy potentes y con capacidad de configuración acorde a los objetivos del proyecto.
- La Pila Elastic ofrece mucha flexibilidad a la hora de tratar datos (Logstash) y mostrarlo de forma gráfica (Kibana) y se podrían preparar un completo Dashboard.
- Por otro lado la Pila ELK es demasiado pesada para funcionar en la propia Raspberry-pi. Se ha optado en este proyecto por instalarla en Windows junto con un sistema de sincronización de ficheros de datos.
- Sería interesante que todo el producto estuviera en la misma Raspberry-pi. Para ellos se podría realizar un adaptación del Dashboard de Pi-hole de forma que también muestre datos sobre alarmas detectadas por Suricata.
- Hay amenazas en la red, que no pueden tratarse con un sistema de control parental como este ya que podemos controlar donde se conectan los dispositivos, pero no el contenido de las comunicaciones que suelen ir cifradas.
- Un sistema como este, no puede dar soluciones completas al control parental, sino que deben utilizarse como un complemento que ofrece información y ayuda a los padres a realizar un buen acompañamiento de sus hijos en lo que respecta al uso responsable de las TIC.

4.1. Seguimiento de la planificación establecida.

A continuación se realiza una revisión sobre la planificación establecida en el inicio del proyecto.

Se puede decir que en general se detecta un pequeño retraso a causa de problemas encontrados en la implementación, que si bien se contempló en el análisis de riesgos del proyecto, no se pudo prever en ese momento el tiempo de retraso que podría penalizar en la planificación.

También es cierto que en muy poco tiempo se ha tenido que implementar distintos sistemas (Raspberry-pi, Suricata, Pila Elastic, Pi-hole..), lo que conlleva una considerable carga de trabajo de estudio de documentación, configuración y solucionar pequeños problemas no previstos en un inicio.

También se han detectado tareas que no se han podido finalizar, pero si que han consumido tiempo en su investigación y por lo tanto han afectado al plan de trabajo previsto.

4.1.1. Problemas encontrados en la implementación del proyecto.

A continuación se detallan algunos de los problemas encontrados en la implementación que han retrasado la planificación general del proyecto.

4.1.1. Configuración de Raspberry-pi como punto de acceso.

Se preparó un prototipo, para comprobar si bajaba la calidad del servicio utilizando una la Raspberry-pi como punto de acceso wifi y Suricata actuando de IPS. Se realizó una instalación completa en otra tarjeta SD y finalmente se descartó esta solución por la caída de velocidad de acceso a internet que se detectó.

4.1.2. Estudio de diseño final del sistema ELK.

Se hicieron prueba de instalación de ElasticSearch y Logstash en la propia Raspberry-pi, pero se descartó, principalmente para evitar cargar demasiado el dispositivo y por diversos problemas en la compilación e instalación de los productos de Elastic en la propia Raspberry-pi que estaban consumiendo mucho tiempo de investigación.

4.1.3. Carga de datos con logstash.

Durante más de tres días se realizaron pruebas sin resultado para cargar datos con logstash por parte del PC Windows10 desde una unidad SAMBA de la Raspberry-pi mapeada para poder acceder directamente a los fichero logs y utilizando el método TAIL.

Finalmente de desestimo este sistema ya que no funcionaba y se impuso la solución de transferencias de ficheros por FTP con periodicidad diaria.

4.1.4. Instalar ELK en un windows10.

Una vez se decide no instalar la pila ELK en la Raspberry-pi, la opción más popular en un hogar es instalarla en un PC con Windows. Esto teniendo en cuenta que el PC no está arrancado las 24 horas del día como un servidor.

Es por ello que se tuvo que buscar una solución que permitiera transferir los datos generados por Suricata al nodo de Elastic Search. La solución fue programar un sistema de transferencia de datos mediante SFTP y con un control de rotado de log diarios mediante la configuración de logrotate.

4.1.5. Tareas pendientes en la implementación.

Finalmente se ha conseguido la implementación de un sistema funcional que permite el control de los dispositivos en el ámbito familiar, pero ha faltado tiempo para experimentar lo suficiente la utilidad de uso como sistema de control parental y realizar más pruebas de concepto.

4.2. Evaluación de objetivos alcanzados

A continuación se realiza una valoración del cumplimiento de los objetivos establecidos al inicio del proyecto.

