



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE MÁSTER
MÁSTER PROPIO EN CIBERSEGURIDAD

IDSpi - Desarrollo de un sistema de detección de intrusos portable en una Raspberry Pi

Autor

Antonio Miguel Ramírez Oliva

Director

Javier Tallón Guerri



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, Septiembre de 2018

IDSpi - Desarrollo de un sistema de detección de intrusos portable en una Raspberry Pi

Antonio Miguel Ramírez Oliva

Palabras clave: IDS, Bro, CriticalStack, Loki, Raspberry Pi, Elasticsearch, Logstash, Kibana.

Resumen

El uso extendido por parte de la sociedad de las tecnologías de la información y la comunicación junto con las mejoras continuas en las soluciones de conectividad a internet lo que hacen que no solo los ordenadores estén conectados sino también de teléfonos móviles, dispositivos IoT ... hace que la superficie frente a posibles ataques esté creciendo exponencialmente.

La transformación digital está llegando a todos los sectores y con el auge del hacktivista y el incremento de la ciberguerra hace que cualquier dispositivo conectado a internet sea de por sí potencialmente atacable. Además, la falta de criterio por parte de los fabricantes en las configuraciones por defecto de los dispositivos dejan la puerta abierta ante ataques y amenazas.

Teniendo todo esto en consideración toma relevancia la posibilidad de poder disponer de un sistema portátil que a bajo coste pueda hacer las labores de un sistema de detección de intrusiones. Este escenario puede ser especialmente relevante en entornos de pequeña y mediana empresa puesto que por muy automático que se pueda diseñar el sistema siempre será necesaria una parte de configuración en los elementos de la red para hacerle llegar una copia del tráfico entrante y saliente que se está consumiendo.

Tras un análisis de las distintas tecnologías existentes se opta por elegir Bro como motor IDS integrado junto con CriticalStack para recolección de feeds de amenazas y poder dotar de una mayor inteligencia al sistema. Todos los logs generados por Bro, junto con los logs generados por el propio sistema y los componentes necesarios para la interfaz web se envían a un servidor ELK que será el responsable de su análisis y sobre el que se podrán explotar diferentes visualizaciones y dashboards que nos permitirán determinar cuando hay un intruso en nuestra red que esté generando tráfico malicioso.

En este trabajo fin de master se ha conseguido integrar Bro y CriticalStack como motores principales de seguridad de nuestro sistema y a través de diferentes componentes adicionales de Bro se consigue por ejemplo tener una réplica de los ficheros detectados en los flujos de tráfico para su posterior análisis con Loki o guardar capturas raw del tráfico de red para que en caso de detectar alguna anomalía poder realizar un análisis más detallado permitiendo acceder a las pymes de manera sencilla a una tecnología que les

permite incrementar sus capacidades de defensa ante ciber ataques a precios muy reducidos.

IDSpi - Development of a portable intrusion detection system over a Raspberry Pi

Antonio Miguel Ramírez Oliva

Keywords: IDS, Bro, CriticalStack, Loki, Raspberry Pi, Elasticsearch, Logstash, Kibana.

Abstract

The widespread use by the society of information and communication technologies, together with continuous improvements of internet connectivity solutions, means that not only computers are connected but also mobile phones, IoT devices... this makes the surface of possible attacks to be growing exponentially. The digital transformation is reaching all sectors and with the rise of hacktivists and the increase in cyberwar, any device connected to the Internet is potentially attackable. In addition, the lack of criteria of manufacturers in the devices default configurations leaves the door open to attacks and threats. Taking all this into consideration, the possibility of having a portable system that can do the work of an intrusion detection system at a low cost seems important. This scenario may be especially relevant in small and medium-sized businesses since no matter how automatically the system can be designed, a part of the configuration of the network elements will always be necessary to provide you with a copy of the incoming and outgoing traffic that is being used.

After an analysis of the different existing technologies, Bro was chosen as the integrated IDS engine together with CriticalStack for the collection of threat feeds and to provide the system with greater intelligence.

All the logs generated by Bro, together with the logs generated by the system itself are sent to an ELK server that will be responsible for their analysis and on which different visualizations and dashboards can be used to determine when there is an intruder on our network that is generating malicious traffic.

