Universitat Oberta de Catalunya

uoc.edu

Implementación de un Sistema de Detección de Intrusos (IDS) mediante la inspección de tráfico de la red.

Miguel Angel Guinea Cabrera
Master en Ciberseguridad y Privacidad
Especialidad en Gestión.

Director del TFM: Joan Caparrós Ramírez
Profesora responsable de la asignatura: Cristina Pérez Sola

Fecha de Entrega: 1 de junio de 2021


Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

La poesía es un buen escondite
una ventriloquía deshojada de mis adentros,
un licor con los abuelos que ya no están
un río cantando con piedras de pastel de frambuesa.

Hay espacio para mí en una poesía
en los llanos de un verso
se adivina un pueblo que habita
todo lo que siento,
lo que transitan los ojos
de dentro.

Te veo a tí de nuevo,
en un corral sin huevos
un gallo negro se levanta
pica amapolas que desfallecen.

Las ranas me recuerdan que aún es verano,
que siempre es verano.

Gracias a mi mujer Carmen, por
apoyarme en la realización de este
máster y por ayudarme a ver la luz.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Implementación de un sistema de detección de intrusos IDS mediante la inspección del tráfico a través de la red.*
Nombre del autor:	*Miguel Ángel Guinea Cabrera*
Nombre del consultor/a:	*Joan Caparrós Ramírez*
Nombre del PRA:	*Cristina Pérez Sola*
Fecha de entrega (mm/aaaa):	06/2021
Titulación:	*Master Ciberseguridad y Privacidad, especialidad Gestión*
Área del Trabajo Final:	*M1.888 TFM-Análisis de Datos*
Idioma del trabajo:	*Castellano*
Palabras clave	*IDS, Intrusos, Protección entorno familiar*

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

La aceleración la implementación del trabajo a distancia y la inversión en transformación digital, el trabajar, comprar más a menudo online o estudiar en casa han traído una serie de desafíos en materia de ciberseguridad y privacidad, puesto que, además, el número de equipos informáticos por hogar, es reducido y compartido.

La intimidad del hogar, ofrece también un marco interesante, para empleados desleales o que puedan llegar a realizar prácticas delictivas.

Un ataque a una red domestica tiene u beneficio mucho mayor ahora que en el periodo pre pandemia por el número de potenciales activos involucrados: datos, fotos, información sensible, datos bancarios, información corporativa, acceso a bases de datos, … El listado es interminable.

Los sistemas de detección de intrusos (IDS, Intrusion Detection System) ofrecen una potente herramienta con la defenderse de intentos de invadir redes privadas de ámbito doméstico monitorizando el tráfico que circula por la red y

avisando en caso de detectar algún evento sospechoso. De hecho, estos sistemas son habituales en las redes corporativas, incluyéndose junto al cortafuegos o incluso con un servidor dedicado en zona especiales como una red DMZ (desmilitarizada).

El objetivo del presente trabajo persigue explorar opciones de bajo coste que permitan aumentar la visibilidad de los problemas de seguridad en los hogares y lugares destinados al trabajo a distancia. Para ello la meta es conseguir un prototipo de IDS (Intrusion Detection System) que ayude a mejorar la seguridad en el entorno familiar, así como su integración en un entorno corporativo.

Abstract (in English, 250 words or less):

The accelerating implementation of remote working and investment in digital transformation, working, shopping more often online or studying at home has brought a number of cybersecurity and privacy challenges, as the number of computers per household is small and shared.

The privacy of the home also provides an interesting setting for disloyal or potentially criminal employees.

An attack on a home network has a much greater benefit now than in the pre-pandemic period because of the number of potential assets involved: data, photos, sensitive information, banking data, corporate information, access to databases, ... The list is endless.

Intrusion Detection Systems (IDS) offer a powerful tool to defend against attempts to invade private home networks by monitoring the traffic flowing through the network and alerting if any suspicious event is detected. In fact, these systems are common in corporate networks, being included alongside the firewall or even with a dedicated server in special zones such as a DMZ (demilitarised) network.

The aim of this work is to explore low-cost options to increase the visibility of security issues in homes and remote workplaces. The goal is to achieve a prototype IDS (Intrusion Detection System) that helps to improve security in the home environment, as well as its integration in a corporate environment.

Índice

Ilustraciones

1. Introducción

1.1 Problema a resolver

1.1.1 Un virus precipita el teletrabajo

El COVID provocó la declaración de estado de emergencia a nivel nacional el sábado 14 de marzo de 2020 (ref. 2, ref. 4, ref. 5) y como principal medida la imposición de una cuarentena nacional.

Debido a la pandemia por el virus Covid-19 y sus diferentes variantes, muchas empresas se han visto obligadas a implementar precipitadamente la opción de **teletrabajo** en muchas de las empresas. Sin embargo, gran parte de ellas no estaban preparadas a nivel técnico y organizativo para ello. Recordemos que en España apenas un 4,3% de los trabajadores se conectaba a distancia para desempeñar su trabajo habitualmente. Usándose el teletrabajo como una medida de conciliación con un límite máximo de entre 1 y 2 días a la semana.

La Administración Pública prevé pasar del 18% de teletrabajo antes de la pandemia al **55% en la era poscovid**.

1.1.2 Transformación Digital y Ciberseguridad

Todas las empresas con una implementación de la norma ISO 27001, tienen la gestión de la Continuidad del Negocio como parte de los controles incluidos en el Anexo A de la citada norma. Como normativas de referencia respecto a la continuidad de negocio, encontramos las siguientes (ref.3):

- ISO 22301. Gestión de la continuidad del negocio.
- ISO 27031. Guía de continuidad de negocio referente a tecnologías de la información y comunicaciones.

En el caso de un evento *Black-Swan* (ref.1) como una pandemia, las compañías de sectores altamente tecnificados y que iniciaron su transformación digital con estrategias de adopción al cloud (ref.8), pudieron adaptarse más rápidamente que el resto de compañías que tuvieran más relegadas las inversiones en transformación digital.

Las empresas que no tenían portátiles para toda la plantilla y que no tenían recursos previstos dentro de su Plan de Continuación de Negocio / Recuperación ante desastres, tuvieron que permitir además que los trabajadores usaran sus equipos particulares como una manera de garantizar la continuidad del negocio.

Además de este problema, muchos otros fueron los problemas a los que se enfrentaron empresa y trabajadores, por citar algunos:

- No existían procedimientos preestablecidos en la empresa para el teletrabajo.
- Muchas no disponían de plataformas adecuadas para la implementación del trabajo a distancia.
- Había más trabajadores con equipos de sobremesa que con portátiles.
- No había suficientes licencias para conectarse a VPNs.
- Los trabajadores se conectan a redes no seguras poniendo en peligro datos sensibles de la empresa.

La adaptación tanto de las empresas como la del gobierno, ha hecho que se actualice la normativa existente para acomodarse a la nueva normalidad que vendrá tras la pandemia, así La publicación del Real Decreto ley 28/2020 de 22 de septiembre de trabajo a distancia (ref.7) introdujo nuevas reglas de juego para una situación de teletrabajo no regulado por más de 3 meses.

Precisamente este último es de suma importancia, puesto que el trabajo a distancia exige una mayor apuesta por la **ciberseguridad**.

El acceso a datos de la empresa desde el exterior supone una nueva puerta de entrada a posibles hackers. Por ejemplo, es habitual en las redes domésticas no cambiar la contraseña por defecto del router. Para un uso recreativo de Internet esto no supone un riesgo significativo, pero para un uso empresarial esto es algo muy sensible.

También fue significativo el número de empleados con un acceso a Internet de baja calidad o directamente inexistente, puesto que debido a las cuarentenas impuestas hubo un porcentaje de trabajadores que se mueven a segundas residencias donde no hay fibra o directamente no tienen contratado Internet. Este dato se verifica con el aumento de los empadronamientos en zonas rurales. Usar redes públicas, o incluso conectarse a redes de otras personas, se convirtió en una salida para algunas familias que no tenían contratado internet. El peligro para la seguridad por el uso de una red desconocida es extremo.

1.1.3 Ecommerce y confinamiento

Al estar en confinamiento, se aumentó el número de transacciones online y el uso de los ordenadores para cubrir más necesidades que de otra manera, no hubieran quedado atendidas o hubieran requerido suavizamiento de las medidas de confinamiento. Al inicio del confinamiento se constató que un 30% de los consumidores pretendía más compras por internet, tendencia que se elevó hasta el 53%. De hecho, el 36% de un estudio sobre hábitos de consumo, consideraba Internet el mejor lugar para comprar y sólo un 12% prefería acudir a una gran superficie. Las plataformas digitales han resultado sin duda beneficiadas en un contexto de pandemia. Sólo deben revisarse las acciones de Amazon durante los dos últimos años. En España, las medidas tomadas aumentaron un 50% la venta online durante las primeras semanas de confinamiento

1.1.4 Riesgo de ciberataque

El beneficio de un ciberataque a una red doméstica ahora es mayor que antes, puesto que se pueden explotar tanto datos personales y sensibles como información empresarial. Además, los ataques a un router medio ofrecido por un proveedor de internet, no tiene nada que ver con los routers y productos utilizados en las redes corporativas.

Si a esto unimos que los trabajadores a veces se conectan a redes vecinales, de locales, trae consigo la potencial materialización de amenazas usando un *Rogue Access Point* o punto de acceso no autorizado o con intencionalidad de atacar la confidencialidad, integridad y disponibilidad de activos personales o corporativos.

1.1.5 Problema: el incremento de riesgos a la seguridad

La transformación digital trajo la posibilidad de teletrabajar. En España, esta medida se había establecido tímidamente, pero la irrupción del COVID-19, ha potenciado esta opción con visos de quedarse como un beneficio que muchas empresas van a dar habitualmente a sus empleados, para más detalles véase Anexo Teletrabajo.

Alrededor de una tercera parte de las actividades comerciales que han optado por el teletrabajo, declaran que lo mantendrán en el futuro. Un 22,4% pretende realizar inversiones en nuevas tecnologías y un 17,4% incrementará la formación de sus trabajadores.

En un estudio de Trend Micro con entrevistas a 13.200 trabajadores remotos en 27 países, incluido España (ref. 11), los resultados muestran que casi tres cuartas partes de los trabajadores (un 64% en España) son más conscientes de las políticas de ciberseguridad de su organización desde que comenzó el confinamiento y que un 85% (89% en España) afirma que se toma en serio las instrucciones de su equipo TI.

Sin embargo, los resultados también indican que un 56% (50% en España) de los encuestados admite haber usado una aplicación que no es de trabajo en un dispositivo corporativo y el 66% (26% en España) de ellos, ha cargado datos corporativos en esa aplicación. Además, un 80% (85% en España) confiesa que usa el portátil de trabajo para la navegación personal.

Por tanto, aparte de acelerar la implementación del trabajo a distancia y la inversión en transformación digital, el trabajar, comprar más a menudo online o estudiar en casa han traído una serie de desafíos en materia de ciberseguridad y privacidad, puesto que, además, el número de equipos informáticos por hogar, es reducido y compartido.

La intimidad del hogar, ofrece también un marco interesante, para empleados desleales o que puedan llegar a realizar prácticas delictivas.

Un ataque a una red domestica tiene u beneficio mucho mayor ahora que en el periodo pre pandemia por el número de potenciales activos involucrados: datos, fotos, información sensible, datos bancarios, información corporativa, acceso a bases de datos, … El listado es interminable.

1.1.6 Necesidad de incrementar la seguridad en los hogares

La migración casi instantánea de millones de usuarios desde redes empresariales y universitarias que se monitorizan y protegen de cerca, a redes Wi-Fi domésticas en gran parte no supervisadas y a menudo inseguras, ofrece una oportunidad inmensa para los cibercriminales.

En un contexto de teletrabajo, la solución favorita para proteger los accesos externos a los recursos de las empresas, ha sido a menudo la VPN: Virtual Private Network, a menudo. Pero los entornos de los hogares son complejos (ref. 9) y la VPN sólo protege cuando se levanta el cliente SSL.

Algunos de los problemas a tener en cuenta:

- Dispositivos IoT usan el mismo router que el resto de equipos.
- Modelos de routers de los ISP con vulnerabilidades conocidas.
- WPA/WPA2 clave compartida.
- Red compartida por familia y/o amigos.
- Cuando se conectan otros equipos, estos pueden no poseer de antivirus o cortafuegos.
- Las políticas BYOD (*Bring Your Own Device*), tienen más dificultan para establecer soporte remoto y upgrades.

El escenario ideal dista en muchos casos de lo que nos encontramos actualmente:

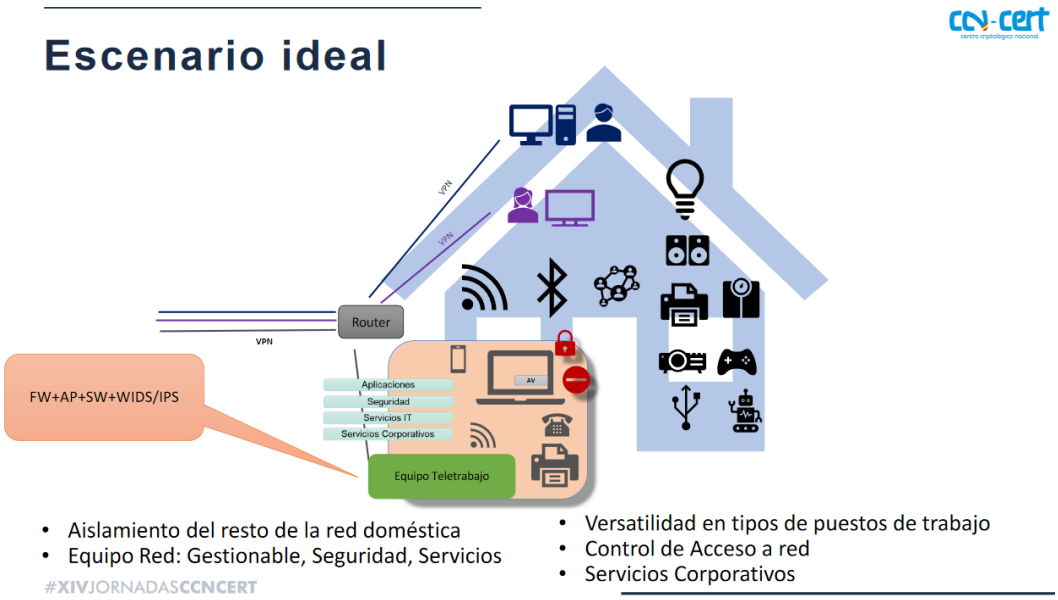

Ilustración 1: Escenario Ideal para trabajo a distancia (ref. 10)

En el escenario ideal descrito, nos encontramos con aislamiento del resto de la red doméstica y con servicios que controlan la seguridad y protegen a los empleadores y las corporaciones de los incidentes en materia de seguridad, pero también que alertan a los trabajadores en sus redes domésticas de posibles incidentes de seguridad.

Los sistemas de detección de intrusos (IDS, *Intrusion Detection System*) ofrecen una potente herramienta con la defenderse de intentos de invadir redes privadas de ámbito doméstico monitorizando el tráfico que circula por la red y avisando en caso de detectar algún evento sospechoso. De hecho, estos sistemas son habituales en las redes corporativas, incluyéndose junto al cortafuegos o incluso con un servidor dedicado en zona especiales como una red DMZ (desmilitarizada). Estas soluciones se muestras eficaces a nivel corporativo, pero pierden visibilidad en las redes desde las que se conectan los trabajadores en sus lugares elegidos para el trabajo a distancia.

Los Sistemas de detección de Intrusos (IDS) (ref. 13) suelen integrarse con un cortafuegos, que quedan fuera del presente documento. El sistema operativo Microsoft Windows trae el Windows Defender Firewall (ref.14) que es un cortafuegos que puede ser controlado mediante políticas corporativas para generar reglas y restricciones en el tráfico de red de los equipos informáticos, como una parte de un producto general llamado Microsoft Defender, que incluye entre otros un antivirus, políticas de control parental, administración de la privacidad, pero no contiene un IDS asociado y su potencia es muy limitada por varios motivos: los índices de detección de malware son inferiores a los de muchos competidores, los controles parentales están limitados a Microsoft Edge, entre otros.

Existen soluciones de alto coste para prevenir intrusiones, pero suponen un incremento de costes no viable para todos los bolsillos. El presente trabajo pretender abordar la implementación de un prototipo de bajo coste.

1.2 Objetivos del Proyecto

El objetivo del presente trabajo consiste en explorar e implementar una herramienta que permita detectar actividad maliciosa y violaciones de políticas mediante el análisis del tráfico de red existente. La disponibilidad de este tipo de herramienta en el lugar de teletrabajo, permitiría reducir el riesgo de incidentes para la organización, que podría barajar adoptar este tipo de dispositivos para mejorar la Confidencialidad, Integridad y Disponibilidad de sus activos.

El objetivo del presente trabajo persigue explorar una opción de bajo coste que permita aumentar la visibilidad de los problemas de seguridad en los hogares y lugares destinados al trabajo a distancia. Para ello la meta es conseguir un prototipo de IDS (*Intrusion Detection System*) que ayude a mejorar la seguridad en el entorno familiar.

A continuación, se describen los objetivos principales del presente trabajo:

1.2.1 Objetivos de investigación

A continuación, se ofrece un listado de los objetivos principales de esta investigación:

- Explorar los diferentes tipos de IDS que pueden interesar en el contexto del teletrabajo.
- Explorar los productos de referencia.
- Investigar la forma de implementar un sistema IDS de bajo coste para una red doméstica y su uso para el entorno doméstico.
- Explorar su uso y escalabilidad/integración en el ámbito empresarial, y evaluación de las disposiciones en materia de privacidad para cumplir la regulación existente.

1.2.2 Objetivos de Implementación

A continuación, se ofrece un listado de los objetivos principales de implementación:

- Aprendizaje sobre instalación y configuración de suricata.
- Aprendizaje sobre el stack ELK y su configuración, así como explorar opciones SaaS.
- Revisar los resultados y analizar el valor de la información obtenida.
- Optimización de la configuración.

1.2.3 Objetivos de Entrega

A continuación, se ofrece un listado de los principales entregables previstos:

- Desarrollar la documentación usando el formato y fechas de PEC.
- Video y Memoria del trabajo.

1.3 Descripción de la Metodología

La metodología escogida se basa en Design Thinking (ref.15). Fue inicialmente popularizada por la firma de Sillicon Valley **Ideo**. En los años 80, Design Thinking se aplicaba principalmente al diseño de productos físicos. Pero posteriormente ha ido ganando terreno para usarse en el diseño de servicios y experiencias de usuario.

Si bien el problema a resolver es técnico, el uso de esta metodología se ofrece como una manera para empatizar con el usuario y definir los requisitos conforme a las necesidades detectadas. El objetivo es la obtención de un prototipo y evaluar los resultados.

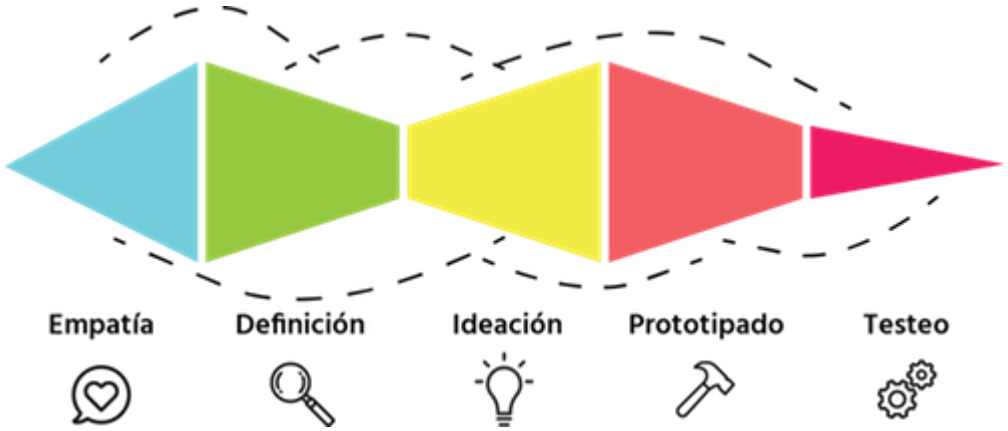

Ilustración 2: Design Thinking

Para cubrir la parte de prototipado, se recurrirá a Lean Startup Lean Startup y metodologías ágiles (ref.16, 17) para desarrollar un plan de cara a su lanzamiento para hogares.

Este último paso, sería tras realizar el Prototipado y medirlo, se procedería a pivotar la solución y generar la solución final.

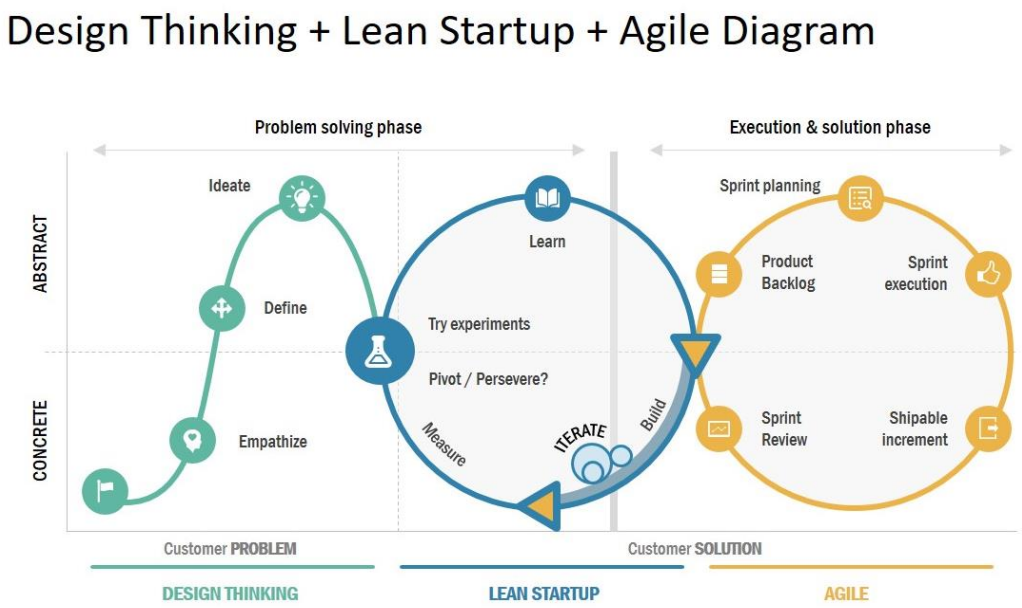

Ilustración 3: Design Thinking + Lean Startup + Agile Diagram (Ref. 15)

En la ilustración 5 se puede ver la relación entre fases / entregas y la metodología propuesta, así como los objetivos perseguidos.

Fases	Metodología propuesta	Objetivo
Introducción	Empatizar con la unidad familiar media con teletrabajadores	Definición de Problemática
Investigación	Definir e Idear Prototipo	
Desarrollo	Crear Experimento / Minimum Viable Product Conclusiones Definir segundo Experimiento / Viable Product	Solución
Conclusiones	Solución para hogares	

Ilustración 4 Relación Entregas - Metodología

El proyecto tiene dos partes diferenciadas:

- Marco teórico y planificación de trabajo, objetivo e investigación cuyos resultados se cubren en la PEC 1 y 2.
- Parte práctica de desarrollo y experimentación en la PEC3 antes de la elaboración de la solución final y la elaboración de la memoria.

Con este mapeo cubrimos las fases en Design Thinking (ref.18):

- **Empatizar**: comenzamos con una profunda comprensión de las necesidades de los usuarios implicados en la solución que estamos buscando y también de su entorno.
- **Definir**: se revisa la información de la fase anterior y se define qué capacidades tendrá la solución a implementar.
- **Idear**: se trata de albergar la posibilidad de generar diferentes opciones. No quedarse con la primera idea que se ocurra.
- **Prototipar**: se trata de construir un modelo rápido que nos a ayudará a dar forma a lo que hasta ahora era una idea o un concepto. Para ello se usará el concepto de Producto Mínimo Viable (MVP, Minimum Viable Product) de Lean Start y se realiza su implementación con Scrum, generando un backlog de prototipo y se definirá el incremento de producto a obtener tras sucesivos sprints de desarrollo.
- **Testear**: por último, se testea en un entorno objetivo hacia el que se orienta la solución que estamos desarrollando. Con el feedback, se generan las conclusiones.
- **Medir**: se trata de revisar los resultados obtenidos con las fases de Design Thinking.
- **Aprender**: revisar qué funciona, qué puede mejorarse y definir solución final para la implantación.