Objetivos de investigación:

- **Investigación sobre como podríamos usar el IDS/SIEM para funciones de control parental.**
  Si bien se ha podido estudiar las posibilidades del sistema para dar respuesta a las necesidades de control parental, se ha detectado que es complicado detectar algunos eventos y que queda mucho por explorar.
- **Investigación sobre las posibilidades que ofrecen los sistemas de detección de intrusos para dispositivos de bajo coste.**
  Se considera que estos dispositivos ofrecen una oportunidad para la implementación de estos sistemas ya que sin mucho consumo eléctrico y coste inicial permiten tener un sistema funcional.
- **Investigación y estudio de Suricata IDS.**
  Se ha investigado como utilizar Suricata en lo que respecta a la configuración y la definición de reglas realizando algún ejemplo como la detección de WhatsApp.
- **Investigación y estudio de sistemas SIEM que puedan integrarse con el IDS en contexto de este trabajo.**
  Se ha podido estudiar el uso de la pila Elastic (ELK) en lo referente al tratamiento de log con Logstash y diseño de gráficos con Kibana.

Objetivos de implantación:

- **Instalación y configuración de dispositivos necesarios y una correcta topología de red.**

  Este objetivo se puede considerar cumplido completamente ya que se han instalado y configurado todos los dispositivos necesarios en la topología de red elegida.
- **Aprender a instalar y configurar Suricata IDS.**

  El objetivo se considera cumplido en lo referente a la instalación y configuración para el funcionamiento del sistema. También es cierto que por falta de tiempo no se ha llegado a realizar una configuración más ajustada en lo referente a los kernel del procesador a utilizar por el IDS.
- **Aprender a configurar reglas y realizar pruebas de detección.**

  Si bien se ha aprendido a configurar reglas en Suricata y se han realizado pruebas concretas, se ha podido comprobar la gran cantidad de posibilidades a la hora de definir reglas. Es por ello que queda mucho por experimentar para sacar el máximo partido al sistema.

- **Aprender instalar y configurar un sistema de monitorización para Suricata IDS, de forma que su uso sea sencillo y visual.**

  Se ha instalado la pila Elastic en un Windows10 y se ha configurado el sistema para que se transfieran datos de un sistema a otro y se puedan mostrar estos de forma visual. Además se ha instalado Pi-hole que también monitoriza datos del sistema. El objetivo está cumplido, pero no es del todo sencillo para alguien sin conocimientos de informática. Lo ideal hubiera sido un sistema todo en uno instalado en la propia Raspberry-pi.

- **Puesta en marcha del producto final y realización de pruebas prácticas para control parental.**

  Se ha acabado la fase con la puesta en marcha de un producto final en real que monitoriza la salida a internet de una familia. Se han realizado pruebas de control parental y por lo tanto también se considera que se ha cumplido las expectativas de este objetivo. Si bien es cierto que ha faltado más tiempo en este punto final para realizar más pruebas.

La valoración global es que se han cumplido los objetivos marcados, y la experiencia ha sido dura en su trabajo pero muy positiva a nivel didáctico, con la satisfacción añadida de poder aplicarlo en la familia.

4.3. Trabajo futuro.

Una vez conseguidos los objetivos didácticos establecidos en este trabajo, se enumeran a continuación varias líneas de trabajo futuro para mejorar el proyecto:

- Se han quedado pendiente un buen sistema que detecte el tiempo que está conectado cada dispositivo. Se ha probado el proyecto mediante un testeo de dispositivos por ping, pero no ha sido útil.
- Definir un paquete de reglas útiles en el IDS para detectar aplicaciones concretas. Se ha realizado prueba con WhatsApp, pero esto se podría extender a otras aplicaciones.
- Mejorar el sistema SIEM. En el proyecto se ha utilizado la pila ELK, pero se ha tenido que instalar en un sistema Windows con las complicaciones de sincronización de datos que esto conlleva. Sería interesante utilizar un sistema de monitorización en la propia Raspberry-pi más ligero similar al utilizado en el proyecto Pi-hole (Admin LTE).
- En este trabajo se ha definido un Dashboard en Kibana con finalidades didácticas para valorar la potencia de la herramienta en lo que respecta a los objetivos del proyecto. Por lo tanto quedaría pendiente como trabajo futuro definir un Dashboard con más gráficos de utilidad para el control parental.