In this project, Bro and CriticalStack have been integrated as the main security engines of our system and through different additional components of Bro, for example, it is possible to have a replica of the files detected in the traffic flows for later analysis with Loki or to save raw captures of the network traffic so that if an anomaly is detected, a further analysis can be carried out, allowing SMEs easy access to a technology that allows them to increase their defense capabilities against cyber attacks at very low prices.

Yo, **Antonio Miguel Ramírez Oliva**, alumno del **Máster Propio en Ciberseguridad** de la **Universidad de Granada**, con DNI 14629575-B, autorizo la ubicación de la siguiente copia de mi *Trabajo Fin de Máster* en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Antonio Miguel Ramírez Oliva

Granada a 7 de Septiembre de 2018.

D. Javier Tallón Guerri, Profesor del Máster en Ciberseguridad de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***IDSpi - Desarrollo de un sistema de detección de intrusos portable en una Raspberry Pi***, ha sido realizado bajo su supervisión por **Antonio Miguel Ramírez Oliva**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 7 de Septiembre de 2018.

El director:

Javier Tallón Guerri

Índice general

1. Motivación y Objetivos	1
1.1. Introducción	1
1.2. Ciberataques en España	2
1.3. Ciberseguridad en entornos domésticos y/o PyME	3
1.4. Taxonomía de las amenazas	5
1.5. ¿Cómo protegerme?	5
1.6. Motivación	5
1.6.1. ¿Qué es un IDS?	5
1.6.2. ¿Por qué usar un IDS?	9
1.6.3. Clasificación de los IDS	11
1.6.4. Requisitos de un IDS	12
1.6.5. Estrategias de respuesta	13
1.7. Objetivos	14
2. Análisis y Planificación	15
2.1. Introducción	15
2.2. Análisis	15
2.2.1. Software IDS	15
2.2.2. Framework de desarrollo	20
2.2.3. Plataforma hardware	23
2.2.4. Sistema Operativo	25
2.2.5. Monitorización y explotación de logs	27
2.3. Planificación	28
2.3.1. Recursos utilizados	29
2.3.2. Estimación de costes	30
3. Diseño	33
3.1. Diseño de la conectividad física para la recogida de datos . .	33
3.2. Diseño de la arquitectura de red y de gestión	34
3.3. Diseño de las herramientas y módulos de seguridad	34
3.3.1. Captura del tráfico de red	35
3.3.2. Análisis del tráfico con Bro e integración con CriticalStack	36

3.3.3. Análisis de malware sobre los ficheros extraídos	36
3.3.4. Estructura de directorios	37
3.3.5. Arquitectura para la interfaz gráfica	38
3.3.6. Diseño de la interfaz gráfica	38
3.3.7. Recogida de logs, envío y tratamiento	44
4. Implementación	47
4.1. Introducción	47
4.2. Instalación de Raspberry Pi	47
4.2.1. Instalación del sistema operativo	47
4.2.2. Configuración de la electrónica de red	48
4.2.3. Configuración inicial del sistema operativo	51
4.2.4. Instalación de las herramientas de seguridad	53
4.2.5. Compilación y configuración de Filebeat y Metricbeat	67
4.2.6. Instalación de la interfaz gráfica web	77
4.3. Instalación del servidor Logstash, Elasticsearch y Kibana . .	89
4.3.1. Instalación y configuración de Elasticsearch y Kibana	91
4.3.2. Instalación y configuración de Logstash	92
4.4. Gestión y análisis de logs	94
4.4.1. Creación de índices	95
4.4.2. Visualización y explotación de logs del sistema . . .	95
4.4.3. Visualización y explotación de logs del IDS	102
5. Conclusiones y Trabajo Futuro	123
5.1. Conclusiones	123
5.2. Líneas de trabajo futuro	124
5.3. Valoración personal	125
Bibliografía	129