1.4 Planificación del Trabajo

					Días	Horas
Planificación Inicial						
Tarea 1	Proposito del trabajo	100%	17-2-21	19-2-21	2	6
Tarea 2	Orientación de la metodología	100%	19-2-21	24-2-21	5	15
Tarea 3	Cronograma y Recursos	100%	24-2-21	27-2-21	3	9
Tarea 4	Análisis de Riesgos	100%	27-2-21	1-3-21	2	6
Tarea 5	Entrega del Plan de Trabajo	100%	2-3-21	2-3-21	0	0
Investigación						
Tarea 1	Problemática Hogares	100%	2-3-21	6-3-21	4	12
Tarea 2	Investigación IDS y Tipos	100%	6-3-21	10-3-21	4	12
Tarea 3	Comparativa de Productos	100%	10-3-21	14-3-21	4	12
Tarea 4	Definición Requisitos y Diseño	100%	14-3-21	29-3-21	15	45
Tarea 5	Entrega de Seguimiento	100%	30-3-21	30-3-21	0	0
Desarrollo						
Tarea 1	Definición Pruebas de rseultados	100%	30-3-21	2-4-21	3	9
Tarea 2	Configuración	100%	2-4-21	15-4-21	13	39
Tarea 3	Implementación del Prototipo	100%	15-4-21	25-4-21	10	30
Tarea 4	Verificació de Resultados	100%	25-4-21	26-4-21	1	3
Tarea 5	Entrega de Resultados del Prototipo	100%	27-4-21	27-4-21	0	0
Conclusiones						
Tarea 1	Configuración Final	5%	27-4-21	30-4-21	3	9
Tarea 2	Medición de Resultados	0%	30-4-21	10-5-21	10	30
Tarea 3	Conclusiones	0%	10-5-21	16-5-21	6	18
Tarea 4	Consideraciones de Privacidad	0%	16-5-21	20-5-21	4	12
Tarea 5	Creación de Video	0%	21-5-21	25-5-21	4	12
Memoria y Video						
Tarea 1	Borrador de Memoria	100%	17-2-21	2-3-21	13	13
Tarea 2	Detalle Investigación	90%	2-3-21	30-3-21	4	4
Tarea 3	Datalle Desarrollo	75%	30-3-21	27-4-21	4	4
Tarea 4	Detalle Conclusiones y Versión Final	0%	27-4-21	25-5-21	15	15
Tarea 5	Entrega	0%	1-6-21	1-6-21	0	0
				Totales	129	315

Ilustración 5 Definición de Tareas

1.5 Planificación Temporal

Implementación de un Sistema de Detección de Intrusos (IDS) mediante la inspección de tráfico de la red

TFM – Master Ciberseguridad y Privacidad
Miguel Ángel Guinea

Inicio del proyecto:
Semana para mostrar: 15

TAREA	ASIGNADO A	PROGRESO	INICIO	FIN
Planificación Inicial				
Tarea 1	Proposito del trabajo	100%	17-2-21	19-2-21
Tarea 2	Orientación de la metodología	100%	19-2-21	24-2-21
Tarea 3	Cronograma y Recursos	100%	24-2-21	27-2-21
Tarea 4	Análisis de Riesgos	100%	27-2-21	1-3-21
Tarea 5	Entrega del Plan de Trabajo	100%	2-3-21	2-3-21
Investigación				
Tarea 1	Problemática Hogares	100%	2-3-21	6-3-21
Tarea 2	Investigación IDS y Tipos	100%	6-3-21	10-3-21
Tarea 3	Comparativa de Productos	100%	10-3-21	14-3-21
Tarea 4	Definición Requisitos y Diseño	100%	14-3-21	29-3-21
Tarea 5	Entrega de Seguimiento	100%	30-3-21	30-3-21
Desarrollo				
Tarea 1	Definición Pruebas de rseultados	100%	30-3-21	2-4-21
Tarea 2	Configuración	100%	2-4-21	15-4-21
Tarea 3	Implementación del Prototipo	100%	15-4-21	25-4-21
Tarea 4	Verificació de Resultados	100%	25-4-21	26-4-21
Tarea 5	Entrega de Resultados del Prototipo	100%	27-4-21	27-4-21
Conclusiones				
Tarea 1	Configuración Final	5%	27-4-21	30-4-21
Tarea 2	Medición de Resultados	0%	30-4-21	10-5-21
Tarea 3	Conclusiones	0%	10-5-21	16-5-21
Tarea 4	Consideraciones de Privacidad	0%	16-5-21	20-5-21
Tarea 5	Creación de Video	0%	21-5-21	25-5-21
Memoria y Video				
Tarea 1	Borrador de Memoria	100%	17-2-21	2-3-21
Tarea 2	Detalle Investigación	90%	2-3-21	30-3-21
Tarea 3	Datalle Desarrollo	75%	30-3-21	27-4-21
Tarea 4	Detalle Conclusiones y Versión Final	0%	27-4-21	25-5-21
Tarea 5	Entrega	0%	1-6-21	1-6-21

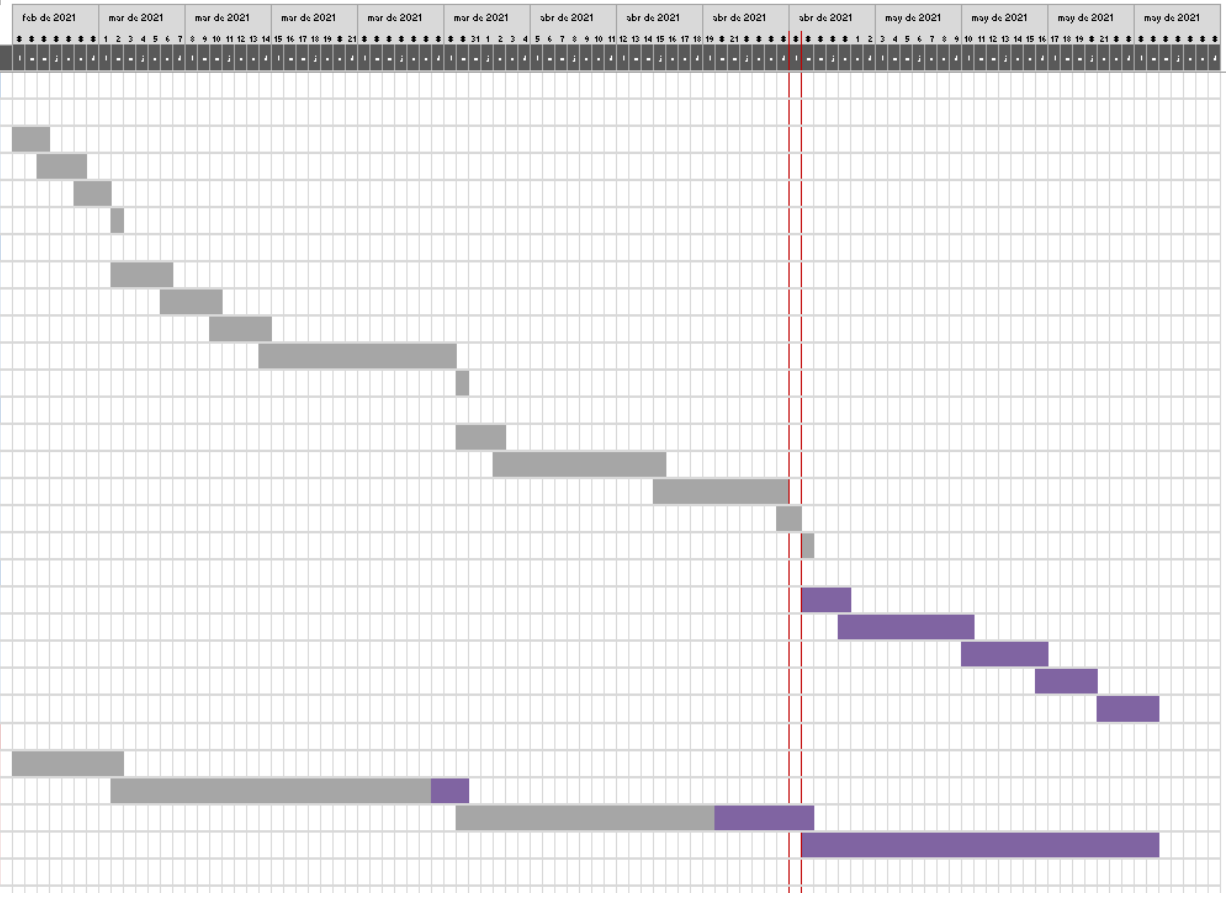

Ilustración 6 Diagrama GANTT del proyecto

1.6 Recursos y presupuesto del proyecto

Los recursos necesarios para la realización de este trabajo son:

- Router internet con ISP.
- Raspberry PI 3.
- Laptop con tráfico real.
- Móvil con tráfico real.
- Smart TV con tráfico real.
- Cuenta de cloud entorno ELK (logz.io.

El presupuesto necesario para cubrir el proyecto:

Activo	Coste (€)	Comentario
Conexión Internet (48,9€/mes)	97,8	Pepephone 600Mb
Raspberry PI3	74,99	Raspberry
Cuenta Cloud ELK	0	Opción free, Logz.io ofrece opción Pro (retención 7 días) a $1,08 por gigabyte indexado
Persona (315h) /13€ hora	4095	Asumiendo un sueldo de 1500€, 160h/mes con un 30% de sueldo base en gastos de Seguridad social
Amortización Activos (Smart TV, Móvil, Laptop)	51,73	Calculado sobre el costo de los equipos en función de su vida útil y el número de horas dedicadas al proyecto.
Electricidad	33,03	Coste de las horas relacionadas con el proyecto
Total	5352,55	

Ilustración 7 Tabla de costes

El presupuesto no incluye las horas de seguimiento del personal docente, así como el tribunal. Los costes de Smart TV, Móvil o Laptop por ser activos ya existentes, sólo se referencia el coste de amortización en función de su vida útil.

1.7 Análisis de riesgos del proyecto

Se usa un sistema para seguir los riesgos del proyecto y sus mitigaciones

CATEGORIZACIÓN		CLASIFICACIÓN				RESPUESTA		
NOMBRE	CATEGORÍA	PUNTUACIÓN DE RIESGO	IMPACTO	PROBABILIDAD	CLASIFICACIÓN	MITIGACIÓN	DISPARADOR	NUEVA CALIFICACIÓN
Complejidad	Implementación	30	100	50	ALTA	MVP	Fase Definición	MEDIA
Indefinición	Implementación	40	80	50	ALTA	Research Validado	Fase Definición	MEDIA
Elección de producto equivocada	Implementación	60	80	50	ALTA	Prototipado Rápido	Fase Implementación	MEDIA
Logz.io Filebeat ARM	Implementación	60	80	60	ALTA	Prototipado Rápido	Fase Implementación	MEDIA

Ilustración 8 Riesgos del Proyecto y Respuestas

En el momento de arranque del proyecto, se registran los riesgos principales que pueden impactar en las fechas de entrega, durante la ejecución del proyecto se ha registrado adicionalmente un riesgo más relacionado con los problemas de implementación.

Se detallan los riesgos detectados y su mitigación:

- **Complejidad**: implementar un IDS y como realizar su integración, requiere un trabajo de investigación con muchas horas de ensayo. **Como mitigación** se propone usar Design Thinking-Lean Startup. Esta metodología nos permite definir un Minimum Viable Product como prototipo y a partir de ahí construir el resto.
- **Indefinición**: el alcance del proyecto puede ser muy grande, y las posibilidades de investigación son altas, usar el MVP como director del trabajo a realizar puede ser una buena ayuda para **mitigar** este riesgo y posteriormente generar el resto en la fase de optimización.
- **Elección de producto**: De los tres finalistas Snort, Suricata y BRO, se opta por Suricata, por tener muy buen balanceo entre características e información existente. Suricata puede sufrir pérdida de paquetes debido a problemas de performance cuando la demanda es alta, por lo que se identifican capacidades de mejora en cuanto a performance en caso de ser necesario. El diseño o intención de uso, de usar un IDS por cada subred, permite escalarlo y enviando los resultados a un ELK central, nos aseguramos la escalabilidad de la solución. **Como mitigación**, se plantea cubrir una Prueba de Concepto para evaluar la implementación y el producto.

- Filebeat ARM: Elastic no tiene un artefacto para la arquitectura ARM de Raspberry. **La mitigación** consiste en la búsqueda de experiencias de usuarios en situación similar. Se plantea la construcción del artefacto en la raspberry. Si bien en un entorno productivo, el soporte de Elastic sobre incidencias en este artefacto sería inexistente.

1.8 Revisión del estado del arte

Antes de la detección de intrusiones, existían las auditorías de seguridad. La auditoría es el proceso de generar, almacenar y revisar eventos de un sistema cronológicamente.

La detección de intrusiones es el fruto de la aplicación del Procesamiento Electrónico de Datos (EDP) a las auditorías de seguridad, utilizando mecanismos de identificación de patrones y métodos estadísticos. Es una parte imprescindible en las modernas tecnologías de seguridad de redes.

James P. Anderson fue el primero en describir el concepto en un estudio encargado por las Fuerzas Aéreas de los EEUU en un informe de 1972. Este informe, que se acabó conociendo ampliamente como el informe Anderson, definió la agenda de investigación en seguridad de la información por al menos una década. A partir de 1980, aparecieron varios sistemas, pero no fue hasta inicios de los 90, tras los daños provocados por el famoso gusano de Internet de 1988, cuando se combinaron la monitorización de hosts y red.

Muchos IDSs se basan en el modelo propuesto por Dorothy Denning en 1987. Ella presentó un modelo de detección de intrusiones en tiempo real capaz de detectar múltiples formas de abuso informático. El modelo se basaba en la hipótesis de que las violaciones de seguridad pueden detectarse mediante la monitorización de los registros de auditoría de un sistema en busca de patrones anormales de uso del sistema. El modelo incluía perfiles para representar el comportamiento de los sujetos con respecto a los objetos en términos de métricas y modelos estadísticos, y reglas para adquirir conocimientos sobre este comportamiento a partir de los registros de auditoría y para detectar comportamientos anómalos. El modelo era independiente de cualquier sistema particular, entorno de aplicación, vulnerabilidad del sistema o tipo de intrusión, proporcionando así un marco para un sistema experto de detección de intrusiones de propósito general.

1.8.1 HIDS y NIDS

A continuación, se detalla las principales características de estos dos tipos de IDS:

- HIDS (Host IDS): intenta detectar modificaciones y rastros de actividades de un intruso en el equipo atacado.
- NIDS (Network IDS): es un IDS basado en red, detectando ataques a todo el segmento de red. Su interfaz debe funcionar en modo promiscuo. En la ilustración de la página anterior, vemos la referencia a WIDS, como un conjunto específico, refiriéndose a Wireless IDS.
- Híbridos: combinan información del host y de red y la detección en ambos frentes.

1.8.2 Diferencia entre basado en firmas y basado en anomalías

A su vez, los IDS pueden operar de diferentes maneras:

- IDS basado en firmas (SBIDS, Signature-based Intrusion Detection Systems): supervisan todos los paquetes de la red y los comparan con la base de datos de firmas (patrones de ataque). Funciona de forma similar a un antivirus.
- IDS basado en anomalías (ABIDS, Anomaly-based Intrusion Detection System): estos IDS monitorizan el tráfico de red y lo comparan con una línea base establecida para determinar que se considera normal y que no.

1.8.3 Diferencia entre IDS pasivos y reactivos

A diferencia de los sistemas pasivos, donde sólo se obtiene alerta, los sistemas reactivos pueden plantear medidas / acciones para evitar la actividad sospechosa, por ejemplo, reprogramando el cortafuegos con reglas específicas para no aceptar tráfico de la IP de origen.

Como primer objetivo se procederá a revisar qué herramientas IDS están consideradas dentro del mercado y se identificar principales diferencias.

1.8.4 IDS disponibles en el mercado

(ref. 20):

Nombre	Plataforma	Tipo	Funcionalidad
Bro 	Unix, Linux, Mac-OS	NIDS	Registro y análisis del tráfico, Proporciona visibilidad de los paquetes, motor de eventos, guiones de políticas, Capacidad para supervisar el tráfico SNMP, Capacidad de seguimiento de la actividad FTP, DNS y HTTP. Bro ha pasado a llamarse Zeek.
OSSEC 	Unix, Linux, Windows, Mac-OS	HIDS	HIDS de código abierto de uso gratuito, Capacidad para detectar cualquier alteración en el registro en Windows, Capacidad de monitorear cualquier intento de llegar a la cuenta raíz en Mac-OS, Los archivos de registro cubiertos incluyen los datos de los servidores de correo, FTP y web.
Snort 	Unix, Linux, Windows	NIDS	Sniffer de paquetes, registrador de paquetes, Inteligencia de amenazas, bloqueo de firmas, Actualizaciones en tiempo real de las firmas de seguridad, informes detallados, Capacidad para detectar una variedad de eventos, incluyendo huellas digitales del sistema operativo, sondas SMB, ataques CGI, ataques de desbordamiento de búfer y escaneos de puertos sigilosos.
Suricata 	Unix, Linux, Windows, Mac-OS	NIDS	Recoge datos en la capa de aplicación, Capacidad para supervisar la actividad de los protocolos en niveles inferiores como TCP, IP, UDP, ICMP y TLS, seguimiento en tiempo real de aplicaciones de red como SMB, HTTP y FTP, Integración con herramientas de terceros

Nombre	Plataforma	Tipo	Funcionalidad
			como Anaval, Squil, BASE y Snorby, módulo de scripting incorporado, utiliza métodos basados en firmas y anomalías, Arquitectura de procesamiento inteligente.
Security Onion 	Linux, Mac-OS	HIDS, NIDS	Completa distribución de Linux centrada en la gestión de registros, Monitoreo de seguridad empresarial y detección de intrusos, Se ejecuta en Ubuntu, integra elementos de varias herramientas de análisis y frontend incluyendo NetworkMiner, Snorby, Xplico, Sguil, ELSA y Kibana, Incluye funciones HIDS también, un sniffer de paquetes realiza el análisis de la red, Incluye bonitos gráficos y diagramas.

Ilustración 9: Tabla de Productos IDS

Revisando el escenario inicial, se consideran los siguientes aspectos:

- IDS existente en el mercado: la definición deberá revisar a fondo Bro IDS, Snort y Suricata y decidir cuál de los tres productos aplicar como base.
- Ser capaz de retransmitir resultados a una interfaz visual accesible por analista IT de la organización.
- Usar un método basado en firmas y anomalías actualizable.
- Ser capaz de recomendar, automatizar algunas acciones.

1.8.5 Snort vs Suricata vs Zeek (BroIDS)

El objetivo es completar una comparativa entre NIDS puros, que permita definir cuál de las tres opciones es la más indicada para usarla como base:

Si comparamos Snort y Suricata, el primero fue creado por Cisto en 1998, y tiene una arquitectura single-threaded. Ha sido el standard de facto como IDS engine por muchos años, con una basta comunidad de usuarios y suscriptores. Sus principales ventajas son:

- Escalabilidad: Snort puede desplegarse con éxito en cualquier entorno de red.
- Flexibilidad y facilidad de uso: Snort puede funcionar en varios sistemas operativos, como Linux, Windows y Mac OS X.
- En vivo y en tiempo real: Snort puede ofrecer información de eventos de tráfico de red en tiempo real.
- Flexibilidad en el despliegue: Hay miles de maneras en las que Snort puede ser desplegado y una miríada de bases de datos, sistemas de registro y herramientas con las que puede trabajar.
- Rapidez en la detección y respuesta a las amenazas de seguridad: Utilizado junto con un cortafuegos y otras capas de la infraestructura de seguridad, Snort ayuda a las organizaciones a detectar y responder a los crackers de sistemas, gusanos, vulnerabilidades de la red, amenazas a la seguridad y a los que abusan de las políticas que pretenden derribar la red y los sistemas informáticos.
- Motor de detección modular: los sensores de Snort son modulares y pueden supervisar varios equipos desde una ubicación física y lógica. Snort puede colocarse delante del cortafuegos, detrás del cortafuegos, al lado del cortafuegos y en cualquier otro lugar para vigilar toda una red.

Como resultado, las organizaciones utilizan Snort como solución de seguridad para averiguar si hay intentos no autorizados de hackear la red o si un hacker ha conseguido acceso no autorizado al sistema de la red.

Suricata es una iniciativa de código abierto promovida por la Open Information Security Foundation (OISF) en 2009. Se ha convertido en una alternativa moderna cuyas principales ventajas son:

- Un motor de código abierto: El poder de la comunidad funciona bien en las defensas de seguridad informática, ya que una comunidad es más eficaz que una sola organización para captar las características de las amenazas emergentes.

- Multihilo: Una arquitectura multihilo permite que el motor aproveche las arquitecturas de múltiples núcleos y multiprocesadores de los sistemas actuales.
- Admite la reputación de IP: Al incorporar la reputación y las firmas en su motor, Suricata puede marcar el tráfico de fuentes malas conocidas.
- Detección automática de protocolos: Los preprocesadores identifican automáticamente el protocolo utilizado en un flujo de red y aplican las reglas adecuadas, independientemente del puerto numérico. La detección automatizada de protocolos también evita los errores del usuario, que son los más comunes.

Aunque Suricata es todavía un producto nuevo y menos extendido en comparación con Snort, la tecnología está ganando impulso entre todas las empresas y usuarios de TI. El mayor rendimiento, la compatibilidad nativa con IPv6, la detección de anomalías estadística de múltiples modelos, la aceleración de la GPU, la reputación de IP, los umbrales de puntuación, la regex de muy alta velocidad y la escalabilidad son algunos de los principales puntos de venta de Suricata (ref 26).

En cuanto a Suricata y Zeek no son productos incompatibles (ref. 27). En comparación, Zeek se diseñó inicialmente para ser una navaja suiza para la supervisión de metadatos de la red. Supervisa los flujos de tráfico y produce registros que registran todo lo que entiende sobre la actividad de la red y otros metadatos que son útiles para analizar y comprender el contexto del comportamiento de la red. Gran parte de los metadatos que produce Zeek sólo estaban disponibles anteriormente en los datos de captura de paquetes (PCAP). Los metadatos también pueden ser buscados, indexados, consultados y reportados en nuevas formas que no eran posibles con PCAP. El lenguaje de programación de Zeek, con una estructura similar a la de C++ (snort y suricata están desarrollados usando C), puede utilizarse para calcular estadísticas numéricas, realizar la coincidencia de patrones de expresiones regulares y personalizar la interpretación de los metadatos según las necesidades específicas de una organización.

En resumen, lo ideal sería que las organizaciones confiaran en Suricata para identificar rápidamente los ataques en los casos en que las firmas están disponibles y utilizaran Zeek para proporcionar los metadatos y el contexto necesarios para clasificar con éxito las alertas de Suricata y crear líneas de tiempo completas de todo el panorama de amenazas. La combinación de Suricata y Zeek también es muy eficaz para la caza de amenazas.

Por ejemplo, Suricata puede enviar una alerta de que un sistema está comprometido y el incidente y las conexiones antes y después de que se produzca son registrados por Zeek y pueden ser analizados para determinar si otras comunicaciones de la red refuerzan o ayudan a explicar el incidente.

Para la detección de amenazas de espectro completo, se necesitan las capacidades que proporcionan Suricata y Zeek, además de la detección de malware basada en ML. En este contexto, aparecen soluciones como la plataforma de seguridad de red de Bricata (Ref. 28) que combina Suricata, Zeek, la detección de malware basada en ML y los datos completos de PCAP en un solo lugar para ofrecer capacidades completas de detección y respuesta de red para entornos en la nube, híbridos y locales.

1.8.6 Futuro

Los IDS encajan en una propuesta de defensa organizada por capas. El trabajo a distancia puede incluir una capa adicional a tener en cuenta. La detección de intrusiones y la tecnología subyacente están todavía en desarrollo y lejos de llegar a su madurez. Algunas áreas donde se ha visto avance en los últimos años:

- Reducción de la dependencia con firmas en la detección.
- El crecimiento de la prevención de intrusiones.
- Avances en la correlación da datos y alertas.
- Avances en la determinación de las fuentes.
- Inclusión de funcionalidad forense en los IDS.

- Mejor uso de los honeypots, o entornos seguros donde un intruso puede relevar su comportamiento sin afectar a activos reales.

Frente a sistemas de detección basados en firmas como Snort o Suricata, las técnicas de Data Mining (decisión tres, SVM, …) y Machine Learning se consideran la evolución necesaria para identificar amenazas y ataques desconocidos o zero day attacks. Muchos ataques son parcialmente variados para evitar el reconocimiento por reglas, de tal modo que los sistemas IDS sean capaces de identificar con mayor grado de certeza amenazas reales. Estás técnicas permiten clasificar anomalías sin conocer la firma exacta del ataque.

Las técnicas de Machine Learning necesitan una cantidad ingente de datos de entrenamiento para que el modelo generado sea capaz de predecir aspectos relacionados con el futuro. Hay técnicas que se basan en:

- aprendizaje supervisado: juegos de datos de entrenamiento son facilitados para entrenar y generar el modelo.
- aprendizaje no supervisado: en este caso no hay un juego de datos predefinido, y en el caso del aprendizaje por refuerzo, los agentes son recompensados por su desempeño y en función de esa recompensa son capaces de reajustar su comportamiento.

Uno de los problemas clásicos en los IDS era el número de falsos positivos y alarmas que podían colapsar al personal responsable de la revisión de los casos. Las redes neuronales pueden ser una técnica muy eficiente a la hora de identificar ataques (ref.40) con menor número de falsos positivos.

Otra técnica para optimización viene del uso de Algoritmos genéticos (ref. 41) para la optimización de parámetros de configuración y clasificación de ataques de seguridad y generación de reglas.

2. Fase de Investigación

La fase de investigación se corresponde con los pasos: **Definir** e **Idear** en la metodología Design Thinking.

Como se comentó en párrafos anteriores, un IDS se basa en el análisis pormenorizado del tráfico de red, el cual, al entrar al analizador, es comparado con firmas de ataques conocidos, o comportamientos sospechosos. Los IDS suelen disponer de una base de datos de “firmas” de ataques conocidos.

Dos buenas herramientas para la fase de ideación son el Mapa de Actores y el Mapa de Empatía. El primero, nos ayuda a entender globalmente el contexto en el que está el trabajador/adulto responsable de cara a la tecnología y responsabilidades/peligro de uso, frente a otros actores. El segundo nos acerca a actores particulares. En este caso se usa el Mapa de Empatía para ofrecer el punto de vista de un empleado y del responsable de Seguridad.