5. Glosario[24]

- **Ciberbullying**: El ciberacoso también denominado acoso virtual o acoso cibernético, es el uso de redes sociales para acosar a una persona o grupo de personas, mediante ataques personales, divulgación de información confidencial o falsa entre otros medios.
- **DHCP**: Protocolo de configuración dinámica de host (Dynamic Host Configuration Protocol). Es un protocolo de red de tipo cliente/servidor mediante el cual un servidor DHCP asigna dinámicamente una dirección IP.
- **DNS**: Sistema de Nombres de Dominio (Domanin name system). Sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP que permite traducir IP a nombres de dominio y viceversa.
- **Elastic Search**: es un servidor de búsqueda basado en Lucene. Provee un motor de búsqueda de texto completo, distribuido y con capacidad de multitenencia con una interfaz web RESTful y con documentos JSON.
- **FTP**: Protocolo de transferencia de archivos (File Transfer Protocol) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP.
- **Gromming**: Conductas y acciones deliberadamente emprendidas por un adulto, a través de Internet,2 con el objetivo de ganarse la amistad de un menor de edad, creando una conexión emocional con el mismo, con el fin de disminuir las preocupaciones del menor y poder abusar sexualmente de él
- **HTTP**: Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol o HTTP) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.
- **IDS**: Sistema de detección de intrusos (Intrusion detection system). Programa de detección de accesos no autorizados a un computador o a una red.
- **IPS**: Sistema de prevención de intrusos (Intrusion prevention system). Es un software que ejerce el control de acceso en una red informática para proteger a los sistemas computacionales de ataques y abusos.
- **Kibana**: Kibana es un complemento de visualización de datos de código abierto para Elasticsearch.

[24] Definiciones consultadas en Wikipedia. https://es.wikipedia.org

- **Logstash**: Herramienta para la administración de logs. Esta herramienta se puede utilizar para recolectar, parsear y guardar los logs para futuras búsquedas
- **Pi-hole:** Pi-hole es una aplicación para bloqueo de anuncios y rastreadores en Internet a nivel de red en Linux que actúa como un sumidero de DNS, destinado para su uso en una red privada
- **Raspbian**: Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian para la placa computadora Raspberry Pi.
- **Raspberry-pi:** Es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste.
- **Sexting**: Término que se refiere al envío de mensajes sexuales, eróticos o pornográficos, por medio de teléfonos móviles.
- **Sextorsión**: Es una forma de explotación sexual en la cual una persona es chantajeada con una imagen o vídeo de sí misma desnuda o realizando actos sexuales que generalmente es compartida con fines de que se haga viral mediante sexting.
- **SIEM**: Gestión de Eventos e Información de Seguridad (Security Information and Event Management)

6. Bibliografía y fuentes consultadas

6.1. Libros consultados.

- Saurabh Chhajed. «Learning ELK Stack». Alliedbooks.Packt Publishing. ISBN: 9781785887154. November 2015.

6.2. Videos consultados.

- Jesse Kurrus. "Test Case: Suricata VS Snort IDS". URL:https://www.youtube.com/watch?v=9FZEaqUAcUs
- "Suricata Network IDS/IPS System Installation, Setup and How To Tune The Rules & Alerts on pfSense". URL:https://www.youtube.com/watch?v=KRlbkG9Bh6I
- Travis Smith. "BSidesSF 110 Sweet Security Deploying a Defensive Raspberry Pi Travis Smith". URL:https://www.youtube.com/watch?v=7DFg9Ez2sJE
- Crihs Jencks. "204 Wireless Intrusion Detection System with Raspberry Pi Chris Jenks". URL:https://www.youtube.com/watch?v=BuTHfZqBBRk
- Rabimba Karanjai. "SecurityPI: IronClad your Raspberry PI - Rabimba Karanjai". URL: https://www.youtube.com/watch?v=nls9t66ecq8
- Kevin Bong. "Monitoring and Filtering Your Child's Web Media Use in our Connected World". URL: https://www.youtube.com/watch?v=3xbEFIaiEro

6.3. Páginas web consultadas.

Configuración de circuitería de red.