Índice de figuras

1.1. Mapa del mundo ciberataques en tiempo real	3
1.2. Taxonomía de ataques parte 1	6
1.3. Taxonomía de ataques parte 2	7
1.4. Fases de un ataque	10
2.1. Arquitectura de Snort	17
2.2. Arquitectura de Suricata	19
2.3. Raspberry Pi	24
2.4. Arquitectura Raspberry Pi 3 Model B	25
2.5. Flujo de ELK	27
2.6. Diagrama de Gantt planificación del proyecto	28
3.1. Diseño de la conectividad física para la recogida de datos . .	34
3.2. Diseño de la arquitectura de red y de gestión	35
3.3. Captura del tráfico de red	35
3.4. Análisis del tráfico con Bro e integración CriticalStack . . .	37
3.5. Arquitectura LAMP para la interfaz web	38
3.6. Wireframe de la página de login	40
3.7. Wireframe de la página Home	41
3.8. Wireframe de la página Dashboard	42
3.9. Wireframe de la página Bro Config	43
3.10. Wireframe de la página Logins	45
3.11. Wireframe de la página Log Tool	46
3.12. Arquitectura para la recogida de logs, envío y tratamiento .	46
4.1. SD Memory Card Formatter	48
4.2. Win 32 Disk Imager	48
4.3. Pantalla de inicio de Raspbian	49
4.4. Menú de inicio de configuración del switch HP	49
4.5. Menú de configuración de VLAN en el switch HP	50
4.6. Menú de asignación de puertos por VLAN en el switch HP .	50
4.7. Menú de configuración de port mirroring en el switch HP .	51
4.8. Pantalla de configuración de Raspbian	52
4.9. Pantalla de configuración de ZeroTier	53

4.10. Visualización de la interfaz virtual creada con ZeroTier	54
4.11. Creación de un sensor en Critical Stack	63
4.12. Suscripción a feeds en Critical Stack	64
4.13. Página de login de la interfaz web del sistema	83
4.14. Menú Home de la interfaz web	84
4.15. Menú Dashboard de la interfaz web	85
4.16. Detalle de los procesos en ejecución del menú Dashboard de la interfaz web	86
4.17. Detalle de las conexiones de red del menú Dashboard de la interfaz web	87
4.18. Menú Bro Config de la interfaz web	88
4.19. Menú Logins de la interfaz web	89
4.20. Menú Log Tool de la interfaz web	90
4.21. Creación de un mapping para tipos de dato Geopoint	95
4.22. Creación de índices en Elasticsearch para logs Bro	96
4.23. Selección de un índice por defecto en Elasticsearch	96
4.24. Elección de los diferentes índices en Kibana	97
4.25. Dashboard de Metricbeat con el estado del sistema (1)	98
4.26. Dashboard de Metricbeat con el estado del sistema (2)	98
4.27. Dashboard de Metricbeat con el estado del sistema (3)	99
4.28. Dashboard del resumen de accesos al servidor Apache2	99
4.29. Dashboard del log de errores del servidor Apache2	100
4.30. Dashboard de syslog	101
4.31. Dashboard de accesos SSH al sistema	101
4.32. Dashboard de comandos lanzados con SUDO en el sistema	102
4.33. Base de datos de firmas de Critical Stack	103
4.34. Ejemplo de log INTEL recibido en Kibana	103
4.35. Geo posicionamiento por coordenadas de IPs en el log de Critical Stack	105
4.36. Geo posicionamiento por países de IPs en el log de Critical Stack	106
4.37. Geo posicionamiento por países de IPs en el log de conexiones	108
4.38. Búsqueda de IP por país de origen en el log de conexiones	109
4.39. Resolución inversa de nombres a partir de dirección IP	109
4.40. Número de archivos identificados agrupados por tipo	112
4.41. Principales certificados detectados agrupados por estado de validación	114
4.42. Tabla con los cifrados utilizados, recurso accedido y direcciones IP	114
4.43. Geo posicionamiento por países del tráfico HTTP	116
4.44. Visualización Heat Zones para el log DNS	119
4.45. Visualización de mayores consultas DNS	119
4.46. Línea base de conexiones activas	121

Índice de cuadros

1.1. Decálogo Básico de Seguridad	8
2.1. Especificaciones Raspberry 3 Model B	26
2.2. Características Raspbian	26
2.3. Estimación de costes relativos a recursos humanos	31
2.4. Estimación de costes relativos a recursos hardware	31
2.5. Estimación de costes totales	31
3.1. Rutas definidas para la navegación web, controladores y funciones que ejecutan	39
3.2. Rutas definidas para la API Dashboard, controladores y funciones que ejecutan	42
3.3. Rutas definidas para la API Bro Config, controladores y funciones que ejecutan	43
3.4. Métodos llamados desde la función getFullSystem	44