2.1 Mapa de actores

El mapa de actores refleja en el centro de un hogar a los responsables que se consideran población activa y pueden estar desarrollando una actividad laboral, usando el teletrabajo o trabajo a distancia (las diferencias pueden revisarse en el Real Decreto 28/2020 de 22 de septiembre) para realizar las actividades.

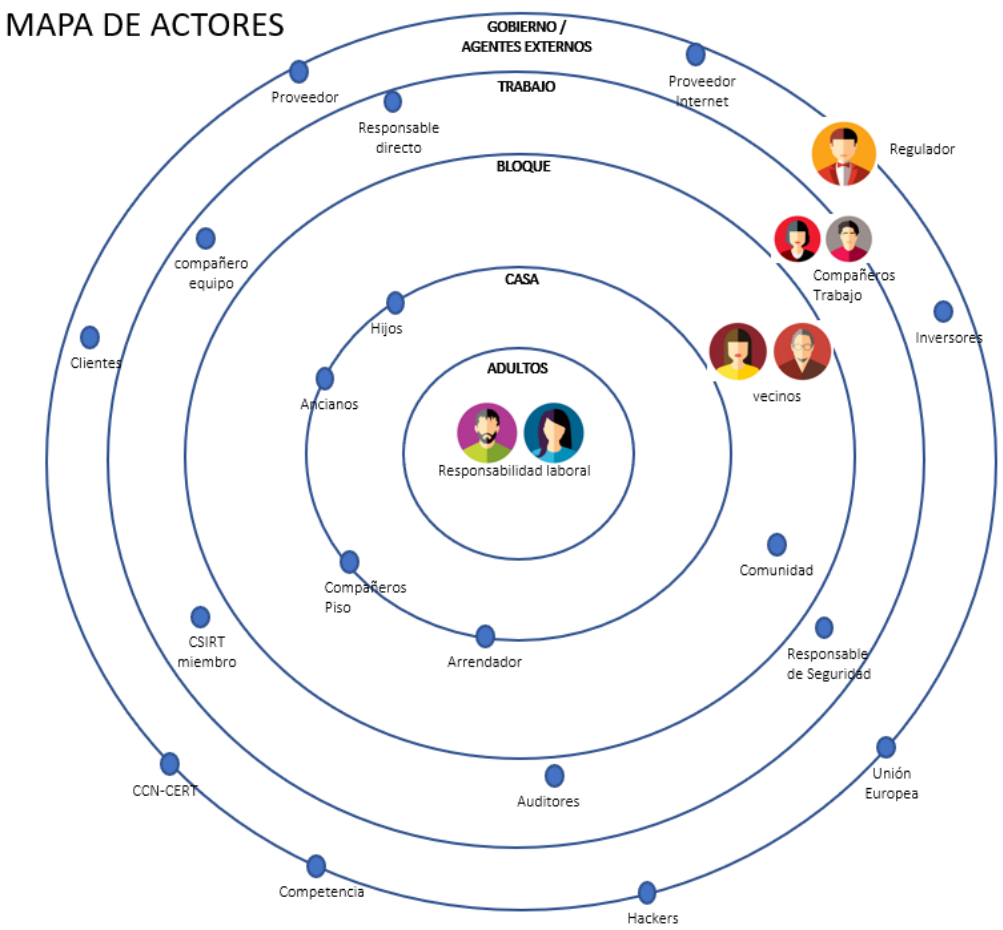

Ilustración 10 Mapa de Actores

En círculos concéntricos se identifican actores que pueden influir en la capacidad de estos actores principales para hacer su trabajo o que pueden requerir explicaciones del mismo, en relación con los incidentes de seguridad. En el hogar, tenemos al resto de familiares, compañeros de piso o arrendador que pueda ser el dueño del router y del contrato con el Proveedor de Acceso a Internet. que pueden conectarse a la misma red, incluso usando el mismo equipo.

Un nivel por encima, tenemos a los vecinos, que pueden robar wifi o nosotros conectarnos al suyo, véase ref.22, en el ámbito de la empresa tenemos al Responsable de Seguridad, encargado de las decisiones para satisfacer las necesidades de los sistemas de información en materia de seguridad de la información y los servicios. Además del CSIRT, Equipo de Respuesta a

Incidentes de Seguridad. Más allá, el proveedor de acceso a internet, el gobierno y las autoridades en materia de seguridad CCN-CERT que deben ser notificados en caso de incidente, y por último clientes e inversores de la empresa en la que los actores principales desarrollan su actividad laboral.

2.2 Mapa de Empatía

Los mapas de empatía son herramientas que nos ayudan a conocer más íntimamente las perspectiva de actores concretos.

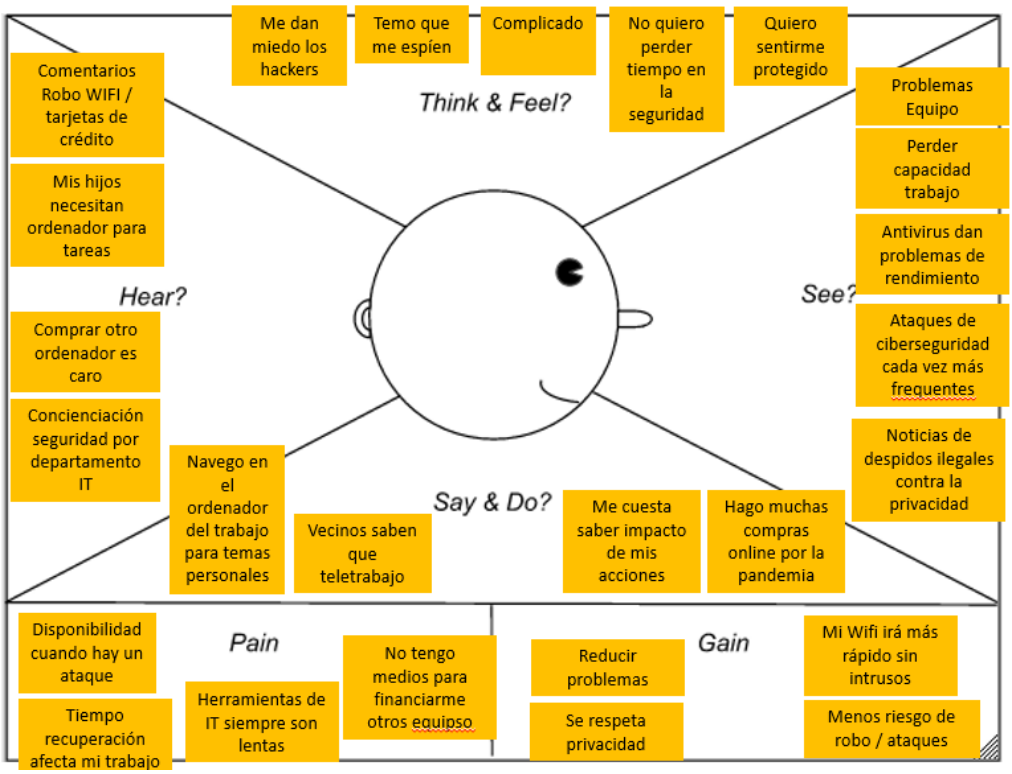

Ilustración 11 Mapa de Empatía: Empleado

Un/a empleado medio, tras las campañas de sensibilización y las noticias, tiene miedo de los hackers, pero necesita teletrabajar y a parte tiene familia y pocos ingresos como para poder permitirse varios equipos en la casa. Pero además está preocupado por que pueda sufrir ataques a la seguridad en su hogar. Puesto que en este caso, además de poner en riesgo los datos corporativos, pueden ponerse en riesgo, los datos más personales de los miembros de la familia (identificación, todo tipo de datos sensibles, fotografías, videos, datos

bancarios) con los que pueden ser extorsionados tanto ellos, como otros miembros de la unidad familiar.

En muchos casos, otros miembros que usan la red doméstica familiar, pueden ser propietarios de PYMEs que tengan los datos de sus negocios y establecimientos en ordenadores en la red local.

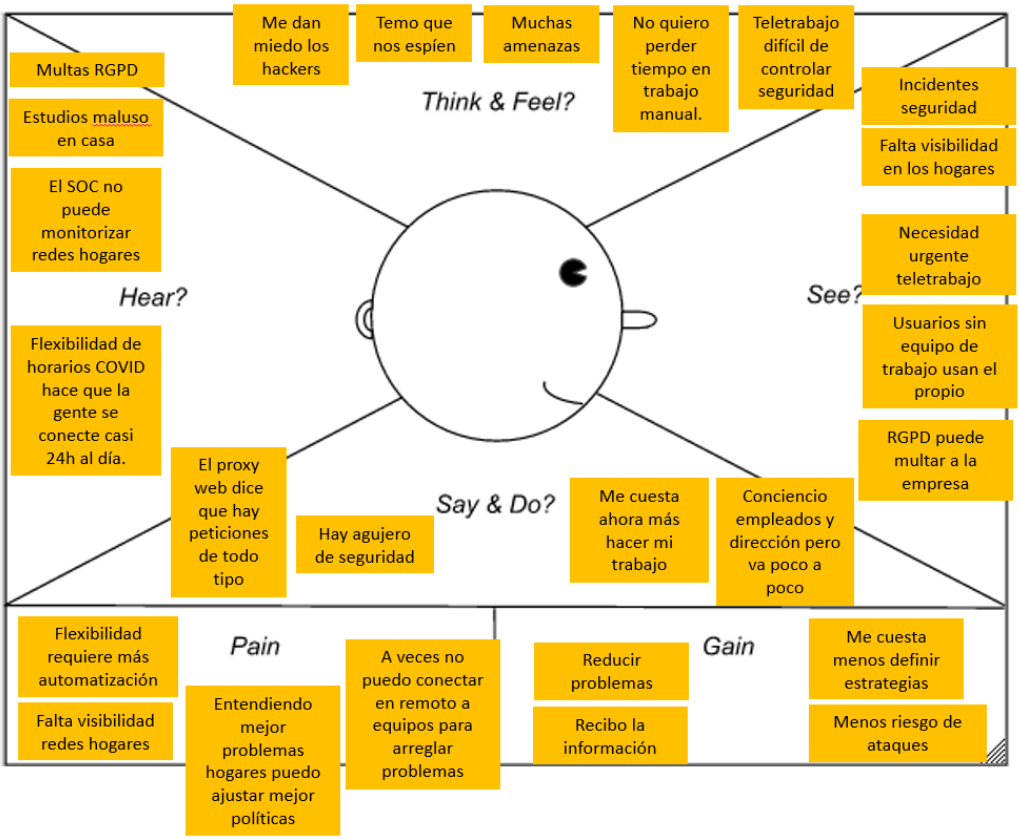

Ilustración 12 Mapa de Empatía: Responsable de Seguridad

El Responsable de Seguridad ve incrementarse sus problemas en materia de seguridad con el teletrabajo y para poder responder mejor frente a incidentes, necesita de herramientas eficaces que puedan automatizar la detección de incidentes y reportar esta información de manera automática para el ciclo de vida del incidente de seguridad por parte del equipo de gestión de incidentes.

El hecho de que un empleado pueda emplear su propio dispositivo (BYOD) o que pueda conectarse a una red wifi en la que puede haber otros actores con intenciones maliciosas, incrementa la posibilidad de que potenciales atacantes intenten localizar vulnerabilidades en el equipo del empleado y desde ahí, y

mediante VPN, ataquen los recursos corporativos. O en los casos en los que los empleados descarguen datos en sus ordenadores, las compañías tienen inclusos menos control. En muchas organizaciones estos recursos todavía incluyen el uso de RDP, recursos compartidos o directorio activo, que po ejemplo, sin las actualizaciones pertinentes para evitar el uso del protocolo SMBv1 (Server Message Block), permitirían a un atacante explotar la vulnerabilidad CVE-2017-0144, que fue empleada por el ransomware Wannacry para infectar más de 230000 PCs corriendo SSOO Microsoft Windows en 150 países. Un smartphone, la Tablet, nuestro Smart TV pueden ser puertas de entrada para estos atacantes.

Los datos indican que a mediados de 2020 se produjo un crecimiento a nivel global de los intentos de ataque al RDP y los accesos remotos en general. Un crecimiento que sólo en América Latina durante el tercer trimestre de 2020 significó un aumento del 141% de los intentos de ataque de fuerza bruta al RDP (ref. 37).

2.3 Requisitos

Tras usar diferentes técnicas de la fase de investigación de la metodología Design Thinking, establecemos una seria de requisitos que necesitamos tener en cuenta para diseñar una solución que cubra los problemas y preocupaciones en el escenario descrito. A gran nivel:

- **Requisito 1:** Foco en NIDS por las potenciales amenazas en un entorno de hogar con wifi, para identificar esas amenazas y alertar / mitigar su impacto en la compañía y en los activos del hogar, así como daños reputacionales tanto de la compañía como del empleado / empleada, además de prevenir los daños particulares que pueda sufrir tanto esta persona como otras personas del núcleo familiar.
- **Requisito 2:** Poder usarse en los hogares e integrarse fácilmente en ecosistema IT.
- **Requisito 3:** Revisar todo el tráfico de internet del empleado.
- **Requisito 4**: Respeta la privacidad sin registrar ningún dato personal.

- **Requisito 5:** Usar bien análisis por firmas o anomalías.

- **Requisito 6:** Tener una forma de exportar la información a herramientas gráficas.
- **Requisito 7:** Capacidad de automatización para acciones reactivas.
- **Requisito 8:** Poder usarse de puente entre analista de seguridad de la organización y los empleados.
- **Requisito 9:** Tener comunidad y buena documentación.
- **Requisito 10:** capacidad de integrar con más empleados e integrar los resultados a un nivel superior mediante herramientas en cloud que permitan la identificación y seguimiento de alertas de modo centralizado.

2.3 Diseño / Minimum Viable Product

Para proceder a desarrollar un prototipo y su testeo, debe definirse un primer Minimum Viable Product (MVP) que ayude a probar el concepto y que nos lleve a la fase de optimización pare refinar la solución. Este MVP cubre los requisitos clave definidos en el apartado anterior, y permite iterar a los siguientes Viable Products 1, 2, …

La idea básica de MVP es probar a montar un NIDS en un artefacto de bajo coste, por tanto:

- Se proponer la puesta en funcionamiento dispositivo raspberry pi 3.
- Obligar a que todo el tráfico de dispositivos pase a través del dispositivo raspberry para la monitorización del tráfico de internet.
- Instalar sistema de monitorización del tráfico (NIDS).
- Prueba del sistema con incidente causado por un dispositivo conectado al sistema.

2.3.1 Funcionamiento Raspberry PI3+

Raspberry es un ordenador del tamaño de una tarjeta de crédito que puede conectarse a un monitor y teclado y en el que se puede instalar un Sistema Operativo tipo Linux. Suele utilizarse a menudo en relación a la IoT “Internet of Things” para generar todo tipo de nuevos dispositivos con capacidades de computación: robótica, domótica, etc. Tiene un consumo muy limitado de electricidad en comparación con un equipo informático.

El proyecto fue ideado en 2006 y lanzado al mercado en 2012 (véase https://www.raspberrypi.org/). Fue ideado por un equipo de investigación de la Universidad de Cambridge para facilitar a los niños el aprendizaje de las ciencias de la computación.

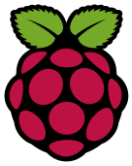

Ilustración 13 Logotipo de Raspberry

En Ref.23 se describen los pasos para instalar Raspian OS en un raspberry PI 3 como la que se utiliza en el actual trabajo.

Ilustración 14 Raspberry PI 3.

Se requiere una tarjeta Micro SD donde pueda volcarse la imagen descargada del sistema operativo, previamente descargado de internet. Una vez copiado y hecho auto arrancable, se procede a completar la instalación en el dispositivo.

A continuación, se ofrece una descripción de las características de las versiones PI 3 en cuanto a procesador y memoria:

	Raspberry Pi 3 Model B	**Raspberry Pi 3 Model B+**
Procesador	Broadcom BCM2837, Cortex-A53 (ARMv8) 64-bit SoC	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
Frecuencia de reloj	1,2 GHz	1,4 GHz
Memoria	1GB LPDDR2 SDRAM	1GB LPDDR2 SDRAM

Ilustración 15 Características Raspberry PI 3.

2.3.2 ELK

ELK es la sigla para tres proyectos open source (ref.32): ElasticSearch, Logtash y Kibana.

- ElasticSearch: es un motor de búsqueda y analítica.

- Logtash: es un pipeline de procesamiento de datos del lado del servidor que ingesta datos de una multitud de fuentes simultáneamente, los transforma y los envía.

- Kibana: permite a los usuarios visualizar los datos en cuadros y gráficos con ElasticSearch.

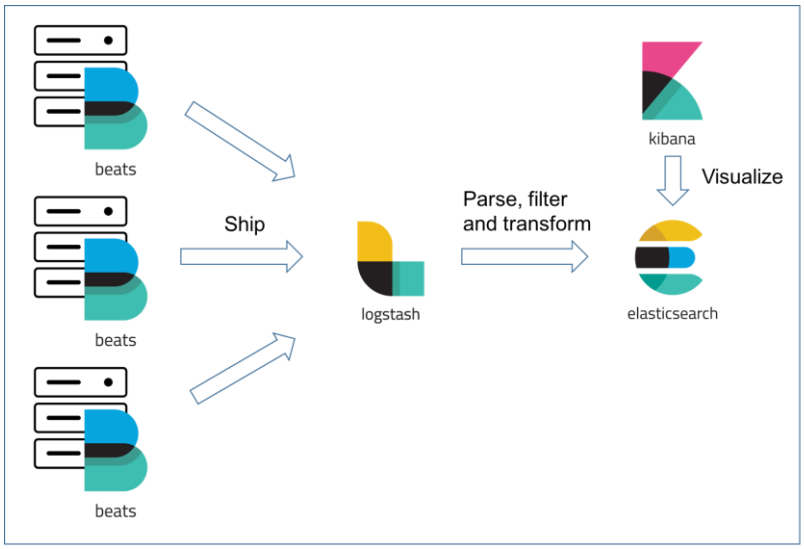

Ilustración 16 Modelo de colaboración ELK

Existe un producto adicional que es Beats (ref. 35), son agentes ligeros de Elastic. El más relevante es filebeat. Se pueden utilizar para recopilar cierta información y tras su transformación y enriquecimiento, podemos usar, para ingestarla en Logstash o ElasticSearch.

2.3.3 Elastic Search

ElasticSearch (ref. 33) permite indexar y analizar grandes cantidades de datos en tiempo real de manera distribuida. Permite almacenar documentos (estructurados o no) e indexar todos los campos de estos documentos en tiempo real. Los conceptos más importantes a tener en cuenta son:

- Cluster: es un conjunto de nodos que mantienen la información de manera distribuida e indexada.
- Nodo: es un servidor que forma parte del cluster, almacena la información y ayuda con las tareas de indexión y búsqueda del cluster.
- Index: es una colección de documentos que tienen características similares.
- Sharding y Réplicas: cuando la información en indexación va más allá del límite de una máquina, el sharding nos permite dividir los índices en distintas piezas, ofreciéndonos la posibilidad de escalar horizontalmente (añadiendo más máquinas), además de paralelizar y distribuir las distintas operaciones sobre los índices. La replicación nos ofrece mecanismo de recuperación en caso de fallo de una de las máquinas.

2.3.4 Logtash

Esta herramienta (Ref. 34) nos permite gestionar los logs de nuestras aplicaciones de manera que podamos usarla para recolectar, parsear u guardar los logs para búsquedas posteriores. Esta desarrollada en java, por lo que necesita la JVM (Java Virtual Machine) para ejecutarse. Es una herramienta tipo ETL (Extract, Transform and Load).

Con Logstash podemos, por ejemplo, consultar una base de datos cierta información, también otros clústeres de elastic y enriquecer el dato para ingestarlo en ElasticSearch.

Mediante la definición de pipelines de procesos, Logstash nos permitirá recoger eventos desde multitud de orígenes de datos (inputs), su posterior

transformación, enriquecimiento y normalización (filters) y su distribución a distintos destinos (outputs).

Para evitar pérdidas de información, es posible activar la característica de cola persistente (PersistentQueue). Esto permite, además, poder asumir picos puntuales en el número de eventos recibidos. Igualmente, otro tipo de colas son las DeadLetterQueues, que permiten tener control sobre los eventos enviados a un destino del cual se recibe un código de error, permitiendo almacenar los eventos en disco para una revisión posterior y evitar así su pérdida.

2.3.5 Kibana

Es la herramienta de visualización y reporte, mediante la cual podremos tener dashboards con visualizaciones interactivas sobre nuestros datos. Pero no sólo tiene esta funcionalidad, también posibiliza:

- Una administración del cluster de ElasticSearch más intuitiva a través de una interfaz gráfica.
- Medir el rendimiento de nuestras aplicaciones mediante el módulo APM.
- Tener una consola centralizada interactiva para análisis de seguridad para SIEM (Security Information and Event Management) (ref. 36).
- Activar funcionalidad de machine learning para por ejemplo detección de anomalías en la infraestructura.

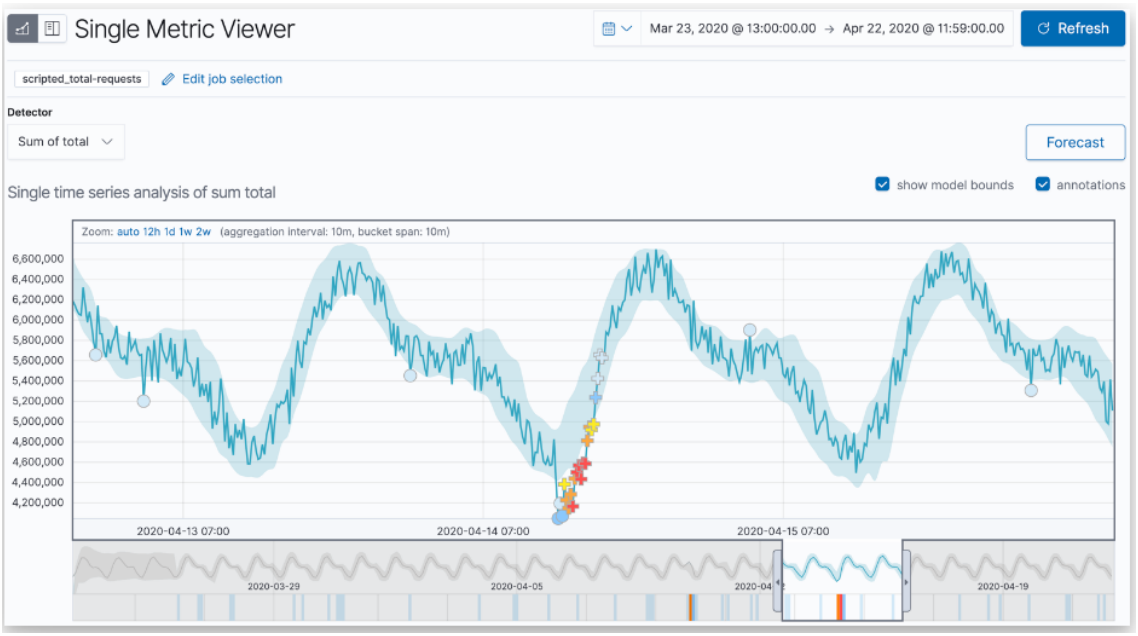

Ilustración 17 Ejemplo de exploración de anomalía en Single Metric Explorer

2.3.6 Beats

El caso más común es para la recogida de logs de equipos remotos y su centralización en los dos productos mencionados. Es capaz de gestionar cortes de comunicación, dispone de control de carga que le permite reudcir la ratio de envío en casos de saturación en Logstash.

Elastic ha introducido multitud de modulos especializados que permiten gestionar más fácilmente los tipos específicos de logs (Apache, MySQL, Nginx, Suricata, etc), optimizando la ingesta a través de templates optimizados.

Otros Beats comunes son:

- Metricbeat: que se especializa en recoger métricas de servicios y sistemas.
- Packetbeat con el que podremos analizar paquetes de red en tiempo real (latencias, tiempos de respuesta, errores, tendencias, …).

- Winlogbeat: que analiza los registros de eventos de log de nuestro parque de Windows.
- Auditbeat: que nos permite recoger datos de auditoría, monitorizando procesos y la actividad de usuario en tiempo real. En Linux se integra con el framework de Audit.
- Heartbeat: con él podremos monitorizar el tiempo de actividad y de respuesta de nuestros endpoints, mediante la configuración centralizada de los mismos.