- Community linksys. "HACER BRIDGE CON WRT54G". URL:https://community.linksys.com/t5/Routers-Inal%C3%A1mbricos/HACER-BRIDGE-CON-WRT54G/td-p/760333
- Community linksys. "Router wireles wrt54g + gateway/ap/bridge". URL: https://community.linksys.com/t5/Routers-Inal%C3%A1mbricos/router-wireles-wrt54g-gateway-ap-bridge/td-p/420866
- TP-Link. "User Guide5-Port Gigabit Easy Smart Switch TL-SG105E". URL: https://static.tp-link.com/2018/201805/20180515/1910012413_TL-SG105E_4.0,TL-SG108E_4.0,TL-SG116E_1.0(UN)_UG.pdf https://static.tp-link.com/2018/201806/20180628/1910012421_Unmanaged%20Pro%20Configuration%20Utility_UG.pdf
- David Lodge. Pentest Partners. "How I can gain control of your TP-LINK home switch". URL: https://www.pentestpartners.com/security-blog/how-i-can-gain-control-of-your-tp-link-home-switch/

Configurar un aRaspberry-pi como punto de acceso wifi.

- MSRobotics. "Tutorial-rasbperry-pi-como-crear-un-punto-de-acceso-wifi". URL: http://msrobotics.net/index.php/laboratorio-pi/227-configura-Raspberry-pi-como-punto-de-acceso-wifi

- Phil Martin. "Using your new Raspberry Pi 3 as a WiFi access point with hostapd". URL:https://frillip.com/using-your-Raspberry-pi-3-as-a-wifi-access-point-with-hostapd/
- Raspbian France. "Create a Wi-Fi hotspot in less than 10 minutes with Pi Raspberry!". URL:https://howtoraspberrypi.com/create-a-wi-fi-hotspot-in-less-than-10-minutes-with-pi-raspberry/
- Raspberry-pi org. "Setting up a Raspberry Pi as an access point in a standalone network (NAT)". URL:https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md
- Rubén Velasco. "Manual para configurar Raspberry Pi como un router Wi-Fi". URL:https://www.redeszone.net/Raspberry-pi/manual-para-configurar-Raspberry-pi-como-un-router-wi-fi/

Sistema de detección de Intrusos

- INCIBE. "Diseño y Configuración de IPS,IDS y SIEM en Sistemasde Control Industrial". URL:https://www.incibe-cert.es/sites/default/files/contenidos/guias/doc/certsi_diseno_configuracion_ips_ids_siem_en_sci.pdf
- BRO vs SNORT or Suricata.URL: https://bricata.com/resources/white-paper/bro-vs-snot-or-suricata/
- Confederación de Empresarios de La Coruña."Herramientas Opensource: IDS (Sistema de Detección de Intrusos)". URL:https://noticias.cec.es/index.php/2017/02/22/herramientas-opensource-ids-sistema-de-deteccion-de-intrusos/
- Seguridad y Redes."Snort. Búsqueda de patrones. Reglas de contenido".Seguridad y Redes. URL:https://seguridadyredes.wordpress.com/2008/06/12/snort-busqueda-de-patrones-reglas-de-contenido/

Suricata IDS

- Suricata User GuideRelease 4.1.0-dev. URL:https://media.readthedocs.org/pdf/suricata/suricata-4.1.2/suricata.pdf
- Alfon. Blog Seguridad y Redes. "IDS / IPS Suricata. Entendiendo y configurando Suricata. Parte I". URL:https://seguridadyredes.wordpress.com/2011/02/22/ids-ips-suricata-entendiendo-y-configurando-suricata-parte-i/
- Alfon. Blog Seguridad y Redes. "Suricata IDS/IPS 1.2.1 Extracción de ficheros de sesiones HTTP tráfico de red". URL:https://seguridadyredes.wordpress.com/2012/02/01/suricata-idsips-1-2-1-extraccion-de-ficheros-de-sesiones-http-trafico-de-red/
- Alfon. Blog Seguridad y Redes. "Snort. Búsqueda de patrones. Reglas de contenido". URL:https://seguridadyredes.wordpress.com/2008/06/12/snort-busqueda-de-patrones-reglas-de-contenido/
- Blog Follow The White Rabbit. "Suricata IDS – Instalación, puesta en marcha y primera prueba". URL:https://www.fwhibbit.es/suricata-ids-instalacion-puesta-en-marcha-y-primera-prueba
- Blog Follow The White Rabbit. "Suricata IDS – Jugando con las reglas". URL:https://www.fwhibbit.es/suricata-ids-jugando-con-las-reglas