Índice de scripts

3.1. Estructura de directorios del sistema	37
4.1. Script instalación ZeroTier	52
4.2. Script instalación GeoIP	53
4.3. Script creación de carpetas e instalación de .debs	55
4.4. Deshabilitar IPv6	56
4.5. Configuración de la interfaz en modo promiscuo	57
4.6. Auxiliar nic.sh	57
4.7. Deshabilita la configuración por DHCP a través de eth0	57
4.8. Instalar y configurar servicio Netsniff	58
4.9. Borrado de .pcaps	59
4.10. Sincronización horaria NTP	60
4.11. Instalación de Bro	61
4.12. Instalación de Loki	62
4.13. Instalación de Critical Stack	63
4.14. Ajuste de Critical Stack para funcionar en Raspberry Pi	65
4.15. Scripts auxiliares de Bro	65
4.16. Tareas recurrentes y configuración de cron	67
4.17. Arranque del docker golang	68
4.18. Compilación de Filebeat	68
4.19. Configuración del módulo system de Filebeat	70
4.20. Configuración del módulo apache2 de Filebeat	70
4.21. Configuración de Filebeat (Sistema y Apache2)	70
4.22. Configuración de Filebeat (Bro)	71
4.23. Configuración de Metricbeat	74
4.24. Configuración del módulo system de Metricbeat	75
4.25. Instalación en Kibana de visualizaciones y dashboards	75
4.26. Configuración de los servicios asociados a Filebeat y Metricbeat	76
4.27. Instalación de los componentes necesarios para la interfaz web	77
4.28. Creación del usuario phpmyadmin para la base de datos	78
4.29. Modificación del fichero de configuración de phpmyadmin	78
4.30. Fichero 000-default.conf	78
4.31. Fichero default-ssl.conf	79
4.32. Creación de la base de datos y tablas necesarias	80
4.33. Instalación de dependencias para ELK	91

4.34. Instalación de clave GPG	91
4.35. Configuración Kibana	91
4.36. Configuración Kibana	91
4.37. Configuración y arranque de servicios elasticsearch y Kibana	92
4.38. Configuración de Nginx	92
4.39. Configuración del enlace simbólico para Nginx	92
4.40. Configuración de Logstash	93
4.41. Configuración del servicio de Logstash	94
4.42. Ejemplo de log Intel en formato json	104
4.43. Ejemplo de log conn en formato json	106
4.44. Ejemplo de log files en formato json	110
4.45. Ejemplo de log ssl en formato json	111
4.46. Ejemplo de log http en formato json	115
4.47. Ejemplo de log dns en formato json	116
4.48. Ejemplo de log stats en formato json	118

Capítulo 1

Motivación y Objetivos

1.1. Introducción

Hoy en día, el desarrollo de la tecnología en general e Internet en particular han conseguido en muy poco tiempo formar parte fundamental de la vida de la mayoría de sus usuarios provocando un cambio en las vidas de los usuarios especialmente en términos de facilidad de comunicación instantánea y la facilidad del acceso a la información, [1].

Internet se ha implantado en nuestra sociedad modificando comportamientos inherentemente humanos como son las relaciones sociales en las que han abierto un nuevo mundo de posibilidades. También ha propiciado cambios en la forma de comprar gracias al comercio electrónico, cambios en la educación facilitando el acceso a la información y como una fuente de conocimiento infinita.

[2] En esta época de cambios vertiginosos y desarrollo constante de la tecnología, es imprescindible cuidar el valor de la información que publicamos en internet o que almacenamos en nuestros dispositivos. Un uso inseguro presenta un riesgo pues puede poner en peligro nuestra información más privada y confidencial, como por ejemplo la numeración de nuestras cuentas bancarias, tarjetas de crédito, la intimidad de nuestros hijos o nuestra propia identidad.

Según se desprende de la encuesta sobre el equipamiento y uso de tecnologías de información y comunicación en los hogares, TIC-H 2017, realizada por el Instituto nacional de estadística (INE), en España, el 83,4% de los hogares españoles tiene actualmente acceso a Internet. El estudio indica que el 78,4% de los hogares con al menos un miembro de 16 a 74 años tiene ordenador; el 52,4% de los hogares tiene *tablet*, el 99,5% dispone de teléfono (fijo o móvil) y el 75,5% tiene ambos tipos de terminales.