2.3.7 Esquema conectividad para enrutamiento tráfico de la red local

Para obligar a realizar todo el paso del tráfico de internet en el hogar a través del dispositivo raspberry (Requisito 3), se analizan dos maneras:

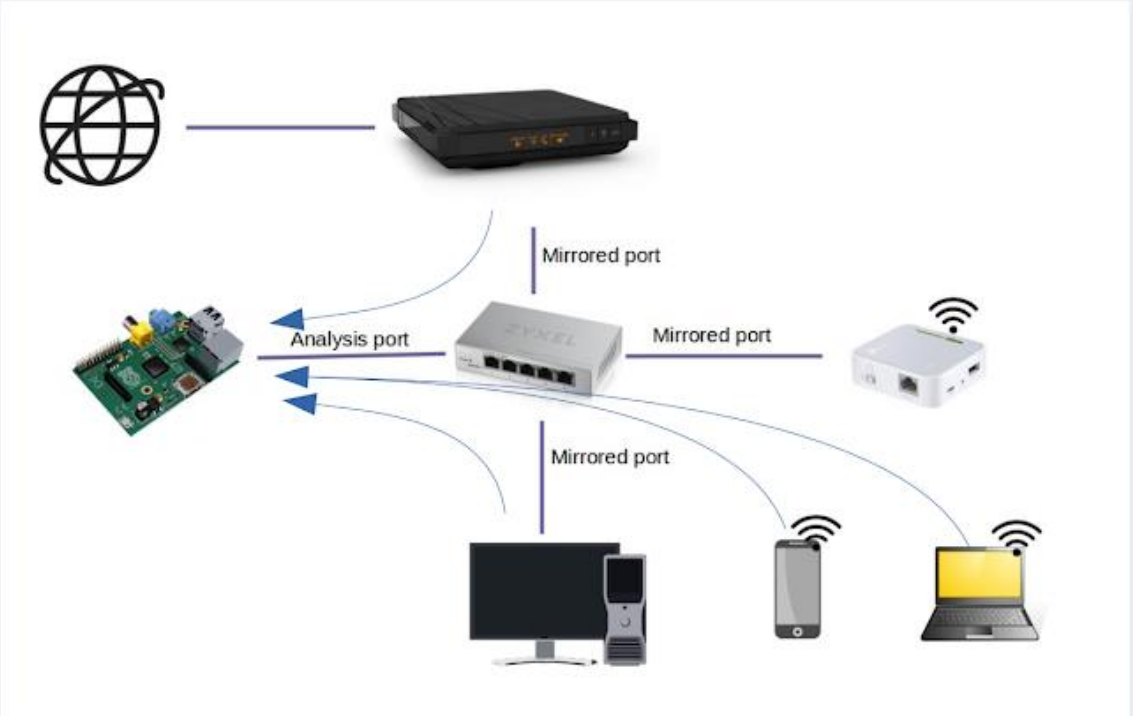

Ilustración 18 Puesto de análisis de switch.

- La primera es mediante el uso de un switch tipo Switch administrado con puerto de duplicación Zyxel GS1200 con un puerto de análisis configurable que permita configurar el reenvío de todo el tráfico al dispositivo raspberry para su análisis. Esta opción tiene los beneficios de aligerar la implementación y configuración del dispositivo raspberry, pero

también incrementa los costes, al tener que tener otro dispositivo adicional a parte del router, que requiere configuración. El esquema queda reflejado en la ilustración 11.

- La segunda es hacer que el dispositivo raspberry ejerza de punto de acceso wifi. Esta opción tiene la ventaja de ser muy exportable con poca configuración ya que la generación de la nueva red wifi, se hace mediante configuración dentro del dispositivo Raspberry.

Para evitar los costes adicionales de un router, se procede a utilizar la segunda opción. Además, en los hogares, el proveedor de internet suele tener un router estandarizado. Si no hay un acuerdo a través de las empresas para pagar los gastos de internet a través de las empresas, el Real Decreto 28/2020 de 22 de septiembre no obliga a ello, lo normal será encontrar varios tipos de dispositivos en los hogares de los empleados.

En este escenario, se genera un punto de acceso Raspberry-PISU mediante una interfaz wifi nueva que se llamará uap0 y que tendrá una dirección IP estática 10.0.0.1

La raspberry se conecta al router mediante wifi con wlan0 y cable con eth0. La primera se mantiene para crear una configuración mediante iptables que reenvié el trafico del punto de acceso nuevo a la interfaz wlan0 y el de vuelta.

Se revisa el estado del arte sobre los productos NIDS disponibles expuesto en el apartado 1.6 para identificar la mejor opción.

Se opta por Suricata por la arquitectura multihilo que puede sacar beneficio de los cuatro cores de la raspberry y que permite procesar paquetes en paralelo en múltiples hilos y se procede a su configuración (ref. 25):

En la ilustración 12 se ve el esquema final de la implementación. Para asegurar el no abuso de la conexión wifi que queda disponible en el router, se puede usar el whitelist/blacklist de MACs que serán permitidos no usar el nuevo punto de acceso con IDS.

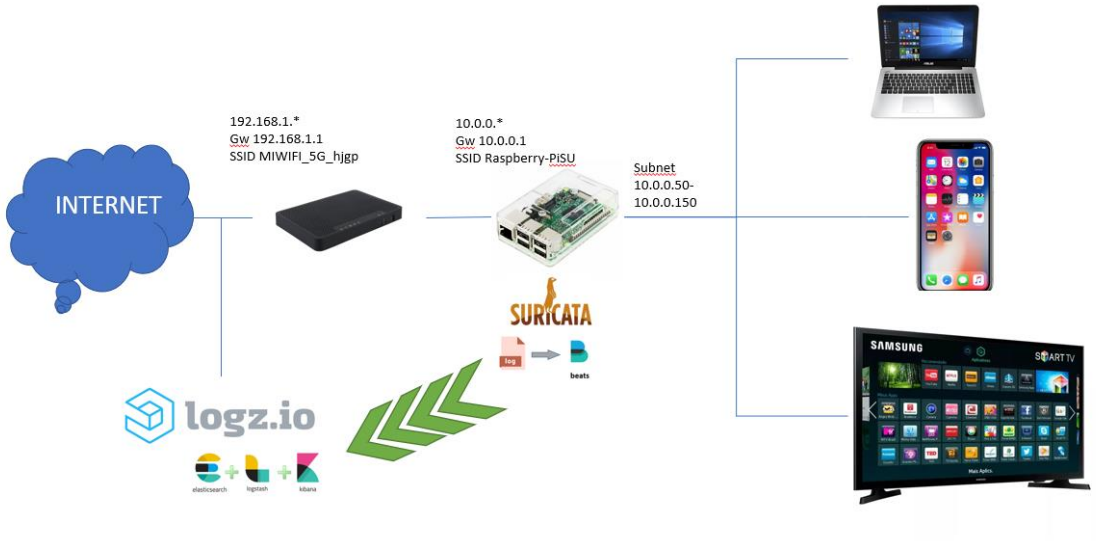

Ilustración 19 Raspberry con configuración como Access Point

Como optimizaciones, que se detallarán en el apartado correspondiente, se propone federar de los logs a través de logz.io (https://logz.io/) que ofrece un SaaS (Software as a Service) ELK (elastic search + logtash + kibana) simulando la interfaz que usaría el SOC (Security Operations Center) (Requisitos 6 y 10). La comunicación de datos del IDS se realizada mediante un agente Beats tipo filebeat.

Esta configuración requiere que la configuración de raspberry ofrezca un punto de acceso pero que además tenga internet local disponible para poner emitir los eventos de suricata a través de beats hacia logz.io.

3. Fase de Desarrollo

La fase de implementación se corresponde con los pasos Prototipar y Testear de la metodología Design Thinking. Design Thinking establece una etapa de ideación donde se presentan resultados y se recogen ideas. El objetivo de este apartado es diseñar e implementar un prototipo que sirva de base para cubrir los 10 requisitos definidos en el apartado anterior.

Como se anunció en el apartado 1.3 sobre metodología, para el prototipado se usará Lean StartUp + Agile para iterar hasta encontrar la solución definitiva durante la fase de optimización.

Se definen las siguientes Historias de Usuario:

Historias de Usuario
Instalación Raspbian
Instalación Hostapd/Dnsmasq
Edición Config Files
Creación Startup Script
Edición rc.local
Desabilitar servicios y reboot
Actualización Suricata
Testeo de la Solución

Ilustración 20 Historias de Usuario

Para ello se siguen los siguientes pasos (ref. 24). Usaremos una configuración con un interfaz wifi virtual uap0 que permite usar el interfaz wifi real tanto en modo cliente como Punto de Acceso.

3.1 Detalle de pasos

3.1.1 Instalación Raspian

Tras seguir los pasos iniciales (Ref.23) para obtener una imagen en tarjeta de Raspian OS y seguir el proceso de instalación una vez insertada en la

Raspberry, se actualiza a la última versión de los paquetes disponibles de Raspbian OS:

```
sudo apt update
sudo apt upgrade
sudo reboot
```

3.1.2 Instalación Hostapd / Dnsmasq

Se instala el paquete que permite gestionar un punto de acceso wifi:

```
sudo apt install hostapd dnsmasq
```

A continuación, se detalla el contenido de los ficheros de configuración:

/etc/hostapd/hostapd.conf

```
channel=1
ssid=Raspberry-PiSU
wpa_psk=1fbc61cef3ac243c9ddc4c7aae7756e1e3703a71fc696fe8efd73efe1dc551
99
country_code=ES
interface=uap0
# Use the 2.4GHz band (I think you can use in ag mode to get the 5GHz
band as well, but I have not tested this yet)
hw_mode=g
# Accept all MAC addresses
macaddr_acl=0
# Use WPA authentication
auth_algs=1
# Require clients to know the network name
ignore_broadcast_ssid=0
# Use WPA2
wpa=2
# Use a pre-shared key
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
#driver=nl80211
# I commented out the lines below in my implementation, but I kept
them here for reference.
# Enable WMM
wmm_enabled=1
# Enable 40MHz channels with 20ns guard interval
ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
```

Con esto conseguimos definir el punto de acceso, que luego aparecerá disponible desde los dispositivos en casa. La contraseña la guardamos encriptada.

A continuación, debe instalarse un paquete que permita la gestión de las peticiones DNS y DCHP, que son protocolos, el primero, para resolver nombres de dominios a IPs y el segundo responsable de gestionar la asignación dinámica de IPs en una red local.

3.1.3 Edición Config files

/etc/dnsmasq.conf

```
interface=lo,uap0            #Use interfaces lo and uap0
no-dhcp-interface=lo,wlan0
bind-interfaces              #Bind to the interfaces
server=8.8.8.8               #Forward DNS requests to Google DNS
#domain-needed                 #Don't forward short names
bogus-priv                   #Never forward addresses in the non-routed address
spaces
dhcp-range=10.0.0.50,10.0.0.150,12h
```

Esta configuración nos va a permitir asignar IPs de manera dinámica para los dispositivos que se conecten al nuevo AP en el rango de IPs desde la 10.0.0.50 a la 10.0.0.150. Como DNS se usará 8.8.8.8 que es la IP del servicio DNS público de Google. Cisco OpenDNS puede ser una gran opción para un control de las direcciones que se pueden visitar y establecer un control parental estricto globalmente sobre las URLs que se podrán navegar. Esto permite evitar un gran número de amenazas sobre visitas a sitios fraudulentos y permite establecer opciones de control parental.

/et/dhcpcd.conf

Este contenido nos permite genera la configuración de la interfaz virtual uap0 que tendrá la IP estática que actuará de Gateway para el punto de acceso y la subred 10.0.0.*.

```
# A sample configuration for dhcpcd.
# See dhcpcd.conf(5) for details.

# Allow users of this group to interact with dhcpcd via the control socket.
#controlgroup wheel

# Inform the DHCP server of our hostname for DDNS.
hostname

# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per
RFC4361.
# Some non-RFC compliant DHCP servers do not reply with this set.
# In this case, comment out duid and enable clientid above.
#duid

# Persist interface configuration when dhcpcd exits.
persistent

# Rapid commit support.
# Safe to enable by default because it requires the equivalent option set
# on the server to actually work.
option rapid_commit

# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# Most distributions have NTP support.
#option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#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
```

```
# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

#__AP_SETUP_START__
interface uap0
   static ip_address=10.0.0.1
   nohook wpa_supplicant
#__AP_SETUP_END__
```

3.1.4 Creación Startup script

~/network-setup/bin/netStart

```
# Check shutdown flag file exists for proper last time shutdown
# and if last time shutdown did not happen properly then reboot to make sure that,
# netStop.service properly do the necessary things before shutdown:

# Output the standard errors and messages of rc.local executions to rc.local.log file.
exec 2> /home/pi/network-setup/log/rc.local.log
exec 1>&2

# Attach script for improper shutdown recovery:
source /home/pi/network-setup/bin/shutdownRecovery


#Make sure no uap0 interface exists (this generates an error; we could probably use an if statement to check if it exists first)
echo "Removing uap0 interface..."
iw dev uap0 del

#Add uap0 interface (this is dependent on the wireless interface being called wlan0, which it may not be in Stretch)
echo "Adding uap0 interface..."
iw dev wlan0 interface add uap0 type __ap

#Modify iptables (these can probably be saved using iptables-persistent if desired)
echo "IPV4 forwarding: setting..."
#sysctl net.ipv4.ip_forward=1
```

```
#echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
sed -i 's/^#net.ipv4.ip_forward=.*$/net.ipv4.ip_forward=1/' /etc/sysctl.conf
echo "Editing IP tables..."
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -F
iptables -t nat -F
sleep 2
iptables -t nat -A POSTROUTING -s 10.0.0.0/24 ! -d 10.0.0.0/24 -j
MASQUERADE
iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
iptables -A FORWARD -i wlan0 -o uap0 -m state --state
RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i uap0 -o wlan0 -j ACCEPT
#iptables-save > /etc/iptables/rules.v4
iptables-save > /etc/iptables.ipv4.nat
#iptables-restore < /etc/iptables.ipv4.nat

# Bring up uap0 interface. Commented out line may be a possible alternative to
using dhcpcd.conf to set up the IP address.
#ifconfig uap0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255
ifconfig uap0 up

# Start hostapd. 10-second sleep avoids some race condition, apparently. It
may not need to be that long. (?)
echo "Starting hostapd service..."
systemctl start hostapd.service
sleep 10

#Start dhcpcd. Again, a 5-second sleep
echo "Starting dhcpcd service..."
systemctl start dhcpcd.service
sleep 20

echo "Starting dnsmasq service..."
systemctl restart dnsmasq.service
#systemctl start dnsmasq.service

echo "Enabling netStop service..."
systemctl enable netStop.service
systemctl start netStop.service

echo "netStart DONE"
bash -c 'echo "$(date +"%Y-%m-%d %T") - Started: hostapd, dnsmasq,
dhcpcd" >> /home/pi/network-setup/log/network.log'
```

Con esta configuración, regeneramos la interfaz uap0 y establecemos mediante reglas iptables la posibilidad de nat sobre la interfaz uap0 que hará forward del tráfico a la interfaz wlan0. Este script puede encontrarse en github bajo el

usuario idev1. En el anexo se adjunta el script completo, así como el contenido del fichero README con instrucciones.

/etc/rc.local

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

/bin/bash /home/pi/network-setup/bin/netStart
exit 0
```

3.1.5 Deshabilitar servicios y reboot

Necesitamos deshabilitar servicios puesto que el script netStart los levantará en determinado orden en el rearranque

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
sudo systemctl stop dhcpcd
sudo systemctl disable hostapd
sudo systemctl disable dnsmasq
sudo systemctl disable dhcpcd
```

3.1.6 Instalación de NIDS Suricata

En apartados anteriores, se vio los diferentes tipos de IDS existentes. El propósito de este trabajo es elegir un NIDS adecuado para monitorizar la red.

Para ello se preparar la instalación de las dependencias necesarias:

```
sudo apt install libpcre3 libpcre3-dbg libpcre3-dev build-essential
libpcap-dev libyaml-0-2 libyaml-dev pkg-config zlib1g zlib1g-dev
make libmagic-dev libjansson-dev rustc cargo python-yaml python3-
yaml liblua5.1-dev
```

Se descargan las fuentes de Suricata:

```
wget https://www.openinfosecfoundation.org/download/suricata-
6.0.1.tar.gz
```

Y se proceder a su compilación (ref.25).

```
./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var --
enable-nfqueue --enable-lua
```

En ocasiones se encontrarán dependencias que tendrán que instalarse con sudo apt install [dependencia].

```
To build and install run 'make' and 'make install'.

You can run 'make install-conf' if you want to install initial configuration
files to /etc/suricata/. Running 'make install-full' will install configuration
and rules and provide you a ready-to-run suricata.

To install Suricata into /usr/bin/suricata, have the config in
/etc/suricata and use /var/log/suricata as log dir, use:
./configure --prefix=/usr/ --sysconfdir=/etc/ --localstatedir=/var/
```

Ilustración 21 Configuración suricata

```
make
sudo make install
```

etc/suricata/suricata.yaml

Se modifique la variable HOME_NET para que contenga la red local compartida a través de la interfaz uap0:

```
HOME_NET: "[10.0.0.0/24]"
```

3.1.7 Actualización de Reglas

Esta es una herramienta desarrollada en Python que permite por defecto obtener las reglas por defecto del conjunto Emerging Threads ruleset (Ref.39). Emerging threads es una división de ProofPoint, Inc. Entre las principales categorías nos encontramos:

- Attack-Response: están diseñadas para identificar un ataque exitoso.
- BotCC: Botnets activos y otros servidores Command and Control.
- Compromised: es una lista de hosts que han sido comprometidos.
- Current_Event: reglas en test, o muy recientes.
- DOS: detectan Denial Of Service.
- DROP: Spammers profesionales.
- DShield: importadas del sitio dshield.
- Exploit: detectan exploits directos como SQL injections.
- Game: identifican el uso de juegos.
- Inappropriate: identifican sitios con contenidos no apropiado (pornografía, pedofilía, etc).
- Malware: spyware y la detección de muchas amenazas están incluidas en esta categoría. Esta se la categoría más importante.
- P2P: detectan este tipo de tráfico (bottorrent, emule, …).
- Policy: reglas usuales en las principales compañías sobre políticas de comportamiento y uso de las redes.
- Scan: Nessus, Nikto, portscannings serían detectados mediante este conjuntos de reglas.
- VOIP: amenazas sobre VOIP.
- Web: sql injection, web server overflows, vulnerable web apps. Importante cuando se tienen servidores web dentro de la red donde el IDS opera.
- WEB-SQL-Injection: intenta capturer ataques sobre aplicaciones. En un entorno como el descrito, puede ser útil para identificar empleados malintencionados.

Suricata-update permite instalar otras listas de reglas, así como deshabilitar partes del conjunto.

Se instala suricata-update:

```
pi install suricata-update.
```

Al lanzarse suricata-update se produce la descarga de más de 20000 reglas, que se instalan en el directorio /var/lib/suricata/rules

Tras cubrir estos pasos, se inicia suricata con el siguiente comando:

```
sudo suricata -c /etc/suricata/suricata.yaml -i uap0 -S
/var/lib/suricata/rules/suricata.rules
```

Ilustración 22 Suricata ejecutándose en Raspberry PI

Para utilizar suricata en modo servicio debe generarse un fichero especial:

/etc/systemd/system/suricata.service

```
# Sample Suricata systemd unit file.
[Unit]
Description=Suricata Intrusion Detection Service
After=network.target syslog.target
[Service]
ExecStartPre=/bin/rm -f @e_rundir@suricata.pid
ExecStart=/usr/bin/suricata -c /etc/suricata/suricata.yaml -i uap0
-S /var/lib/suricata/rules/suricata.rules --pidfile
@e_rundir@suricata.pid
ExecReload=/bin/kill -HUP $MAINPID
ExecStop=/bin/kill $MAINPID
[Install]
WantedBy=multi-user.target
```

Para su arranque puede usarse el comando:

```
sudo systemctl enable suricata.service
```

3.1.8 Testeo de la Solución

En este apartado, tras la implementación del MVP, se proceder a definir escenarios donde debe verse la alerta correspondiente en Suricata. Para ello se utiliza un portátil conectado al Access Point ofrecido por la Raspberry PI 3.

Las evidencias se obtienen del fichero /var/log/suricata/fast.log

Escenario 1

Al intentar acceder a la URL http://testmyids.com/ debe producirse una alerta.

Evidencia:

22/03/2021-16:13:42.020071 [**] [1:2100498:7] GPL ATTACK_RESPONSE id check returned root [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 31.3.245.133:80 -> 10.0.0.142:46196

Escenario 2

Ejecutando el siguiente comando desde un shell de Linux:

dig a 3wzn5p2yiumh7akj.onion o directamente abriendo en el navegador TOR ese dominio (3wzn5p2yiumh7akj.onion) debe producir una siguiente alerta:

Evidencia:

22/03/2021-16:19:40.496621 [**] [1:2022048:3] ET MALWARE Cryptowall .onion Proxy Domain [**] [Classification: A Network Trojan was detected] [Priority: 1] {UDP} 10.0.0.141:37423 -> 10.0.0.1:53

3.1.9 Conclusiones

Tras implementar el MVP seleccionado en esta primera iteración, el NIDS es capaz de evaluar el impacto en todos los equipos conectados al Access Point. Por tanto, el objetivo de conseguir monitorizar todo el tráfico se consigue de forma eficiente.

Pérdida de paquetes

Se realiza una prueba con múltiples dispositivos conectados para ver si suricata pierde paquetes, pero en todos los casos probados el packet loss es 0. Esto se puede ver a través de las trazas que se consiguen en el fichero /var/log/suricata/stats.log en el parámetro capture.kernel_drops. Una forma de mitigación en caso de aparecer pérdida de paquetes es a través de la variable ring-size en el fichero /etc/suricata/suricata.yml que por defecto es 2048 y probar con valores más altos como 3000.

Rotado de logs

El tamaño del archivo de registro eve.log puede volverse muy grande rápidamente. El mecanismo logrotate de Linux es muy útil para prevenirlo. Una configuración típica podría permitir el tráfico de sólo 10 días y limitar el tamaño de cada archivo a 1 GB. Los historiales más antiguos se eliminan automáticamente. En la fase de optimizaciones al asegurar, que mandamos los datos mediante filebeat a ELK fuera del equipo, nos aseguramos que de esta manera la información no se pierde, lo que permite la prueba en caso de llegar a un tribunal. El fichero a generar es el siguiente:

/etc/logrotate.d/suricata:

```
{
    daily
    maxsize 1G
    rotate 10
    missingok
    nocompress
    create
    sharedscripts
    postrotate
    systemctl restart suricata.service
    endscript
}
```

Eliminación de reglas

La eficacia del IDS se basa en una buena gestión de reglas. Tanto para evitar falsos positivos como para asegurar que las reglas utilizadas estén actualizadas y permitan la detección de amenazas recientes.

Cada regla tiene el siguiente formato:

- Action: determina la acción de detección.
- Header: define protocol, IPs y puertos.
- Options: información adicional para calificar si la regla aplica.

Para deshabilitar reglas se puede usar el fichero:
/etc/suricata/disable.conf

Usando posteriormente la opción:
suricata-update --disable-conf=/etc/suricata/disable.conf

Un ejemplo, si se usa habitualmente Skype puede usar esta regla:

```
1:2002157 # CHAT Skype User-Agent detected
```

Programación de suricata-update

Para evitar perder efectividad, se puede añadir la ejecución de suricata-update como tarea programada usando la funcionalidad crontab -e.

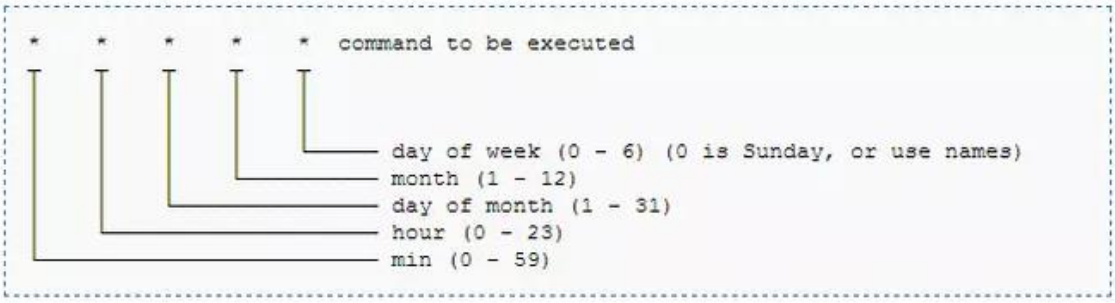

Ilustración 23 Crontab configuración de tarea

3.1.10 Siguientes Pasos

El siguiente paso es analizar optimizaciones a cubrir como el siguiente Viable Product en el apartado de conclusiones y enlazar con la solución SIEM seleccionada (stack ELK) en la opción SaaS.

De esta manera, se completará la implementación final, se medirán resultados y se cubrirán todos los requisitos analizados incluyendo la generación de un dashboard y notificaciones.

4. Fase de Optimización

La fase de optimización desarrolla el MVP detectado y que se ha probado como Prueba de Concepto y se corresponde con las etapas de Medir y Aprender de Lean Startup. En esta fase, se procederá a conectar con un sistema que podría ser usado por el Equipo de Respuesta a Incidentes de Seguridad y se detallan las mediciones y resultados obtenidos y se genera conclusión sobre el prototipo y su puesta en funcionamiento.

Las User Stories que se definen en este apartado incluyen:

- Integración logs con solución ELK: Logstash + Elasticsearch + Kibana para tener una forma de gestionar los resultados externamente y accesible a un analista de seguridad de la organización. Esta solución se empleará mediante la modalidad SaaS.
- Configuración del dashboard y explotación de la información.
- Análisis de resultados y comportamiento del sistema.

Las conclusiones de esta fase incluyen oportunidades de mejora y se genera un Roadmap para futura implementación y puesta en producción.

4.1. Solución SaaS ELK

Para la agregación de todos los resultados de empleados en una herramienta central se usa el servicio Logz.io (https://logz.io/) que es un Software as a Service, y que ofrece una cuenta de prueba que como restricción sólo mantiene el tráfico por dos días y con un máximo de 1 gigabyte.