- José Vila."Presentando Suricata". URL:https://www.securityartwork.es/2015/04/30/presentando-suricata-i/
- José Vila."Presentando Suricata (II)". URL:https://www.securityartwork.es/2015/05/08/presentando-suricata-ii/
- José Vila."Presentando Suricata III". URL:https://www.securityartwork.es/2016/01/22/21589
- José Vila."Presentando Suricata (y IV)". URL:https://www.securityartwork.es/2016/04/04/presentando-suricata-y-iv/
- FloCon 2016."Suricata Tutorial". URL:https://resources.sei.cmu.edu/asset_files/Presentation/2016_017_001_449890.pdf
- Infosec Institute. "How to Configure & Use Suricata for Threat Detection".URL: https://resources.infosecinstitute.com/configure-use-suricata-threat-detection/#gref
- Sergio De Luz. "SELKS: Conoce esta distribución Linux con el sistema de detección y prevención de intrusiones". URL:https://www.redeszone.net/2017/05/30/selks-conoce-esta-distribucion-linux-sistema-deteccion-prevencion-intrusiones-suricata/

Pila de productos Elastic

- ELASTIC. "Elastic blog". URL: https://www.elastic.co/es/blog
- David. Blog Ocho bits hacen un byte. "¿Qué es y cómo funciona Elasticsearch?". URL: https://www.ochobitshacenunbyte.com/2018/08/28/que-es-y-como-funciona-elasticsearch/
- ELASTIC. "Producto kibana". URL: https://www.elastic.co/es/products/kibana
- Peter Manev. "Logstash Kibana and Suricata JSON output". URL: https://redmine.openinfosecfoundation.org/projects/suricata/wiki/_Logstash_Kibana_and_Suricata_JSON_output
- ELASTIC. "Filebeat - Suricata module". URL: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-module-suricata.html
- Lorenzo. Blog Atareao. "Compartir archivos en red con Samba en tu Raspberry". URL: https://www.atareao.es/tutorial/Raspberry-pi-primeros-pasos/compartir-archivos-en-red-con-samba/
- Rubén Velasco. "Cómo instalar un servidor Samba en Raspberry Pi para compartir carpetas en red". URL: https://www.redeszone.net/Raspberry-pi/como-instalar-un-servidor-samba-en-Raspberry-pi-para-compartir-carpetas-en-red/

Peligros en la red y control Parental

- CCN-CERT. "Informe ciberamenazas CCN-CERT 2018". URL:https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/2856-ccn-cert-ia-09-18-ciberamenazas-y-tendencias-2018-resumen-ejecutivo-2018/file.html
- INCIBE. Internet Segura for Kids. URL: https://www.is4k.es/
- INCIBE. “Guía para el uso seguro y responsable de internet por los menores”. URL:https://www.is4k.es/sites/default/files/contenidos/herramientas/is4k_guia_mediacion_parental_internet.pdf

- Dong Ngo. CNET Magazine. "iBoss Home Parental Control Wireless-N router". URL: https://www.cnet.com/reviews/iboss-home-parental-control-wireless-n-router-review/
- Milan Stanojevic. "The 15 best firewall devices to protect your home network". URL: https://windowsreport.com/firewall-device-for-home/
- Derten Ciberseguridad. "Pozos negros y frambuesas – Montando un servidor DNS y bloqueando anuncios a nivel de red con la Raspberry Pi". URL:https://derten.com/pozos-negros-frambuesas-montando-servidor-dns-bloqueando-anuncios-nivel-red-la-Raspberry-pi/
- Ben Dews. "How to implement DNS-Over-HTTPS on PiHole, Ubiquiti USG and dnsmasq devices". URL:https://bendews.com/posts/implement-dns-over-https/
- Pedro Delgado. "Cómo bloquear WhatsApp en una red WiFi ". URL:https://www.elgrupoinformatico.com/como-bloquear-whatsapp-una-red-wifi-t26789.htmlE
- El-brujo. Blog El hacker."Bloquear todos los subdominios de Facebook, Google o WhatsApp ". URL: https://blog.elhacker.net/2018/04/bloquear-todos-los-subdominios-de.Facebook-Google-WhatsApp.html
- Pi-hole. "Seven Things You May Not Know About Pi-hole" URL: https://pi-hole.net/2017/05/12/seven-things-you-may-not-know-about-pi-hole/
- "Proyecto Pi-hole". URL: https://pi-hole.net/
- Alejandro Aliaga. "Improving security on your local network". URL: https://www.sothis.tech/en/improving-security-on-your-local-network/