1.2. Ciberataques en España

La generalización del uso de los medios electrónicos en el normal desarrollo de la sociedad ha incrementado la superficie de exposición a ataques y, en consecuencia, los beneficios potenciales derivados, lo que constituye sin duda uno de los mayores estímulos para los atacantes. En los últimos años se ha mantenido la tendencia, incrementándose el número, tipología y gravedad de los ataques contra los sistemas de información del Sector público, de las empresas e instituciones de interés estratégico o de aquellas poseedoras de importantes activos de propiedad intelectual e industrial y, en general, contra todo tipo de entidades y ciudadanos.

La deslocalización y ubicuidad que proporciona Internet hace que los conceptos de localización y entorno se diluyan. A través de Internet, todo el mundo está interconectado haciendo que para recibir un ciberataque no haga falta otra cosa más que estar conectado a Internet. [3] En 2016, el Instituto Nacional de Ciberseguridad (INCIBE) contabilizó 115.000 incidentes informáticos en España, un 130 % más que los registrados el año anterior. Esta estadística sitúa a España como tercer país más atacado del mundo, ver figura 1.1. Pero, ¿quién nos ataca? Esta pregunta tiene difícil respuesta aunque se estima que los principales orígenes son China y Rusia.

La complejidad de la ciberguerra nace en la niebla que inunda el mundo virtual, que ciega tanto a estados, como a empresas y usuarios domésticos. Ni siquiera los propios países actúan con criterios tradicionales, opera la ley de la jungla y el culto al camuflaje. Más allá de los conflictos internacionales que llevan la geopolítica al ordenador surge una cantidad de actores que complican todavía más las cosas para los encargados de la ciberdefensa. El Centro Criptológico Nacional (CCN), en su informe Ciberamenazas 2015/Tendencias 2016, elaboró una lista de los principales ciberenemigos a los que se enfrenta España. Estos agentes son los citados Estados, ciberdelincuentes, hackivistas, grupos yihadistas, terroristas, cibervándalos, actores internos (motivados por venganza o búsqueda de beneficios económicos o ideológicos), ciberinvestigadores (revelación de debilidades de un usuario que pueden usarse contra él) y organizaciones privadas (con intereses en obtener o vender información).

La transformación digital impulsada por Internet, como se ha visto, es totalmente generalizada y ha llegado a cualquier sector y a todos los ámbitos, empresariales, domésticos, ... haciendo foco especial en este último, sobre todo con el internet de las cosas. El gran reto es ser capaces de visualizarlo y protegerlo. Ese es el foco de este trabajo fin de máster: la visualización, el análisis y la seguridad en el entorno doméstico o entorno PYME.



Figura 1.1: Mapa del mundo ciberataques en tiempo real

1.3. Ciberseguridad en entornos domésticos y/o PyME

Dentro del ámbito doméstico o PyME en conjunción con el incremento de la tecnología IoT hace que se presenten una serie de riesgos y vulnerabilidades a tener en cuenta como base motivacional de este proyecto. Según [4] estos riesgos son:

- **Recursos limitados:** en un entorno doméstico y la mayoría de los dispositivos IoT las capacidades económicas, las habilidades técnicas y formativas de los usuarios y las capacidades de procesamiento, memoria y potencia son muy limitados por lo que los controles de seguridad avanzados no pueden aplicarse eficazmente.
- **Ecosistema complejo:** las preocupaciones de seguridad se agravan ya que el IoT no debe verse como una colección de dispositivos independientes, sino como un ecosistema rico, diverso y amplio que involucra aspectos como dispositivos, comunicaciones, interfaces y personas.
- **Bajo costo:** en algunos casos los fabricantes podrían estar inclinados a limitar las características de seguridad para asegurar un bajo costo y por lo tanto, la seguridad del producto podría no ser capaz de proteger contra ciertos tipos de ataques.
- **Falta de experiencia:** la ciberseguridad es un ámbito bastante novedoso por lo que hay una falta de conocimientos básicos por los usuarios. Tampoco se cuenta con un background previo de histórico de amenazas o problemas que permitan disponer de unas lecciones aprendidas aplicables a la tecnología.