El flujo que se utiliza es el siguiente:

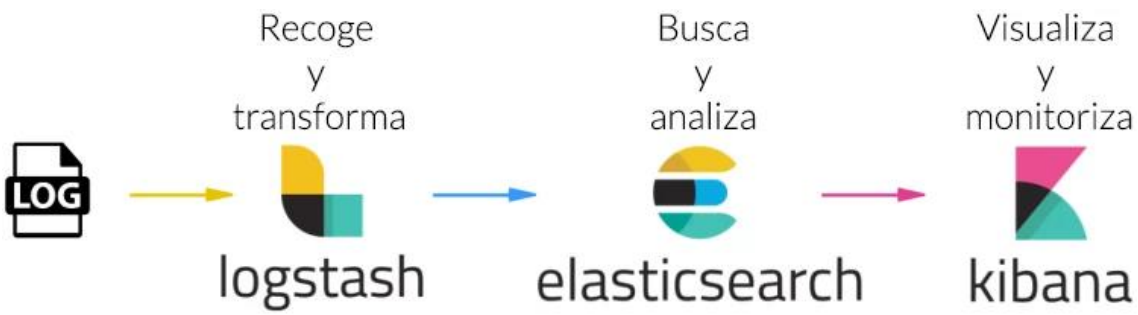

Ilustración 24 Solución ELK y flujo desde log

Suricata genera toda la información en /var/log/suricata/eve,json que es un log en formato json con una línea por evento.

El salto desde el log a logtash no es automático y requiere el uso de un agente intermedio tipo filebeat. La desventaja de este producto es que no hay una fácil instalación en Raspberry puesto que no existe paquete precompilado en arquitectura ARM que es la usada por el dispositivo raspberry. Elastic no mantiene esta arquitectura en el momento de creación de este trabajo.

Esto requiere la descarga del producto y su compilación (Ref.31). El producto está construido con Golang (https://golang.org/), para generar el build se procede a usar una imagen de Docker con golang instalado. Consecuentemente, se procede a la instalación de docker:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

Una vez instalado puede procederse a la definición de dos variables

```
GOVERSION=1.13.10
BEATSVERSION=v7.10.0
```

Y la ejecución del siguiente commando:

```
docker run --rm -it -v pwd:/build golang:${GOVERSION} /bin/bash -c "go
get         -d         -v         github.com/elastic/beats;         cd
```

```
/go/src/github.com/elastic/beats/filebeat/;        git        checkout
${BEATSVERSION}; GOARCH=arm go build; cp filebeat /build/filebeat-
${BEATSVERSION}"

echo "A new filebeat has been built"
```

La imagen entonces estará disponible y se podrá copia al directorio /usr/local/bin:

```
root@raspberrypi:/var/lib/docker/volumes/pwd/_data# ls -lsa
total 76604
    4 drwxr-xr-x 2 root root     4096 may 23 17:03 .
    4 drwx-----x 3 root root     4096 may 23 16:41 ..
76596 -rwxr-xr-x 1 root root 78430657 may 23 17:03 filebeat-v7.10.0
root@raspberrypi:/var/lib/docker/volumes/pwd/_data#
```

Ilustración 25 Imagen filebeat

Se procede a habilitar el servicio, para ellos se genera el siguiente fichero:

/etc/systemd/system/filebeat.service

```
[Unit]
Description=Filebeat sends log files to Logstash or directly to
Elasticsearch.
Documentation=https://www.elastic.co/products/beats/filebeat
Wants=network-online.target
After=network-online.target

[Service]
ExecStartPre=-/bin/mkdir -p /usr/share/filebeat \
/etc/filebeat /var/lib/filebeat \
/var/log/filebeat /usr/share/filebeat/module
ExecStart=/usr/local/bin/filebeat -c /etc/filebeat/filebeat.yml \
-path.home /usr/share/filebeat -path.config /etc/filebeat \
-path.data /var/lib/filebeat -path.logs /var/log/filebeat
Restart=always

[Install]
WantedBy=multi-user.target
```

Posteriormente, para poder operar con elk en cloud, debe procederse a configurar el siguiente fichero en notación ansible:

```
/etc/filebeat/filebeat.yml
```

```
################ Filebeat #####################################

filebeat.inputs:

- type: log
  paths:
    - /var/log/suricata/eve.json
  fields:
    logzio_codec: json
    token: QxLBFFFUcnPxFMlccRaKIweEtnimrvdv
    type: suricata
  fields_under_root: true
  encoding: utf-8
  ignore_older: 3h

#For version 6.x and lower
#filebeat.registry_file: /var/lib/filebeat/registry

#For version 7 and higher
filebeat.registry.path: /var/lib/filebeat

#The following processors are to ensure compatibility with version 7
processors:
- rename:
    fields:
     - from: "agent"
       to: "beat_agent"
    ignore_missing: true
- rename:
    fields:
     - from: "log.file.path"
       to: "source"
    ignore_missing: true

############################                                    Output
#########################################

output:
  logstash:
    hosts: ["listener-uk.logz.io:5015"]
    ssl:
```

```
      certificate_authorities:
['/etc/pki/tls/certs/COMODORSADomainValidationSecureServerCA.crt']
```

Debe procederse a la descarga del certificado para generar una comunicación segura:

```
sudo curl https://raw.githubusercontent.com/logzio/public-
certificates/master/AAACertificateServices.crt --create-dirs -o
/etc/pki/tls/certs/COMODORSADomainValidationSecureServerCA.crt
```

Posteriormente se procede a habilitar el módulo para leer logs de suricata:

```
filebeat -c /etc/filebeat/filebeat.yml modules enable suricata
```

Si no están los módulos disponibles, puede descargarse de la página de elastic el producto filebeat para Windows y en el .zip estarán disponibles. Se copian a /etc/filebeat/modules.d/

Y acto seguido se lanza el producto:

```
filebeat -c /etc/filebeat/filebeat.yml
```

4.1.1 Filtros

Ahora, ya estamos listos para analizar la información mediante el ELK en cloud de logz.io. Cuando nos logamos en el sistema, si vamos ahora a Kibana:

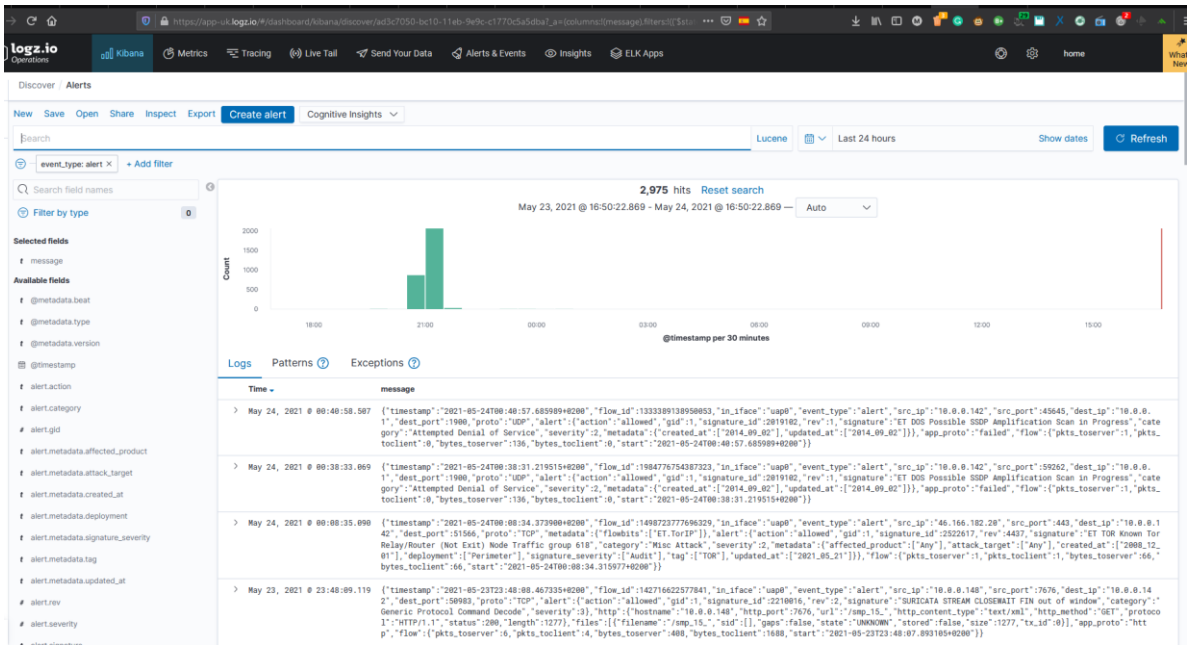

Ilustración 26 Kibana

Tal y como aparece en la ilustración 31, ya se puede ver información proveniente del fichero eve.log que ha sido tratada y transferida al cluster de elasticsearch e indexada. La imagen anterior muestra la aplicación de un filtro sencillo: event_type: alert para ver las alertas disponibles. Se pueden apreciar 3 filas con severidad 2.

La primera y la segunda provienen de la aplicación de la regla 2019102 ET DOS Amplification Scan in Progress y la tercera tipo 2, de la regla 2522617 ET TOR known Tor Relay/Router Node traffic group 618.

La explicación proviene por el lanzamiento de un escaneo de puertos con NMAP en la ip del punto de acceso desde un portátil con Windows conectado a la red servida por la raspberry:

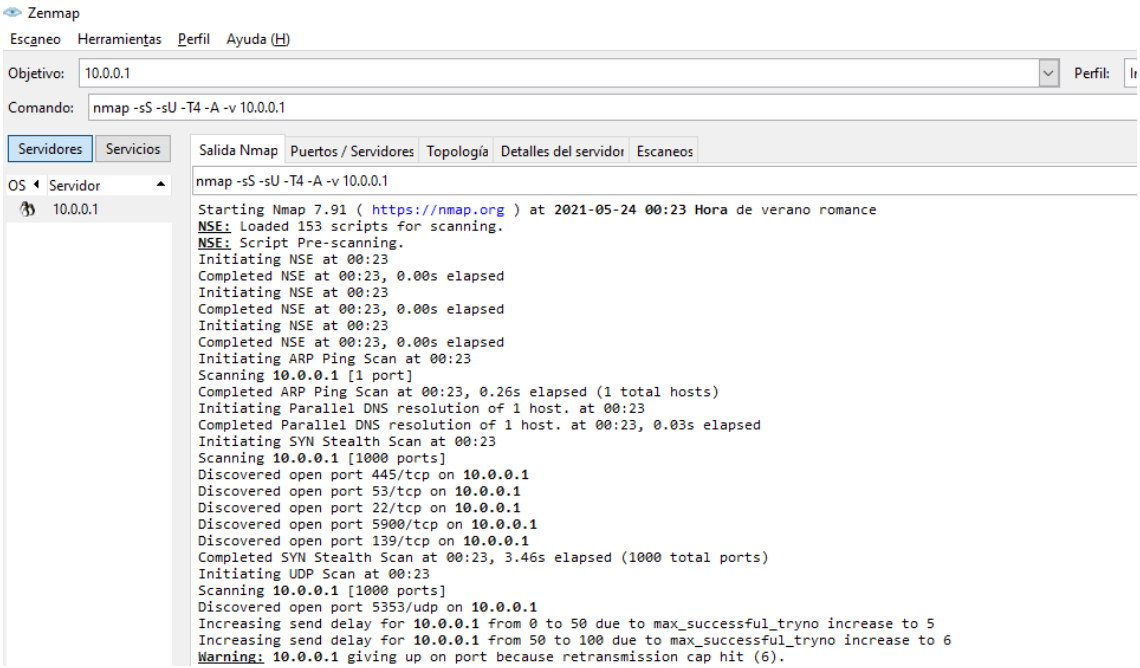

Ilustración 27 NMAP escaneo de puertos activa alerta con severidad 2

Y en el tercer caso, se explica por el uso de TOR desde el mismo portátil:

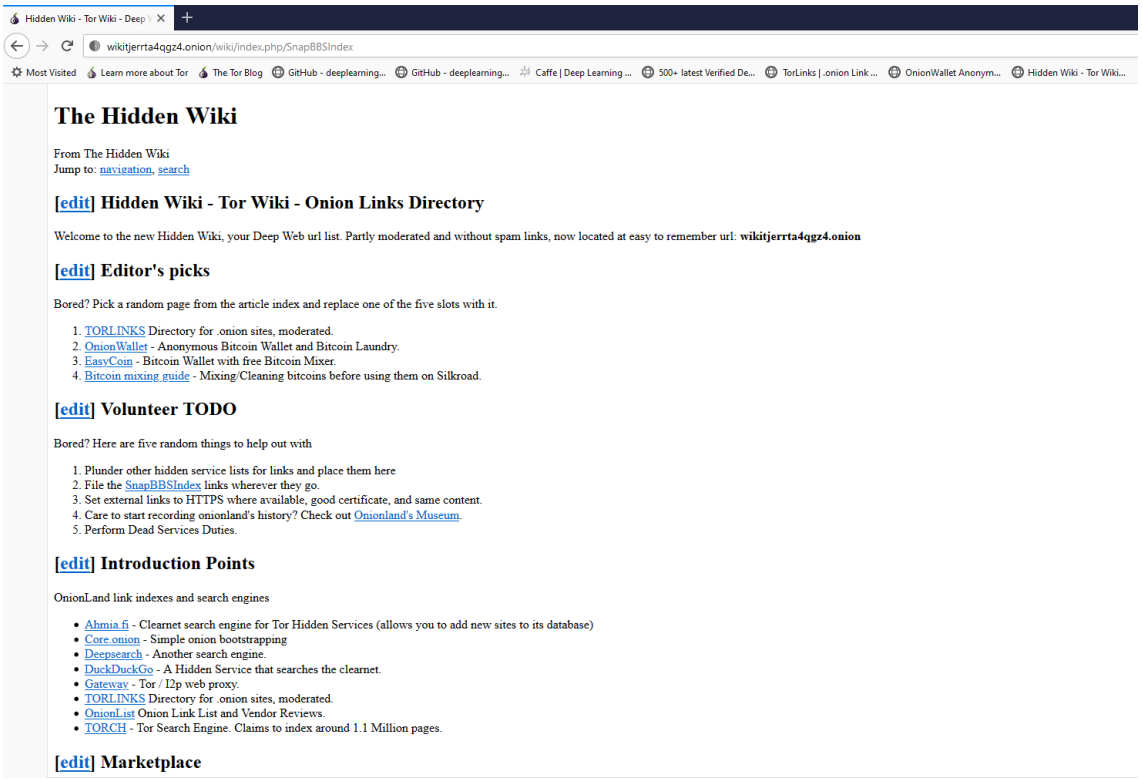

Ilustración 28 TOR con página cargada de la deepweb.

Se pueden almacenar las búsquedas, como por ejemplo una por anomalías:

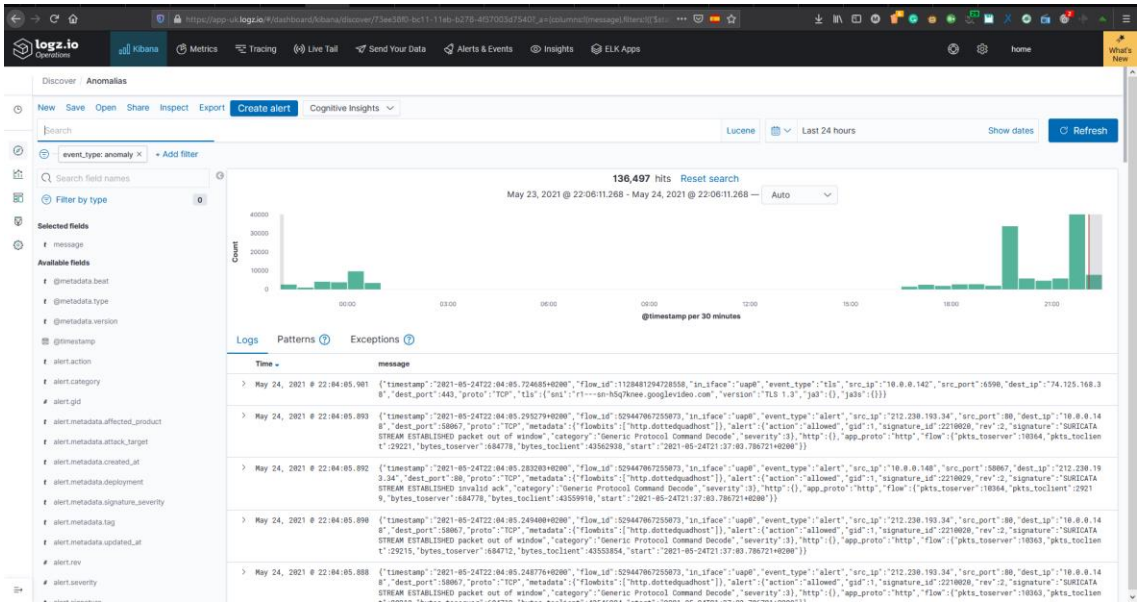

Ilustración 29 Búsqueda por anomalías

Hay un pico de anomalías claramente visible, revisando la información de las entradas puedo apreciar rápidamente que la ip 212.230.193.34 hace saltar la regla 2210020 SURICATA STREAM ESTABLISHED packet out of window con severidad 3. Al realizar un whois sobr la ip, se identifca que pertenece a MASMOVIL, el ISP usado, en concreto es la IP de su grupo de servidores DNS.

A pesar de tener el DNS de Google configurado para todos dispositivos conectados a la red servida por el punto de acceso en 10.0.0.1, la raspberry tiene conexión propia con las interfaces eth0 y wlan0 que usan dhcp para coger la configuración del servidor DCHP del router de acceso a internet.

4.1.2 Gráficos

A parte de realizar búsquedas, Kibana nos permite generar gráficos. Por ejemplo, uno por tipo de tráfico por dispositivo mediante piechart:

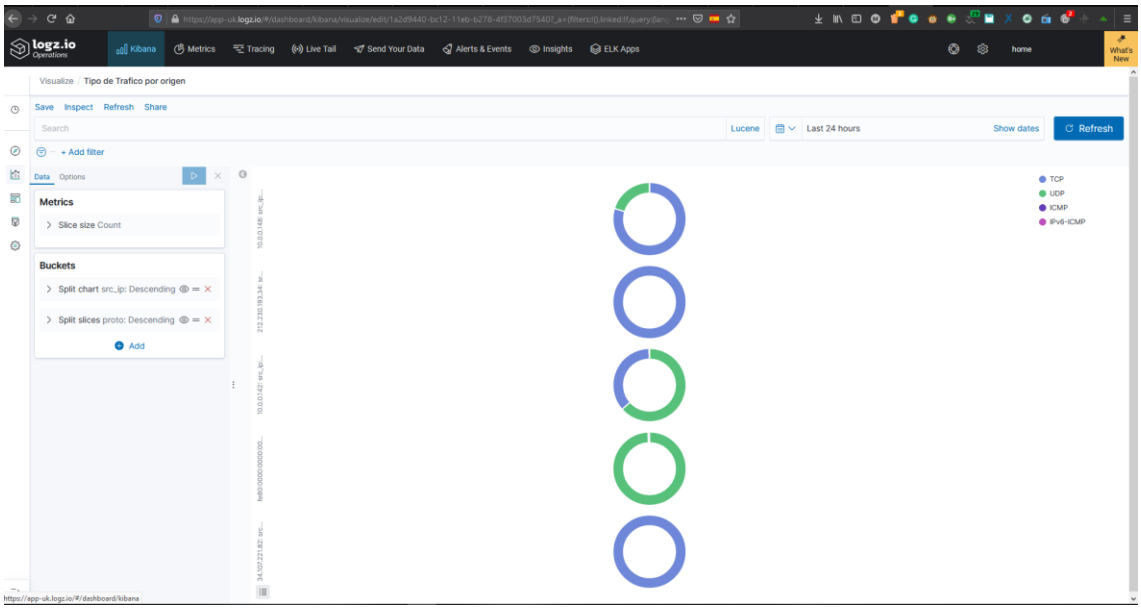

Ilustración 30 Gráfico con tipo de tráfico por dispositivo

Dentro de las cinco IPs identificadas, se aprecia que una usa IPv6. Revisando la IP me permite identificar que algún dispositivo puede estar usando IPv6 para conectarse. Procedo a usar Whireshark para saber más sobre el tráfico:

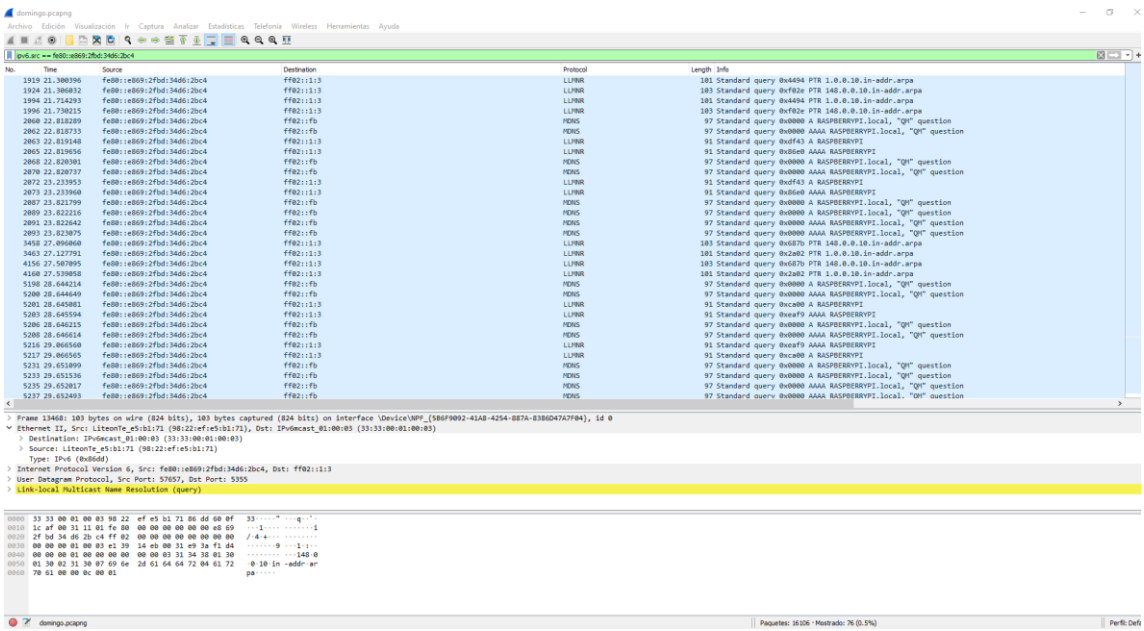

Ilustración 31 Captura de Whireshark

Y veo que correponde a tráfico con el protocolo LLMNR (Link Local Multicast Name Resolution) o protocolo de resolución local que funciona tanto para IPv4 como IPv6.

Las posibilidades de filtrado y del uso de gráficos permiten ajustar la configuración de reglas. Por ejemplo, un gráfico por tipo de alertas recibidas en las últimas 24h muestra que las reglas están provocando un exceso de reglas de severidad baja, que, usando el filtro equivalente en la búsqueda, pueden identificarse y añadir a la configuración de reglas a ignorar por suricata-update.

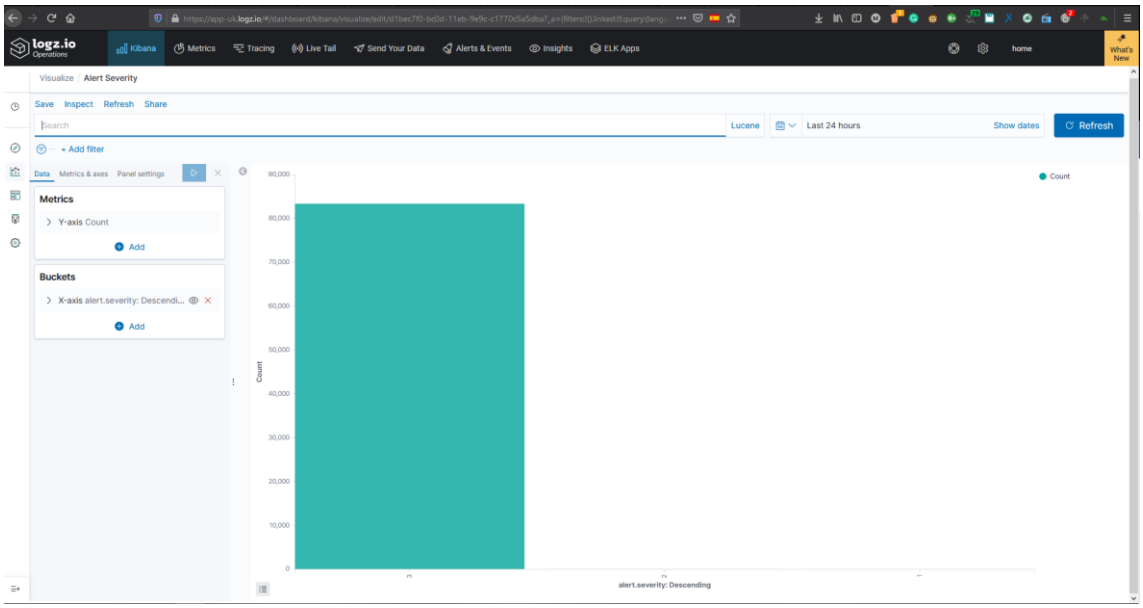

Ilustración 32 Alertas en las últimas 24h

4.1.3 Dashboard

Una vez guardadas y nombradas varías búsquedas y gráficos, se puede componer un cuadro de mando para ayudar a hacer la monitorización más simple.

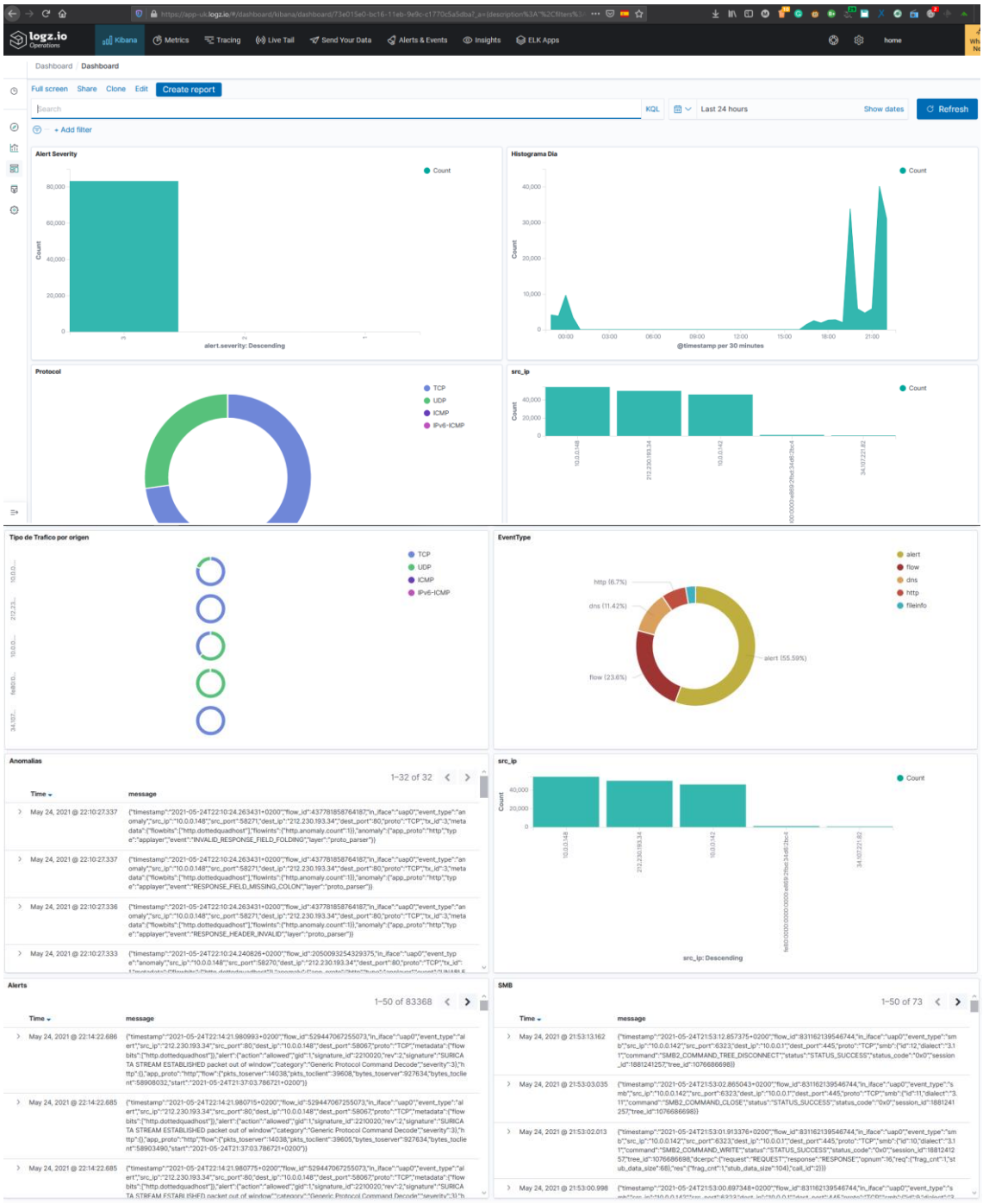

Ilustración 33 Dashboard

En el dashboard que se puede ver en la imagen, se ven diferentes áreas:

- Severidad de alertas: para evaluar el nº de alertas recibidas y el tipo.
- Histograma día: nos permite ver patrones en la cantidad de tráfico registrado en el cluster.
- Protocol: nos permite ver los tipos de protocolos en el tráfico registrado.
- Src_ip: se puede ver el tráfico generado por cada dispositivo dentro del beat raspberry. En caso de tener varios trabajadores podrían hacerse gráficos unificando los datos de los diferentes filebeats.
- Tipo de tráfico por origen: relaciona protocolos con los dispositivos de origen. Esto permite comparar comportamientos.
- EventType: permite ver los diferentes tipos de eventos registrados e identificar patrones de comportamiento.
- Búsquedas de Anomalías y Alertas: estas nos permiten ver las entradas concretas de los logs.
- SMB: búsqueda definida para identificar tráfico SMB. Esto puede ser un control establecido en la gestión de riesgos dentro del sistema de gestión de seguridad de la información para verificar que todos los equipos conectados en remoto tienen deshabilitado SMBv1 por los riesgos que entraña.

4.1.4 Alertas

Kibana permite generar alertas y notificaciones para no tener constantemente que estar monitorizar para identificar riesgos.

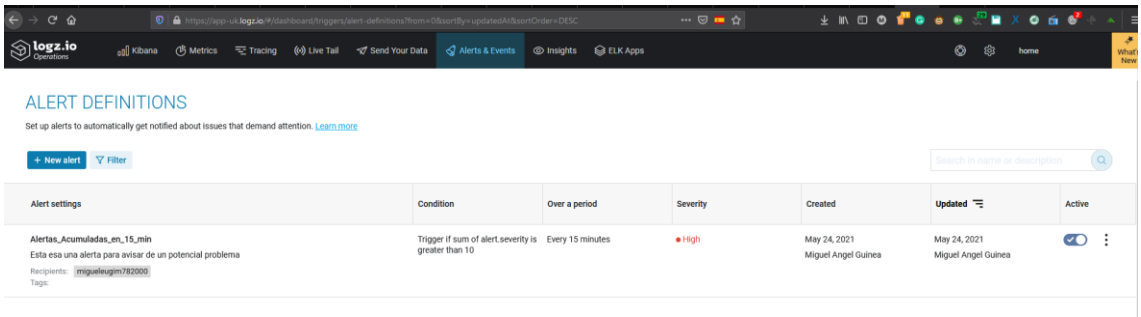

Ilustración 34 Alerta y estado de activación

En este caso la alerta definida, revisa que, en los últimos 15 minutos, la suma de todas las criticidades es mayor a 10.

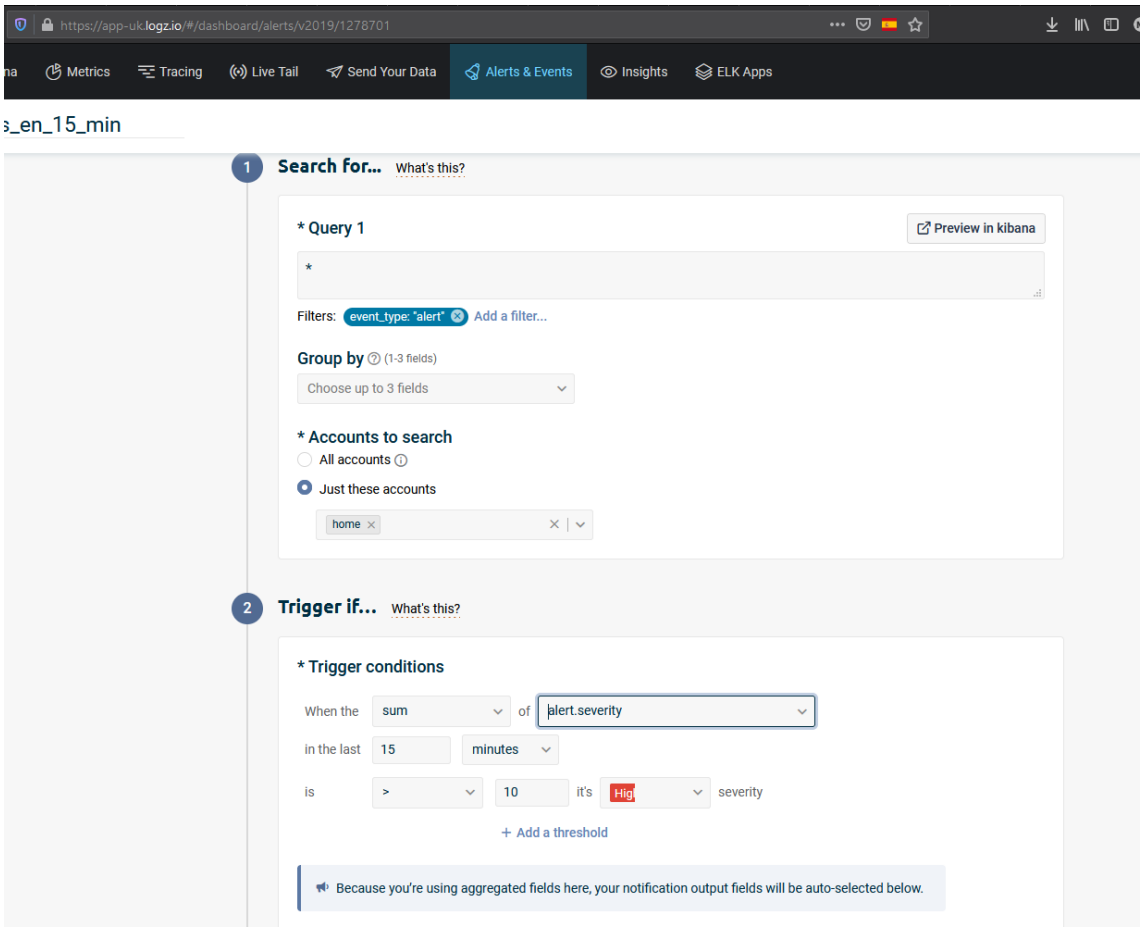

Ilustración 35 Configuración de alerta

En el caso de que se cumpla la condición se mandará un email a la dirección indicada. Se pueden definir otros webhooks como slack o servicenow que pueden bien avisar a grupos en un herramienta de comunicación interna, o formar parte de la gestión de incidentes con procedimiento ITIL para la apertura de casos.

Ilustración 36 Ejemplo de notificación

4.1.5 Reports

Se permite también la configuración para la recepción de emails periódicos con el estado del dashboard de las últimas 24h en formato pdf.

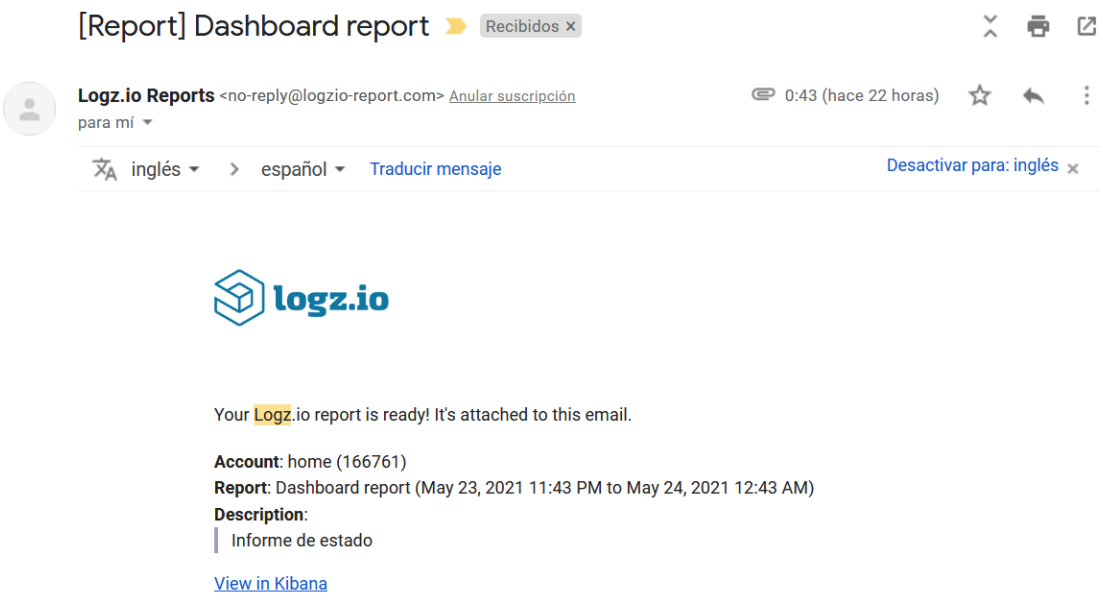

Ilustración 37 Email de recepción de informe

4.2 Consideraciones legales

Filebeat usa certificado de logz.io para enviar de forma segura la información a través de internet. Si esta solución quisiera usarse en el ámbito de empresa, a pesar de que el archivo eve.log de suricata no contiene datos personales, si contiene referencias a reglas que pueden mostrar acceso a sitios inapropiados, uso de juegos, etc y también IPs de origen y destino que permitirían desvelar el anonimato.

Una forma de prevenirlo, sería aplicar transformaciones a las IPs de origen, limitando el conocimiento de la transformación utilizada a un grupo reducido de personas. Esto puede reducir la incidencia. Otra solución es firmar acuerdos de confidencialidad, uso y borrado entre empleados y empresa.

5. Fase de Conclusiones

En este trabajo se han ofrecido datos sobre un riesgo para la seguridad creciente en las redes de los hogares mediante el aumento del teletrabajo. Se procedió a explicar los tipos de riesgos, así como los actores involucrados.

Con la aplicación de Design Thinking y Lean Startup se definió una potencial herramienta que ayude en los hogares tanto a detectar potenciales intrusos como vulneraciones de las políticas de seguridad. Y durante la implementación se ha cubierto la viabilidad de este experimento.

Lo más complejo, ha sido encontrar la forma más eficiente de convertir el dispositivo Raspberry en un Punto de Acceso Wifi y la creación de un artefacto ejecutable de Filebeat para arquitecturas ARM. El resto ha requerido más exploración mediante la práctica y test del tráfico generado o de las herramientas.

Sobre convertir el dispositivo Raspberry en un punto de acceso, se han llegado a testear diferentes opciones de reenvío de tráfico de wlan0 a eth0 mediante bridge-utils (ref. 24), pero en todas se perdía acceso desde la Raspberry a Internet y las configuraciones no eran tan elegantes ni eficientes como la tomada en este trabajo pudiendo generar una red específica a través de una interfaz de red inalámbrica sin necesidad de puentes.

Este apartado además bloqueaba el resto de investigación, por lo que llegar a una solución óptima era algo crítico en tiempo, pero también en resultados.

Para la ejecución desde un teclado y dos pantallas, y la simplificación del trabajo con un laptop y la raspberry. Opté por el uso de varias alternativas:

- Habilitación de ssh para poder abrir sesiones de terminal desde el laptop a la raspberry.
- Uso de VNC para la visualización de la pantalla de la raspberry desde mi portátil.
- Habilitación de samba para compartición de ficheros. Esto en algún punto de compartición de reglas o módulos en filebeat ha sido extremadamente útil.

Una mejora significativa al trabajo realizado en experimentación, habría sido sin duda la posibilidad de usar git para el control de versiones en cambios en ficheros clave, para revisar diferentes ramas y resultados.

Durante la fase de optimización, el pequeño dispositivo usado se integró con herramientas de agregación y análisis de datos para simular un potencial entorno corporativo.

5.1 Cobertura de Requisitos

De los requisitos definidos a cubrir por la solución:

- **Requisito 1:** Foco en NIDS por las potenciales amenazas en un entorno de hogar con wifi, para identificar esas amenazas y alertar / mitigar su impacto en la compañía y en los activos del hogar, así como daños reputacionales tanto de la compañía como del empleado / empleada, además de prevenir los daños particulares que pueda sufrir tanto esta persona como otras personas del núcleo familiar.
- **Requisito 2:** Poder usarse en los hogares e integrarse fácilmente en ecosistema IT.
- **Requisito 3:** Revisar todo el tráfico de internet del empleado.
- **Requisito 4**: Respeta la privacidad sin registrar ningún dato personal.
- **Requisito 5:** Usar bien análisis por firmas o anomalías.

- **Requisito 6:** Tener una forma de exportar la información a herramientas gráficas.
- **Requisito 7:** Capacidad de automatización para acciones reactivas.
- **Requisito 8:** Poder usarse de puente entre analista de seguridad de la organización y los empleados.
- **Requisito 9:** Tener comunidad y buena documentación.
- **Requisito 10:** capacidad de integrar con más empleados e integrar los resultados a un nivel superior mediante herramientas en cloud que permitan la identificación y seguimiento de alertas de modo centralizado.

Este trabajo ha conseguido cubrir los 10 requisitos definidos, el uso de Design Thinking ha sido útil a la hora de ampliar el escenario habitual del trabajo a distancia y diseñar requisitos que permitan escalar la solución.

El aumento de los casos de intentos de ciberataque y la confianza como situación estable de las empresas con la opción del teletrabajo podría llevar al endurecimiento de las condiciones para los IPSs para ofrecer servicios de internet. Puesto que mejorar de seguridad de su lado, supondrían un impacto en la productividad de un gran número de empresas.

5.2 Backlog de mejoras

Un potencial backlog para un segundo Minimum Viable Product podría consistir en los siguientes elementos:

- Para la detección de amenazas de espectro completo, se necesitan las capacidades que proporcionan Suricata y Zeek, además de la detección de malware basada en ML. En este contexto, aparecen soluciones como la plataforma de seguridad de red de Bricata (Ref. 28) que combina Suricata, Zeek, la detección de malware basada en ML y los datos completos de PCAP en un solo lugar para ofrecer capacidades completas de detección y respuesta de red para entornos en la nube, híbridos y locales. El uso de Bricata y las técnicas de ML puede ser interesante.

- Probar el modelo con más de un hogar y agregar los resultados bajo el mismo cluster elasticsearch con diferentes filebeats, siendo capaz de reconocer los datos.
- Implementar mecanismos para añadir privacidad. Por ejemplo, ofuscando las IPs de origen.
- Usar Logstash como ETL para definir diferentes tipos de SIEM en ELK, uno más operacional y otro donde se use resumen con datos más ejecutivos con diferentes clusters elasticsearch.

A nivel personal, el trabajo realizado me ha resultado extremadamente útil para crear un dispositivo que ahora uso de forma habitualmente en casa. El consumo en electricidad es reducido si lo comparamos con un laptop y las prestaciones a la hora de incremento en capacidad para analizar el tráfico hacen que sea una opción interesante. En mi caso, he combinado con los servidores de DNS de Cisco que mantienen control sobre tipos de IPs a visitar.

De esta manera me aseguro que cualquier persona que use la red de casa, sólo pueda visitar ciertos que no tengan mala reputación o peligro de ataque.

6. Glosario

blacklist, 41
Lista de accesos no autorizados

Black-Swan, 4
Tipo de evento no predecible que produce daños graves.

BYOD, 19
Tipo de política que hace referencia al uso de los propios dispositivos personales en el ámbito laboral como móviles u ordenadores.

CCN-CERT, 35
Capacidad de respuesta a Incidentes de seguridad del Centro Criptológico Nacional

ciberataque, 6
Explotación deliberada de una debilidad en un sistema informático.

ciberseguridad, 4
Es la práctica de defender activos (tecnológicos, redes, datos) de ataques maliciosos.

Cluster, 29
Se refiere a una agrupación de datos, computadoras, recursos.

confidencialidad, 6
Propiedad de la seguridad que se refiere a la revelación de datos exclusivamente al personal autorizado para ello.

confinamiento, 5
Reclusión o aislamiento con el propósito de contener una amenaza.
Covid-19, 3
Tipo de Virus que ocasionó una pandemia durante 2020-2021.

CSIRT, 34
Equipo de Respuesta a Incidentes

Design Thinking, 11

disponibilidad, 6
Característica de la seguridad que se refiere a la capacidad de un sistema de estar accesible y operable por los periodos y modos definidos.

Ecommerce, 5
Actividad de Comercio electrónico

ELK, 9, 16
Stack de análisis y procesamiento de datos distribuidos de alto rendimiento.

ETL, 29
Extract, trasnform and Load, tipos de herramientas capaces de extraer, limpiar y enriquecer datos antes de llevarlo a un repositorio enfocado en la agregación y análisis de datos.

Golang, 57
Lenguaje de programación de Google.

GPU, 25
Graphic Process unit.

hackers, 5
Experto entusiasta que se dedica a la programación.

IDS, ii, 9
Sistema de Detección de Intrusos.

integridad, 6
Propiedad de la seguridad referida a la capacidad de mantener los activos sin corrupción o alteración no autorizada.

IoT, 19

7. Bibliografía

Todas las visitas a las webs referencias se han hecho durante el tiempo previsto en el cronograma adjunto en el presente trabajo.

- Ref.1: Eventos tipo Blackswan.

https://www.investopedia.com/terms/b/blackswan.asp

- Ref.2: Cuarentena y Estado de Alarma en España durante 2020.

https://es.wikipedia.org/wiki/Cuarentena_en_Espa%C3%B1a_de_2020)

- Ref.3: Apuntes del Master de Ciberseguridad y Privacidad de la UOC. Sistema de Gestión de la Seguridad. Módulo5
- Ref.4: Cronograma de eventos relacionados con la progresión del COVID:

https://covidreference.com/timeline_es

- Ref.5: Cronograma por parte de la Organización Mundial de la Salud.

https://www.who.int/es/news/item/29-06-2020-covidtimeline

- Ref.6: Datos estadísticos de medidas adoptadas por empresas frente al COVID.

https://www.ine.es/daco/daco42/ice/ice_mod_covid_0320.pdf

- Ref.7: Ley 28/2020 de Trabajo a distancia

https://boe.es/buscar/pdf/2020/BOE-A-2020-11043-consolidado.pdf

- Ref.8: Ejemplo de marco de trabajo para la migración al cloud como elemento de transformación digital. AWS Cloud Adoption Framework

https://aws.amazon.com/es/professional-services/CAF/

- Ref.9: Desafíos a la ciberseguridad por el COVID.

https://www.muycomputerpro.com/2020/03/19/desafios-de-ciberseguridad-del-covid-19

- Ref.10: Ponencia sobre los riesgos para la seguridad del teletrabajo

https://www.ccn-cert.cni.es/pdf/documentos-publicos/xiv-jornadas-stic-ccn-cert/ponencias-1/5602-s19-d03-06-teletrabajo-mas-alla-de-las-vpns/file.html

- Ref. 11: Resultados confinamiento en concienciación por ciberseguridad

https://revistabyte.es/ciberseguridad/el-teletrabajo-ciberseguridad/

- Ref. 12: Consejos para teletrabajar de forma segura

https://www.ccn-cert.cni.es/comunicacion-eventos/comunicados-ccn-cert/9941-como-teletrabajar-de-forma-segura-sin-poner-en-riesgo-a-usuarios-y-organizaciones.html

- Ref. 13: Definición de Sistema de detección de Intrusos

https://es.wikipedia.org/wiki/Sistema_de_detecci%C3%B3n_de_intrusos

- Ref.14: Windows Firewall

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/windows-firewall-with-advanced-security

- Ref.15: Design Thinking Methodology:

https://en.wikipedia.org/wiki/Design_thinking

https://designthinking.es/inicio/index.php

https://slidemodel.com/templates/design-thinking-lean-startup-agile-diagram-for-powerpoint/

- Ref. 16: Lean Startup:

https://es.wikipedia.org/wiki/Lean_startup

- Ref. 17: Scrum como referente marco de trabajo ágil:

https://es.wikipedia.org/wiki/Scrum_%28desarrollo_de_software%29

- Ref. 18: Detalle Design Thinking

https://www.luisan.net/blog/diseno-grafico/que-es-design-thinking

- Ref. 19: Libro electrónico Sistemas de Detección de Intrusiones 1.01. Diego González Gómez.

http://www.criptored.upm.es/guiateoria/gt_m481a.htm

- Ref. 20: Referencia de productos IDS

https://www.softwaretestinghelp.com/intrusion-detection-systems/

- Ref. 21: "Ventajas e Implementación de un sistema IDS/SIEM en el ámbito familiar", José Antonio Salom Martín. 2019.
- Ref. 22: Blog "Un informático en el lado del mal" de Chema Alonso. Artículo Hackeando al vecino Hax0r que me roba el wifi.

https://www.elladodelmal.com/2013/04/hackeando-al-vecino-hax0r-que-me-roba.html

- Ref 23. Instalación de Raspberry Pi 3+

https://turinconinformatico.com/raspberrypi/instalar-raspbian/

Ref 24. Instalación punto de acceso WIFI en Raspberry PI3

https://www.raspberrypi.org/forums/viewtopic.php?t=211542

https://howchoo.com/pi/setup-a-raspberry-pi-wireless-access-point

https://www.raspberrypi.org/documentation/configuration/wireless/access-point-routed.md

https://learn.sparkfun.com/tutorials/setting-up-a-raspberry-pi-3-as-an-access-point/all

https://thepi.io/how-to-use-your-raspberry-pi-as-a-wireless-access-point/

- Ref 25. Instalación de Suricata

https://blog.elhacker.net/2021/02/instalar-configurar-reglas-ids-ips-suricata-en-una-raspberry-pi.html

- Ref.26 Suricata vs Snort

https://tacticalflex.zendesk.com/hc/en-us/articles/360010678893-Snort-vs-Suricata

- Ref. 27 Suricata vs Zeek

https://bricata.com/blog/suricata-or-zeek-the-answer-is-both/

- Ref.28 Bricata Platform

https://bricata.com/

- Ref. 29 Comparativa NIDS, HIDS

https://cybersecurity.att.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

- Ref. 30 Carácterísticas Raspberry PI3

https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando

- Ref. 31 Instalación FileBeat en Raspberry

https://bløgg.no/2020/11/filebeat-on-a-raspberry-pi/

- Ref. 32 ELK

https://www.elastic.co/es/what-is/elk-stack

- Ref. 33: ELK componentes

https://www.adictosaltrabajo.com/2014/09/12/primeros-pasos-elasticsearch/

- Ref, 34: Logstash

https://www.adictosaltrabajo.com/2015/09/17/logstash/

- Ref. 35 Beats

https://www.davincigroup.es/beats-elastic-ejemplo-filebeat/

- Ref. 36: SIEM

https://www.ibm.com/topics/siem

- Ref. 37: Tendencias en Ciberseguridad para el 2021.

https://www.welivesecurity.com/wp-content/uploads/2020/12/Cybersecurity_Trends_2021_ES.pdf

- Ref. 38: Cisco OpenDNS

https://www.opendns.com/cisco-opendns/

- Ref. 39: Emerging threads ruleset

https://rules.emergingthreats.net/OPEN_download_instructions.html

- Ref. 40. Artificial Intelligence based Network Intrusion Detection with Hyper-parameter optimisation tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing.

https://www.sciencedirect.com/science/article/pii/S2405959518305976

- Ref. 41: Intrusion Detection System Based on Artificial Intelligence Techniques in Wireless Sensor Networks

https://journals.sagepub.com/doi/full/10.1155/2013/351047

- Ref. 42: An Intrusion-Detection Model. Dorothy Denning. 1987. IEEE Transactions on Software Engineering, Vol. SE-13, NO.2

https://www.cs.colostate.edu/~cs656/reading/ieee-se-13-2.pdf

6. Anexos

6.1 Teletrabajo

Durante la pandemia, las empresas, se vieron en la situación de pedir a sus trabajadores usar sus equipos informáticos personales en casa para realizar teletrabajo.

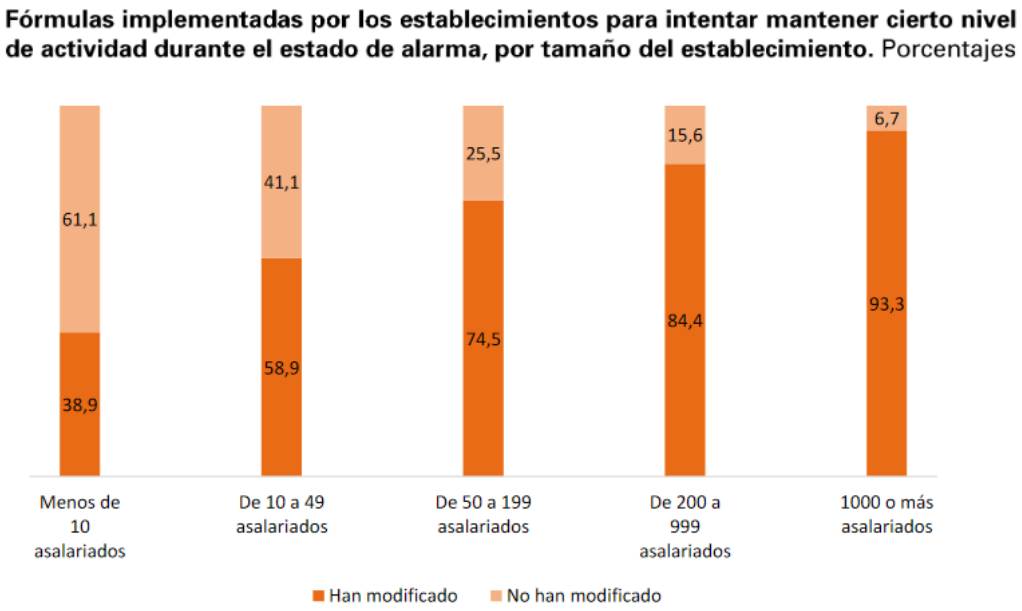

Ilustración 38: Fórmulas implementadas en estado de alarma. Por tamaño. (ref. 6)

El trabajo en remoto (ref.6) fue usado por el 48,8% de las empresas para mantener su actividad durante el tiempo de la pandemia, seguido de un incremento en el nivel de digitalización del 15,11%. Además del teletrabajo, se han introducido durante la pandemia, el servicio a domicilio (16,6%) y el comercio electrónico (16,4%).

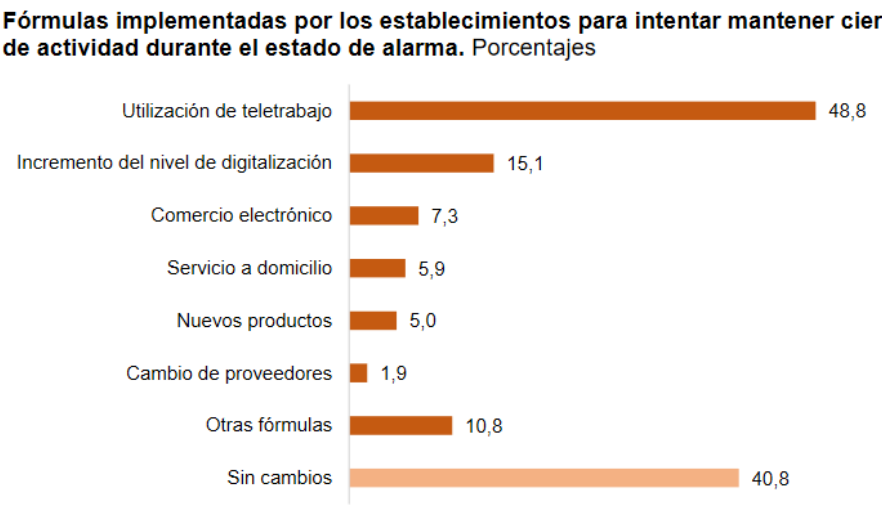

Ilustración 39: Fórmulas implementadas por los establecimientos para mantener la actividad. (ref. 6)

6.2 NetSetup

A continuación, se ofrece el script completo usado para la configuración del Punto de Acceso:

```
#! /bin/bash
set -e

# Author: Pankaj Shelare
# Email: pankaj.shelare@gmail.com

# This script is created and enhanced using the thoughts of the below given link reference.
# REFERENCE: https://lb.raspberrypi.org/forums/viewtopic.php?t=211542
# Although, the script is created using the thought process of the above link,
# there are many enhancements made to solve BUGS (occurred during evaluation and testing phase)
# and to promote many advanced features to automate and setup the single WiFi chip of
# Raspberry Pi as an Access Point(AP) and Station(STA) Network both (and hence, supporting
# HOTSPOT feature in Raspberry Pi using the execution of this script).

# BUG FIXES:
# TODO: DHCPCD created problem in new Buster OS so, while upgrating OS make sure that, DHCPCD
# and other dependancies are locked so that, it will have less conflict in re-installation.
# This is presently not done but, it can be done using the below command. This feature will
# be release in future version of the script.
# sudo apt-mark hold dhcpcd5

# Store all input options in an array:
options=("$@")
#echo "Input options are: ${options[@]}"

OS_VERSION=`cat /etc/os-release 2>/dev/null | grep -i "VERSION_ID" | awk -F '=' '{gsub("\"", "", $2);
print $2}'`
```

```
echo ""

echo "[INFO]: Processing network setup for OS Version: $OS_VERSION"


apIpDefault="10.0.0.1"

apDhcpRangeDefault="10.0.0.50,10.0.0.150,12h"

apSetupIptablesMasqueradeDefault="iptables -t nat -A POSTROUTING -s 10.0.0.0/24 ! -d 10.0.0.0/24 -j
MASQUERADE"

apCountryCodeDefault="IN"

apChannelDefault="1"


apIp="$apIpDefault"

apDhcpRange="$apDhcpRangeDefault"

apSetupIptablesMasquerade="$apSetupIptablesMasqueradeDefault"

apCountryCode="$apCountryCodeDefault"

apChannel="$apChannelDefault"

apSsid=""

apPassphrase=""

apPasswordConfig=""


# REFERENCE: Country codes taken from: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

countryCodeArray=('AD', 'AE', 'AF', 'AG', 'AI', 'AL', 'AM', 'AO', 'AQ', 'AR', 'AS', 'AT', 'AU', 'AW',
'AX', 'AZ',

'BA', 'BB', 'BD', 'BE', 'BF', 'BG', 'BH', 'BI', 'BJ', 'BL', 'BM', 'BN', 'BO', 'BQ', 'BR', 'BS', 'BT',
'BV', 'BW',

'BY', 'BZ', 'CA', 'CC', 'CD', 'CF', 'CG', 'CH', 'CI', 'CK', 'CL', 'CM', 'CN', 'CO', 'CR', 'CU', 'CV',
'CW', 'CX',

'CY', 'CZ', 'DE', 'DJ', 'DK', 'DM', 'DO', 'DZ', 'EC', 'EE', 'EG', 'EH', 'ER', 'ES', 'ET', 'FI', 'FJ',
'FK', 'FM',

'FO', 'FR', 'GA', 'GB', 'GD', 'GE', 'GF', 'GG', 'GH', 'GI', 'GL', 'GM', 'GN', 'GP', 'GQ', 'GR', 'GS',
'GT', 'GU',

'GW', 'GY', 'HK', 'HM', 'HN', 'HR', 'HT', 'HU', 'ID', 'IE', 'IL', 'IM', 'IN', 'IO', 'IQ', 'IR', 'IS',
'IT', 'JE',

'JM', 'JO', 'JP', 'KE', 'KG', 'KH', 'KI', 'KM', 'KN', 'KP', 'KR', 'KW', 'KY', 'KZ', 'LA', 'LB', 'LC',
'LI', 'LK',

'LR', 'LS', 'LT', 'LU', 'LV', 'LY', 'MA', 'MC', 'MD', 'ME', 'MF', 'MG', 'MH', 'MK', 'ML', 'MM', 'MN',
'MO', 'MP',

'MQ', 'MR', 'MS', 'MT', 'MU', 'MV', 'MW', 'MX', 'MY', 'MZ', 'NA', 'NC', 'NE', 'NF', 'NG', 'NI', 'NL',
'NO', 'NP',

'NR', 'NU', 'NZ', 'OM', 'PA', 'PE', 'PF', 'PG', 'PH', 'PK', 'PL', 'PM', 'PN', 'PR', 'PS', 'PT', 'PW',
'PY', 'QA',

'RE', 'RO', 'RS', 'RU', 'RW', 'SA', 'SB', 'SC', 'SD', 'SE', 'SG', 'SH', 'SI', 'SJ', 'SK', 'SL', 'SM',
'SN', 'SO',

'SR', 'SS', 'ST', 'SV', 'SX', 'SY', 'SZ', 'TC', 'TD', 'TF', 'TG', 'TH', 'TJ', 'TK', 'TL', 'TM', 'TN',
'TO', 'TR',

'TT', 'TV', 'TW', 'TZ', 'UA', 'UG', 'UM', 'US', 'UY', 'UZ', 'VA', 'VC', 'VE', 'VG', 'VI', 'VN', 'VU',
'WF', 'WS',

'YE', 'YT', 'ZA', 'ZM', 'ZW')


workDir="/home/pi"

installDir="$workDir/network-setup"

logDir="$installDir/log"

execDir="$installDir/bin"

downloadDir="$installDir/downloads"

netStartFile="$execDir/netStart"
```

```
netStopFile="$execDir/netStop.sh"

netLogFile="$logDir/network.log"

netStopServiceFile="/etc/systemd/system/netStop.service"

netStationConfigFile="/etc/network/interfaces.d/station"

netShutdownFlagFile="$logDir/netShutdownFlag"

shutdownRecoveryFile="$execDir/shutdownRecovery"

rcLocalLogFile="$logDir/rc.local.log"

cleanup=false

install=false

installUpgrade=false

apSsidValid=false

apPassphraseValid=false

apCountryCodeValid=true

apIpAddrValid=true

rebootFlag=true

wlanInterfaceNameValid=true

# Defined common WLAN and AP Interface names here as in the recent and future versions of Debian
based OS

# may change the Networking Interface name.

wlanInterfaceNameDefault="wlan0"

wlanInterfaceName="$wlanInterfaceNameDefault"

apInterfaceName="uap0"

hostNameDefault="raspberrypi"

hostName="$hostNameDefault"

# FIX: for https://github.com/idev1/rpihotspot/issues/12#issuecomment-605552834

if [ ! -z "$( hostname )" ]; then

hostName="$( hostname )"

fi

echo "[INFO]: Hostname is: $hostName"

function setWlanDetails()

{

# Set Country Code:

wlanCountryCode="$( cat /etc/wpa_supplicant/wpa_supplicant.conf | grep -i 'country=' | awk -F '='
'{print $2}' )"

# FIX: #12, trimming spaces and carriage return for WLAN Country code if there are any.

wlanCountryCode="$( echo $wlanCountryCode | tr -d '\r' )"

if [[ ! -z "${wlanCountryCode}" && \

("${countryCodeArray[@]}" =~ "${wlanCountryCode}") ]]; then

apCountryCode="$wlanCountryCode"

apCountryCodeDefault="$wlanCountryCode"

fi

# Read WiFi Station(${wlanInterfaceName}) IP, Mask and Broadcast addresses:

read wlanIpAddr wlanIpMask wlanIpCast <<< $( echo $( ifconfig ${wlanInterfaceName} | grep 'inet ' ) |
```

```
awk -F " " '{print $2" "$4" "$6}' )

# Set AP Channel:

wlanChannel="$( iwlist ${wlanInterfaceName} channel | grep 'Current Frequency:' | awk -F '('
'{gsub("\)", "", $2); print $2}' | awk -F ' ' '{print $2}' )"

if [ ! -z "${wlanChannel}" ]; then

apChannel="$wlanChannel"

apChannelDefault="$wlanChannel"

fi

}

setWlanDetails

# REFERENCE: https://en.wikipedia.org/wiki/Private_network#Private_IPv4_addresses

# Visit above site to know more about Reserved Private IP Address for LAN/WLAN communication.

function validIpAddress()

{

local ip=$1

local status=1

if [[ $ip =~ ^(10|172|192)\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]]; then

IFS='.' read ipi1 ipi2 ipi3 ipi4 <<< "$ip"

IFS='.' read -r -a wlanIpMaskArr <<< "$wlanIpMask"

IFS='.' read -r -a wlanIpAddrArr <<< "$wlanIpAddr"

wlanIpStartWith=""

wlanIpStartWithCount=0

for i in ${!wlanIpMaskArr[@]}; do

mskVal=${wlanIpMaskArr[$i]}

if [ $mskVal == 255 ]; then

if [ -z "$wlanIpStartWith" ]; then

wlanIpStartWith="${wlanIpAddrArr[$i]}"

else

wlanIpStartWith="$wlanIpStartWith.${wlanIpAddrArr[$i]}"

fi

wlanIpStartWithCount=$((wlanIpStartWithCount+1))

fi

done

wlanIpStartWith="$wlanIpStartWith."

case $ipi1 in

10)

[[ ( $ip != $wlanIpAddr && ! $ip =~ ${wlanIpStartWith}* ) && \

((${#ipi2} -eq 1 && ${ipi2} -le 255) || (${#ipi2} -gt 1 && "${ipi2}" != 0* && ${ipi2} -le 255)) && \

((${#ipi3} -eq 1 && ${ipi3} -le 255) || (${#ipi3} -gt 1 && "${ipi3}" != 0* && ${ipi3} -le 255)) && \

((${#ipi4} -eq 1 && ${ipi4} -le 255) || (${#ipi4} -gt 1 && "${ipi4}" != 0* && ${ipi4} -le 255))

]]

status=$?
```

```
;;
172)
[[ ( $ip != $wlanIpAddr && ! $ip =~ ${wlanIpStartWith}* ) && \
("${ipi2}" != 0* && ${ipi2} -ge 16 && ${ipi2} -le 31) && \
((${#ipi3} -eq 1 && ${ipi3} -le 255) || (${#ipi3} -gt 1 && "${ipi3}" != 0* && ${ipi3} -le 255)) && \
((${#ipi4} -eq 1 && ${ipi4} -le 255) || (${#ipi4} -gt 1 && "${ipi4}" != 0* && ${ipi4} -le 255))
]]
status=$?
;;
192)
[[ ( $ip != $wlanIpAddr && ! $ip =~ ${wlanIpStartWith}* ) && \
("${ipi2}" != 0* && ${ipi2} -eq 168) && \
((${#ipi3} -eq 1 && ${ipi3} -le 255) || (${#ipi3} -gt 1 && "${ipi3}" != 0* && ${ipi3} -le 255)) && \
((${#ipi4} -eq 1 && ${ipi4} -le 255) || (${#ipi4} -gt 1 && "${ipi4}" != 0* && ${ipi4} -le 255))
]]
status=$?
;;
esac
fi
return $status
}

# Check first if a valid --wifi-interface name option is provided or not.
# If a valid --wifi-interface name option is provided then, reset the WLAN details
# by calling the function: setWlanDetails.
for i in ${!options[@]}; do

option="${options[$i]}"

if [[ "$option" == --wifi-interface=* ]]; then
wlanInterfaceNameTemp="$(echo $option | awk -F '=' '{print $2}')"
if [ ! -z "$wlanInterfaceNameTemp" ]; then
if [ "$(iwlist $wlanInterfaceNameTemp scan 2>/dev/null)" ]; then
wlanInterfaceNameValid=true
wlanInterfaceName="$wlanInterfaceNameTemp"
setWlanDetails
else
wlanInterfaceNameValid=false
wlanInterfaceName="$wlanInterfaceNameDefault"
fi
fi
fi

done

# Process input options other than --wifi-interface.
for i in ${!options[@]}; do
```

```
option="${options[$i]}"

if [ "$option" = "--clean" ]; then
cleanup=true
fi

if [ "$option" = "--install" ]; then
install=true
fi

if [ "$option" = "--install-upgrade" ]; then
installUpgrade=true
fi

if [[ "$option" == --ap-ssid=* ]]; then
apSsid="$(echo $option | awk -F '=' '{print $2}')"
if [[ "$apSsid" =~ ^[A-Za-z0-9_-]{3,}$ ]]; then
apSsidValid=true
fi
fi

if [[ "$option" == --ap-password=* ]]; then
apPassphrase="$(echo $option | awk -F '=' '{print $2}')"
if [[ "$apPassphrase" =~ ^[A-Za-z0-9@#$%^\&*_+-]{8,}$ ]]; then
apPassphraseValid=true
apPasswordConfig="wpa_passphrase=$apPassphrase"
fi
fi

if [[ "$option" == --ap-country-code=* ]]; then
apCountryCodeTemp="$(echo $option | awk -F '=' '{print $2}')"
if [ ! -z "$apCountryCodeTemp" ]; then
if [[ "${countryCodeArray[@]}" =~ "${apCountryCodeTemp}" ]]; then
if [[ ! -z "${wlanCountryCode}" && \
(( ! "${countryCodeArray[@]}" =~ "${wlanCountryCode}") || \
( ! "${apCountryCodeTemp}" =~ "${wlanCountryCode}")) ]]; then
apCountryCodeValid=false
else
apCountryCodeValid=true
apCountryCode="$apCountryCodeTemp"
fi
else
apCountryCodeValid=false
fi
fi
fi

if [[ "$option" == --ap-ip-address=* ]]; then
apIpAddrTemp="$(echo $option | awk -F '=' '{print $2}')"
```

```
if [ ! -z "$apIpAddrTemp" ]; then

if validIpAddress "$apIpAddrTemp"; then

apIpAddrValid=true

# Successful validation. Now set apIp, apDhcpRange and apSetupIptablesMasquerade:

apIp="$apIpAddrTemp"

IFS='.' read -r -a apIpArr <<< "$apIp"

apIpSubnetSize=24

apIpFirstThreeDigits="${apIpArr[0]}.${apIpArr[1]}.${apIpArr[2]}"

apIpLastDigit=${apIpArr[3]}

div=$((apIpLastDigit/100))

minCalcDigit=1

maxCalcDigit=100


case $div in

# Between (0-99)

0) minCalcDigit=$((apIpLastDigit+1)); maxCalcDigit=$((minCalcDigit+100)) ;;

# Between (100-199)

1) minCalcDigit=$((200-apIpLastDigit)); maxCalcDigit=$((minCalcDigit+100)) ;;

# Between (200-255)

2) minCalcDigit=$((256-apIpLastDigit)); maxCalcDigit=$((minCalcDigit+100)) ;;

*) minCalcDigit=1; maxCalcDigit=100 ;;

esac


case ${apIpArr[0]} in

10) apIpSubnetSize=24 ;;

172) apIpSubnetSize=20 ;;

192) apIpSubnetSize=16 ;;

*) apIpSubnetSize=24 ;;

esac


apDhcpRange="${apIpFirstThreeDigits}.${minCalcDigit},${apIpFirstThreeDigits}.${maxCalcDigit},12h"

apSetupIptablesMasquerade="iptables -t nat -A POSTROUTING -s
${apIpFirstThreeDigits}.0/${apIpSubnetSize} ! -d ${apIpFirstThreeDigits}.0/${apIpSubnetSize} -j
MASQUERADE"

else

apIpAddrValid=false

fi

fi

fi


done


# Process AP Password encryption:

for i in ${!options[@]}; do

option="${options[$i]}"

if [ "$apSsidValid" = true -a "$apPassphraseValid" = true -a "$option" = "--ap-password-encrypt" ];
then

apWpaPsk="$( wpa_passphrase ${apSsid} ${apPassphrase} | awk '{$1=$1};1' | grep -P '^psk=' | awk -F
'=' '{print $2}' )"
```

```
apPasswordConfig="wpa_psk=$apWpaPsk"
fi
done

doRemoveDisableIPv6Setup() {
result=$(sed -n '/^#__IPv6_SETUP_START__/,/^#__IPv6_SETUP_END__/p' /etc/sysctl.conf)
if [ ! -z "$result" ]; then
echo "[Remove]: IPv6 config from /etc/sysctl.conf"
sed '/^#__IPv6_SETUP_START__/,/^#__IPv6_SETUP_END__/d' /etc/sysctl.conf > ./tmp.conf
rm -f /etc/sysctl.conf
mv ./tmp.conf /etc/sysctl.conf
rm -f ./tmp.conf
fi
}

doAddDisableIPv6Setup() {
doRemoveDisableIPv6Setup
cat >> /etc/sysctl.conf <<EOF

#__IPv6_SETUP_START__
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
net.ipv6.conf.lo.disable_ipv6=1
net.ipv6.conf.eth0.disable_ipv6=1
net.ipv6.conf.${wlanInterfaceName}.disable_ipv6=1
#__IPv6_SETUP_END__
EOF

}

# FIX: Raspbian Buster OS creating problem while reloading dhcpcd.service after cleanup.
# This is causing because of IPv6 and hence, disabling IPv6.
# You can enable IPv6 again by calling doRemoveDisableIPv6Setup() function.
# doAddDisableIPv6Setup

# Create initial directories:
mkdir -p $installDir
mkdir -p $logDir
mkdir -p $execDir
mkdir -p $downloadDir

doRemoveDhcpdApSetup() {
# May work with this pattern also: /^#__AP_SETUP_START__/,/^#__AP_SETUP_END__/p;/^#__AP_SETUP_END__/q
result=$(sed -n '/^#__AP_SETUP_START__/,/^#__AP_SETUP_END__/p' /etc/dhcpcd.conf)
if [ ! -z "$result" ]; then
echo "[Remove]: AP config from /etc/dhcpcd.conf"
sed '/^#__AP_SETUP_START__/,/^#__AP_SETUP_END__/d' /etc/dhcpcd.conf > ./tmp.conf
rm -f /etc/dhcpcd.conf
mv ./tmp.conf /etc/dhcpcd.conf
```

```
rm -f ./tmp.conf
fi
}

doAddDhcpdApSetup() {
doRemoveDhcpdApSetup
cat >> /etc/dhcpcd.conf <<EOF

#__AP_SETUP_START__
interface ${apInterfaceName}
static ip_address=$apIp
nohook wpa_supplicant
#__AP_SETUP_END__
EOF

}

doRemoveRcLocalNetStartSetup() {
if [ $(cat /etc/rc.local 2>/dev/null | grep -c "$netStartFile") -gt 0 ]; then
echo "[Remove]: entry -> '$netStartFile' from /etc/rc.local"
sed '/netStart/d' /etc/rc.local > ./tmp.conf
rm -f /etc/rc.local
mv ./tmp.conf /etc/rc.local
rm -f ./tmp.conf
fi
}

doAddRcLocalNetStartSetup() {
doRemoveRcLocalNetStartSetup
sed '/exit 0/d' /etc/rc.local > ./tmp.conf
echo "/bin/bash $netStartFile
exit 0" >> ./tmp.conf
rm -f /etc/rc.local
mv ./tmp.conf /etc/rc.local
rm -f ./tmp.conf
}

doRemoveApIpEntriesFromHostFile() {
if [ `cat /etc/hosts | grep -c ^10.` -gt 0 -o \
`cat /etc/hosts | grep -c ^172.` -gt 0 -o \
`cat /etc/hosts | grep -c ^192.168.` -gt 0 ]; then
sed '/^10./d;/^172./d;/^192.168./d' /etc/hosts > ./tmp.conf
mv ./tmp.conf /etc/hosts
rm -f ./tmp.conf
echo "[Cleanup]: Cleaned all AP IP entries from /etc/hosts file."
fi
}
```

```
doAddApIpEntriesToHostFile() {

cat > /etc/hosts <<EOF
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

127.0.1.1 $hostName
$apIp $hostName
EOF

}

doRemoveIpTableNatEntries() {
# Clean other network entries:
#iw dev uap0 del
apDelCmd='iw dev '${apInterfaceName}' del'
bash -c '$apDelCmd'
iptables -F
iptables -t nat -F
bash -c 'cat /dev/null > /etc/iptables.ipv4.nat'
bash -c 'cat /dev/null > /proc/sys/net/ipv4/ip_forward'
sed -i 's/^net.ipv4.ip_forward=.*$/#net.ipv4.ip_forward=1/' /etc/sysctl.conf
echo "[Cleanup]: Cleaned all NAT IP Table entries."
}

doRestartSysDaemon() {
if [ ! `sudo systemctl status dhcpcd 2> /dev/null | grep "systemctl daemon-reload"` ]; then
systemctl daemon-reload
echo "[Restart]: System Daemon restarted!"
fi
}

doAptClean() {
apt-get clean
apt-get autoclean -y
apt-get autoremove -y
echo "[Cleanup]: apt-get clean/autoremove done."
}

doCleanup() {
echo "[Cleanup]: cleaning ..."

# Do apt-get clean:
doAptClean

# Cleanup: /etc/dhcpcd.conf
doRemoveDhcpdApSetup
```

```
# Cleanup: /etc/rc.local

doRemoveRcLocalNetStartSetup


if [ $(dpkg-query -W -f='${Status}' hostapd 2>/dev/null | grep -c "ok installed") -eq 1 ]; then

echo "[Remove]: hostapd"

apt-get purge -y hostapd

# FIX: broken link if purge did not remove the dirctory as the dirctory is not empty and the
directory has user-data.

if [ -d "/etc/hostapd" ]; then

rm -rf /etc/hostapd*

echo "[Remove]: Forcibly removed directory: /etc/hostapd."

fi

fi


if [ $(dpkg-query -W -f='${Status}' dnsmasq 2>/dev/null | grep -c "ok installed") -eq 1 ]; then

echo "[Remove]: dnsmasq"

apt-get purge -y dnsmasq

# FIX: broken link if purge did not remove the dirctory as the dirctory is not empty and the
directory has user-data.

if [ -d "/etc/dnsmasq.d" ]; then

rm -rf /etc/dnsmasq*

echo "[Remove]: Forcibly removed directory: /etc/dnsmasq.d and all related dnsmasq files."

fi

fi


# FIX: In Buster OS, dns-root-data creating problem while installing dnsmasq and hence, purge
required for dns-root-data:

if [ $(dpkg-query -W -f='${Status}' dns-root-data 2>/dev/null | grep -c "ok installed") -eq 1 ]; then

echo "[Remove]: dns-root-data"

apt-get purge -y dns-root-data

fi


if [ $(dpkg-query -W -f='${Status}' iptables-persistent 2>/dev/null | grep -c "ok installed") -eq 1
]; then

echo "[Remove]: iptables-persistent"

apt-get purge -y iptables-persistent

fi


if [ -f "$netStationConfigFile" ]; then

echo "[Remove]: $netStationConfigFile"

rm -f $netStationConfigFile

fi


if [ -f "/etc/dnsmasq.conf" ]; then

echo "[Remove]: /etc/dnsmasq.conf"

rm -f /etc/dnsmasq.conf

fi


if [ -f "/etc/dnsmasq.conf.orig" ]; then

echo "[Remove]: /etc/dnsmasq.conf.orig"

rm -f /etc/dnsmasq.conf
```

```
fi

if [ -f "/etc/hostapd/hostapd.conf" ]; then
echo "[Remove]: /etc/hostapd/hostapd.conf"
rm -f /etc/hostapd/hostapd.conf
fi

if [ -f "/etc/default/hostapd" ]; then
echo "[Remove]: /etc/default/hostapd"
rm -f /etc/default/hostapd
fi

if [ $(systemctl list-unit-files --type=service 2>/dev/null | grep -c 'netStop.service') -gt 0 ];
then
systemctl stop netStop.service
systemctl disable netStop.service
echo "[Remove]: stop/disable service -> netStop"
fi

if [ -f "$netStopServiceFile" ]; then
echo "[Remove]: $netStopServiceFile"
rm -f $netStopServiceFile
fi

if [ -f "$netStartFile" ]; then
echo "[Remove]: $netStartFile"
rm -f $netStartFile
fi

if [ -f "$netStopFile" ]; then
echo "[Remove]: $netStopFile"
rm -f $netStopFile
fi

if [ -f "$netLogFile" ]; then
echo "[Remove]: $netLogFile"
rm -f $netLogFile
fi

if [ -f "$rcLocalLogFile" ]; then
echo "[Remove]: $rcLocalLogFile"
rm -f $rcLocalLogFile
fi

if [ -f "$shutdownRecoveryFile" ]; then
echo "[Remove]: $shutdownRecoveryFile"
rm -f $shutdownRecoveryFile
fi

if [ -f "$netShutdownFlagFile" ]; then
```

```
echo "[Remove]: $netShutdownFlagFile"

rm -f $netShutdownFlagFile

fi

doRemoveIpTableNatEntries

# FIX: for https://github.com/idev1/rpihotspot/issues/12#issuecomment-605552834

doRemoveApIpEntriesFromHostFile

# Clean and auto remove the previously install dependant component if they exists by improper
purging.

doAptClean

#Restart DHCPCD service:

# FIX: it seems daemon-reload required on Buster OS+ as the dhcpcd don't start by default

# if the dhcpcd service unit is changed and then, it wait for sometime indicating that a

# daemon-restart is required.

doRestartSysDaemon

systemctl restart dhcpcd

# FIX: For Buster OS, forcibly enabling dhcpcd if its previously disabled.

if [ $OS_VERSION == 10 ]; then

systemctl enable dhcpcd

echo "[Cleanup]: Forcibly enabled dhcpcd."

fi

sleep 5

echo "[Cleanup]: DONE"

}

downloadReqDependancies() {

apt-get update --fix-missing

if [ "$installUpgrade" = true ]; then

apt-get upgrade -y --fix-missing

apt-get dist-upgrade -y

fi

apt-get install -y hostapd dnsmasq iptables-persistent

}

isAvailableReqDependancies() {

[ $(dpkg-query -W -f='${Status}' hostapd 2>/dev/null | grep -c "ok installed") -eq 1 -a \

$(dpkg-query -W -f='${Status}' dnsmasq 2>/dev/null | grep -c "ok installed") -eq 1 -a \

$(dpkg-query -W -f='${Status}' iptables-persistent 2>/dev/null | grep -c "ok installed") -eq 1 ]

status=$?

return $status

}

doInstall() {

echo ""

echo "[WLAN]: ${wlanInterfaceName} IP address: $wlanIpAddr"
```

```
echo "[WLAN]: ${wlanInterfaceName} IP Mask address: $wlanIpMask"
echo "[WLAN]: ${wlanInterfaceName} IP Broadcast address: $wlanIpCast"
echo "[WLAN]: ${wlanInterfaceName} Country Code: $wlanCountryCode"
echo "[WLAN]: ${wlanInterfaceName} Channel: $wlanChannel"

doCleanup

touch $netLogFile
chmod ug+w $netLogFile

#Silent install iptables:
echo iptables-persistent iptables-persistent/autosave_v4 boolean true | debconf-set-selections
echo iptables-persistent iptables-persistent/autosave_v6 boolean true | debconf-set-selections

#If Internet is available then, install hostapd, dnsmasq, iptables-persistent from internet:
if [ $(curl -Is http://www.google.com 2>/dev/null | head -n 1 | grep -c '200 OK') -gt 0 ]; then
echo "[Install]: installing: hostapd dnsmasq iptables-persistent from net ..."
downloadReqDependancies
else
if [ -f $downloadDir/1_libnl-route-3-200.deb -a \
-f $downloadDir/2_hostapd.deb -a \
-f $downloadDir/3_libnfnetlink0.deb -a \
-f $downloadDir/4_dnsmasq-base.deb -a \
-f $downloadDir/5_dnsmasq.deb -a \
-f $downloadDir/6_netfilter-persistent.deb -a \
-f $downloadDir/7_iptables-persistent.deb ]; then
echo "[Install]: installing: hostapd dnsmasq iptables-persistent from local available dependencies
..."
dpkg --install $downloadDir/1_libnl-route-3-200.deb
dpkg --install $downloadDir/2_hostapd.deb
dpkg --install $downloadDir/3_libnfnetlink0.deb
dpkg --install $downloadDir/4_dnsmasq-base.deb
dpkg --install $downloadDir/5_dnsmasq.deb
dpkg --install $downloadDir/6_netfilter-persistent.deb
dpkg --install $downloadDir/7_iptables-persistent.deb
fi
fi

# FIX: For issue #13, Raspbian Buster OS unable to correct nameserver entry in /etc/resolv.conf
hence,
# need to correct this entry for downloading the files again:
if ! isAvailableReqDependancies; then
if [ ! `sudo cat /etc/resolv.conf 2>/dev/null | grep "8.8.8.8"` ]; then
echo "nameserver 8.8.8.8" >> /etc/resolv.conf
echo "[Install]: Google nameserver 8.8.8.8 added into /etc/resolv.conf."
echo "[Install]: Now retrying 2nd time to download required dependancies ..."
downloadReqDependancies
fi
fi
```

```
if ! isAvailableReqDependancies; then

echo "[Install]: Required dependancies: hostapd, dnsmasq and iptables-persistent are not available.
Please check your internet connection!"

echo "[Install]: Installation FAILED! Exiting installation now.."

exit 0

fi


systemctl stop hostapd

systemctl stop dnsmasq


doAddDhcpdApSetup


if [ ! -f "/etc/dnsmasq.conf.orig" ]; then

echo "[Move]: /etc/dnsmasq.conf to /etc/dnsmasq.conf.orig"

mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig

fi


cat > /etc/dnsmasq.conf <<EOF

interface=lo,${apInterfaceName} #Use interfaces lo and ${apInterfaceName}

no-dhcp-interface=lo,${wlanInterfaceName}

bind-interfaces #Bind to the interfaces

server=8.8.8.8 #Forward DNS requests to Google DNS

#domain-needed #Don't forward short names

bogus-priv #Never forward addresses in the non-routed address spaces

dhcp-range=$apDhcpRange

EOF


cat > /etc/hostapd/hostapd.conf <<EOF

channel=$apChannel

ssid=$apSsid

$apPasswordConfig

country_code=$apCountryCode

interface=${apInterfaceName}

# Use the 2.4GHz band (I think you can use in ag mode to get the 5GHz band as well, but I have not
tested this yet)

hw_mode=g

# Accept all MAC addresses

macaddr_acl=0

# Use WPA authentication

auth_algs=1

# Require clients to know the network name

ignore_broadcast_ssid=0

# Use WPA2

wpa=2

# Use a pre-shared key

wpa_key_mgmt=WPA-PSK

wpa_pairwise=TKIP

rsn_pairwise=CCMP
```

```
#driver=nl80211
# I commented out the lines below in my implementation, but I kept them here for reference.
# Enable WMM
#wmm_enabled=1
# Enable 40MHz channels with 20ns guard interval
#ht_capab=[HT40][SHORT-GI-20][DSSS_CCK-40]
EOF

sed -i 's/^#DAEMON_CONF=.*$/DAEMON_CONF="\/etc\/hostapd\/hostapd.conf"/' /etc/default/hostapd

cat > $netStationConfigFile <<EOF
allow-hotplug ${wlanInterfaceName}
EOF

# Create shutdown recovery script when last time shutdown did not go well.
cat > $shutdownRecoveryFile <<EOF
# ----------------------------------------------------------------------------------------------
# IMPORTANT:
# ----------------------------------------------------------------------------------------------
# Improper shutdown/reboot by directly switching of the device or taking off the power plug
# may result in malfuctioning of Access Point (AP) Network setup or may harm other
# functionalies of the application. Hence, below script will ensure improper shutdown recovery.
# You can disable this feature by setting: 'rebootFlag=false' or 'rebootFlag=n' in this script
# or in main script: 'setup-network.sh'.
# ----------------------------------------------------------------------------------------------

if [ ! -f "$netShutdownFlagFile" ]; then
#sudo bash -c 'echo "\$(date +"%Y-%m-%d %T") - [WARNING]: Last time shutdown did not happen
properly!" >> $netLogFile'
echo "[WARNING]: Last shutdown errors may affect Access Point(AP) Network to become non-functional!"
echo "[SOLUTION]: Reboot system to solve the shutdown errors."
#read -n 1 -p "Reboot System [y/n]: " "rebootFlag"
if [ "$rebootFlag" = "y" -o "$rebootFlag" = true ]; then
sudo $netStopFile
echo "Rebooting in 5 seconds ..."
sleep 5
sudo reboot
fi
elif [ -f "$netShutdownFlagFile" ]; then
sudo rm -f $netShutdownFlagFile
fi

EOF

chmod ug+x $shutdownRecoveryFile

# Create startup script
cat > $netStartFile <<EOF

# Check shutdown flag file exists for proper last time shutdown
```

```
# and if last time shutdown did not happen properly then reboot to make sure that,

# netStop.service properly do the necessary things before shutdown:


# Output the standard errors and messages of rc.local executions to rc.local.log file.

exec 2> $rcLocalLogFile

exec 1>&2


# Attach script for improper shutdown recovery:

source $shutdownRecoveryFile


#Make sure no ${apInterfaceName} interface exists (this generates an error; we could probably use an
if statement to check if it exists first)

echo "Removing ${apInterfaceName} interface..."

iw dev ${apInterfaceName} del


#Add ${apInterfaceName} interface (this is dependent on the wireless interface being called
${wlanInterfaceName}, which it may not be in Stretch)

echo "Adding ${apInterfaceName} interface..."

iw dev ${wlanInterfaceName} interface add ${apInterfaceName} type __ap


#Modify iptables (these can probably be saved using iptables-persistent if desired)

echo "IPV4 forwarding: setting..."

#sysctl net.ipv4.ip_forward=1

#echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf

sed -i 's/^#net.ipv4.ip_forward=.*$/net.ipv4.ip_forward=1/' /etc/sysctl.conf

echo "Editing IP tables..."

echo 1 > /proc/sys/net/ipv4/ip_forward

iptables -F

iptables -t nat -F

sleep 2

$apSetupIptablesMasquerade

iptables -t nat -A POSTROUTING -o ${wlanInterfaceName} -j MASQUERADE

iptables -A FORWARD -i ${wlanInterfaceName} -o ${apInterfaceName} -m state --state
RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i ${apInterfaceName} -o ${wlanInterfaceName} -j ACCEPT

#iptables-save > /etc/iptables/rules.v4

iptables-save > /etc/iptables.ipv4.nat

#iptables-restore < /etc/iptables.ipv4.nat


# Bring up ${apInterfaceName} interface. Commented out line may be a possible alternative to using
dhcpcd.conf to set up the IP address.

#ifconfig ${apInterfaceName} 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255

ifconfig ${apInterfaceName} up


# Start hostapd. 10-second sleep avoids some race condition, apparently. It may not need to be that
long. (?)

echo "Starting hostapd service..."

systemctl start hostapd.service

sleep 10


#Start dhcpcd. Again, a 5-second sleep
```

```
echo "Starting dhcpcd service..."

systemctl start dhcpcd.service

sleep 20

echo "Starting dnsmasq service..."

systemctl restart dnsmasq.service

#systemctl start dnsmasq.service

echo "Enabling netStop service..."

systemctl enable netStop.service

systemctl start netStop.service

echo "netStart DONE"

bash -c 'echo "\$(date +"%Y-%m-%d %T") - Started: hostapd, dnsmasq, dhcpcd" >> $netLogFile'

EOF

chmod ug+x $netStartFile

doAddRcLocalNetStartSetup

doAddApIpEntriesToHostFile

# Disable regular network services:

# The netStart script handles starting up network services in a certain order and time frame.
Disabling them here makes sure things are not run at system startup.

systemctl unmask hostapd

cat > $netStopFile <<EOF

#! /bin/bash

sudo systemctl stop hostapd

sudo systemctl stop dnsmasq

sudo systemctl stop dhcpcd

sudo systemctl disable hostapd

sudo systemctl disable dnsmasq

sudo systemctl disable dhcpcd

sudo bash -c 'echo "\$(date +"%Y-%m-%d %T") - Stopped: hostapd, dnsmasq, dhcpcd" >> $netLogFile'

# Handle proper shutdown by touching a empty shutdown flag file:

sudo touch $netShutdownFlagFile

EOF

chmod ug+x $netStopFile

# REFERENCE: https://raspberrypi.stackexchange.com/questions/89732/run-a-script-at-shutdown-on-
raspbian

cat > $netStopServiceFile <<EOF

[Unit]

Description=Stops all the WiFi dependencies: hostapd, dnsmasq and dhcpcd.

[Service]

Type=oneshot
```

```
RemainAfterExit=true

ExecStart=/bin/true

ExecStop=$netStopFile

[Install]

WantedBy=multi-user.target

EOF

echo "[Install]: enabling netStop.service ..."

systemctl enable netStop.service

systemctl start netStop.service

chmod ug+x /etc/rc.local

echo "[Install]: DONE"

}

if [ "$cleanup" = true ]; then

doCleanup

echo "[Reboot]: In 10 seconds ..."

sleep 10

reboot

fi

if [ "$install" = true -o "$installUpgrade" = true ]; then

if [ "$apSsidValid" = false -o "$apPassphraseValid" = false \

-o "$apCountryCodeValid" = false -o "$apIpAddrValid" = false ]; then

echo '

Invalid Access Point(AP) setup options are specified for installation.

Please provide the below [OPTION] for installation:

================================================================================

'

if [ "$wlanInterfaceNameValid" = false ]; then

echo '

[WARNING]: Invalid --wifi-interface name provided. Proceeding with default WiFi interface name as:
'$wlanInterfaceNameDefault' ...

'

fi

if [ "$apSsidValid" = false ]; then

echo '

--ap-ssid Mandatory field for installation: Set Access Point(AP) SSID. Atleast 3 chars long.

Allowed special chars are: _ -

'

fi

if [ "$apPassphraseValid" = false ]; then
```

```
echo '

--ap-password Mandatory field for installation: Set Access Point(AP) Password. Atleast 8 chars long.

Allowed special chars are: @ # $ %% ^ & * _ + -

'

fi


if [ "$apCountryCodeValid" = false ]; then

echo '

--ap-country-code Optional field for installation: Set Access Point(AP) Country Code. Default value
is: '$apCountryCodeDefault'.

Make sure that the entered Country Code matches WiFi Country Code if it exists in
/etc/wpa_supplicant/wpa_supplicant.conf

Allowed Country codes are:

'${countryCodeArray[@]:0:30}'

'${countryCodeArray[@]:30:30}'

'${countryCodeArray[@]:60:30}'

'${countryCodeArray[@]:90:30}'

'${countryCodeArray[@]:120:30}'

'${countryCodeArray[@]:150:30}'

'${countryCodeArray[@]:180:30}'

'${countryCodeArray[@]:210:30}'

'${countryCodeArray[@]:240:9}'

'

fi


if [ "$apIpAddrValid" = false ]; then

echo '

--ap-ip-address Optional field for installation: Set Access Point(AP) IP Address. Default value is:
'$apIpDefault'.

LAN/WLAN reserved private Access Point(AP) IP address must in the below range:

[10.0.0.0 - 10.255.255.255] or [172.16.0.0 - 172.31.255.255] or [192.168.0.0 - 192.168.255.255]

(Refer site: https://en.wikipedia.org/wiki/Private_network#Private_IPv4_addresses to know more

about above IP address range).

Access Point(AP) IP address must not be equal to WiFi Station('${wlanInterfaceName}') IP address:
'${wlanIpAddr}'

with its submask: '${wlanIpMask}' and broadcast: '${wlanIpCast}'

'

fi


echo '


--------------------------------------------------------------------------------

Example installation without upgrade:

--------------------------------------------------------------------------------

sudo ./setup-network.sh --install --ap-ssid="abc-1" --ap-password="password@1" --ap-password-encrypt
--ap-country-code="IN" --ap-ip-address="192.168.0.1" --wifi-interface="wlan0"



--------------------------------------------------------------------------------

Example installation with upgrade:

--------------------------------------------------------------------------------
```

```
sudo ./setup-network.sh --install-upgrade --ap-ssid="abc-1" --ap-password="password@1" --ap-password-
encrypt --ap-country-code="IN" --ap-ip-address="192.168.0.1" --wifi-interface="wlan0"

'


exit 0

fi


doInstall

# Sleep for 10s before restarting:

echo "[Reboot]: In 10 seconds ..."

sleep 10

reboot

fi


if [ "$cleanup" = false -a "$install" = false -a "$installUpgrade" = false ]; then

echo '

No Options specified for script execution.

Usage command is sudo setup-network.sh [OPTION].

See [OPTION] below:

================================================================================

--clean Cleans/undo all the previously made network configuration/setup.


--install Install network configuration/setup required to make WiFi chip('${wlanInterfaceName}') as
Access Point(AP) and Station(STA) both.


--install-upgrade Install & Upgrade network configuration/setup required to make WiFi
chip('${wlanInterfaceName}') as Access Point(AP) and Station(STA) both.


--ap-ssid Mandatory field for installation: Set Access Point(AP) SSID. Atleast 3 chars long.

Allowed special chars are: _ -


--ap-password Mandatory field for installation: Set Access Point(AP) Password. Atleast 8 chars long.

Allowed special chars are: @ # $ % ^ & * _ + -


--ap-password-encrypt Optional field for installation. If specified, it will encrypt password in
hostapd.conf file for security reason.


--ap-country-code Optional field for installation: Set Access Point(AP) Country Code. Default value
is: '$apCountryCodeDefault'.

Make sure that the entered Country Code matches WiFi Country Code if it exists in
/etc/wpa_supplicant/wpa_supplicant.conf

Allowed Country codes are:

'${countryCodeArray[@]:0:30}'

'${countryCodeArray[@]:30:30}'

'${countryCodeArray[@]:60:30}'

'${countryCodeArray[@]:90:30}'

'${countryCodeArray[@]:120:30}'

'${countryCodeArray[@]:150:30}'

'${countryCodeArray[@]:180:30}'

'${countryCodeArray[@]:210:30}'

'${countryCodeArray[@]:240:9}'


--ap-ip-address Optional field for installation: Set Access Point(AP) IP Address. Default value is:
```

```
'$apIpDefault'.

LAN/WLAN reserved private Access Point(AP) IP address must in the below range:

[10.0.0.0 - 10.255.255.255] or [172.16.0.0 - 172.31.255.255] or [192.168.0.0 - 192.168.255.255]

(Refer site: https://en.wikipedia.org/wiki/Private_network#Private_IPv4_addresses to know more

about above IP address range).

Access Point(AP) IP address must not be equal to WiFi Station('${wlanInterfaceName}') IP address:
'${wlanIpAddr}'

with its submask: '${wlanIpMask}' and broadcast: '${wlanIpCast}'


--wifi-interface Optional field for installation: Set hardware in-built WiFi interface name to be
used.

Default value is: '$wlanInterfaceNameDefault'.

If an invalid WiFi interface name is provided then the installation will disregard this

WiFi interface name and will not throw any error but, the installation will proceed with

default in-built WiFi interface name as: '$wlanInterfaceNameDefault'.


--------------------------------------------------------------------------------

Example cleanup:

--------------------------------------------------------------------------------

sudo ./setup-network.sh --clean


--------------------------------------------------------------------------------

Example installation without upgrade:

--------------------------------------------------------------------------------

sudo ./setup-network.sh --install --ap-ssid="abc-1" --ap-password="password@1" --ap-password-encrypt
--ap-country-code="IN" --ap-ip-address="192.168.0.1" --wifi-interface="wlan0"


--------------------------------------------------------------------------------

Example installation with upgrade:

--------------------------------------------------------------------------------

sudo ./setup-network.sh --install-upgrade --ap-ssid="abc-1" --ap-password="password@1" --ap-password-
encrypt --ap-country-code="IN" --ap-ip-address="192.168.0.1" --wifi-interface="wlan0"

'

exit 0

fi
```

Readme NetSetup

El uso es el mostrado a continuación:

"sudo ./setup-network.sh [OPTION]"

Las opciones diponibles son:

--clean

Limpia todo el trabajo previo de configuración

--install

Realiza la configuración para establecer el chip WIFI como Access Point (AP) y como estación (STA) al mismo tiempo.

--install-upgrade

Instala y actualiza la configuración en modo AP y STA.

--ap-ssid

Identificador del Access Point. Al menos 3 caracteres.

--ap-password

Password requerida, al menos 8 caracteres.

--ap-password-encrypt

Encripta la contraseña del access point en los ficheros de configuración.

--ap-country-code

Código de País para configuración del Access Point. "/etc/wpa_supplicant/wpa_supplicant.conf" file.

--ap-ip-address

La IP del punto de acceso. Por defecto: 10.0.0.1.

--wifi-interface

Interfaz wifi a usar.

Ejemplos de Uso

Ejemplo de cleanup:

```
sudo ./setup-network.sh --clean
```

Ejemplo de instalación sin actualización:

```
sudo ./setup-network.sh --install --ap-ssid="abc-1" --ap-
password="password@1" --ap-password-encrypt
--ap-country-code="IN" --ap-ip-address="192.168.0.1" --wifi-
interface="wlan0"
```

Ejemplo de instalación con actualización de configuración anterior:

```
sudo ./setup-network.sh --install-upgrade --ap-ssid="abc-1" --ap-
password="password@1" --ap-password-encrypt
--ap-country-code="IN" --ap-ip-address="192.168.0.1" --wifi-
interface="wlan0"
```