- **Fallos de seguridad en el diseño propios del dispositivo y de su explotación:** la práctica más habitual es que los fabricantes se centren en minimizar el tiempo de lanzamiento de los productos des-
cuidando a veces en la fase del diseño aspectos esenciales de ciberseguridad (cifrado de la información transmitida, controles de acceso, etc) en muchos casos debido a la necesidad de anticiparse a la competencia en el lanzamiento.
- **Falta de control y asimetría de la información:** en muchas oca-
siones el usuario no es consciente del tratamiento de datos llevado a
cabو por los dispositivos conectados en su red. Los mecanismos con-
vencionales utilizados para obtener el consentimiento de los usuarios
son considerados consentimientos “de baja calidad” debido a que en
muchos casos se basan en la falta de información que recibe el usuario
sobre el posterior tratamiento de los datos personales que está propor-
cionando. Además esta información puede llegar a manos de terceros
sin que el usuario sea consciente de la difusión de la misma. También la
falta de control que existe en tecnologías como los servicios en la nube
y el Big Data, incluso en la problemática que surge de la combinación
de ambos, hace que la falta de control y la asimetría de la información
estén muy presentes.
- **Limitaciones en la posibilidad de permanecer en el anonima-
to cuando se utilizan servicios:** el avance del uso de servicios en
la nube provocará la pérdida del anonimato en los que a día de hoy se
presupone como algo garantizado. Para proteger dicho anonimato será
necesario mejorar las técnicas de control de acceso y de cifrado, desa-
rrollar técnicas de apoyo al concepto de Privacidad por Diseño, evitar
la inferencia de información y preservar la privacidad de la ubicación
del usuario.
- **Seguridad frente a eficiencia:** a la hora de equilibrar la optimi-
zación de los recursos hardware del dispositivo con los requisitos de
seguridad que exigen estos dispositivos, se plantean varios desafíos pa-
ra los fabricantes. Dado que la presión de tiempo de comercialización,
especialmente en los productos IoT, es mayor que en otros ámbitos, a
veces se imponen limitaciones a los esfuerzos para desarrollar dispositi-
vos seguros. Por esta razón, y a veces también debido a problemas de
presupuesto, las empresas que desarrollan productos IoT ponen más
énfasis en la funcionalidad y usabilidad que en la seguridad.
- **Responsabilidades poco claras:** la falta de una asignación clara de
responsabilidades (fabricante/prestador del servicio/usuario) podría
dar lugar a ambigüedades y conflictos en caso de ocurrir un suceso que
afecte a la seguridad.

1.4. Taxonomía de las amenazas

Para poder comprender ante qué nos estamos enfrentando es necesario hacer una clasificación clara de las diferentes tipologías de amenazas. A continuación se muestra una posible clasificación en las figuras 1.2 y 1.3 elaborada por CERTSI, CERT de Seguridad e Industria.

1.5. ¿Cómo protegerme?

Existen una serie de medidas básicas de protección al alcance de cualquier usuario, tanto en el ámbito empresarial, PyME o doméstico, en el que encontrar unas recomendaciones básicas que harán que podamos estar más seguros en el mundo virtual en el que como se ha expuesto anteriormente estamos inmersos. El CCN-CERT en su informe de Buenas Prácticas CCN-CERT BP-01/16. Principios y recomendaciones básicas de ciberseguridad [5] se detallan una serie de medidas fundamentales a la hora de mantener la seguridad en las Tecnologías de la Información y Comunicación (TIC). El CCN-CERT considera que "la concienciación, el sentido común y las buenas prácticas son las mejores defensas para prevenir y detectar contratiempos en la utilización de sistemas TIC" y, aunque no existe un sistema que garantice al 100% la seguridad, es preciso combinar diferentes prácticas para proporcionar un nivel de protección mínima. En la tabla 1.1 se muestra el decálogo básico de seguridad. Un decálogo de buenas prácticas que pretende sentar las bases para establecer una cultura de seguridad.

1.6. Motivación

Una vez analizado el contexto, dada una visión general del escenario y la ciberseguridad en España, los principales riesgos a los que se expone cualquier equipo conectado a internet y una introducción a las buenas prácticas que cualquier usuario debe de tener en cuenta surge la siguiente pregunta, ¿cómo detecto si me están atacando? Es aquí donde surge la idea de un **IDS**.

1.6.1. ¿Qué es un IDS?

[6] Un IDS es un Sistema de Detección de Intrusos (Intrusion Detection System). Se trata de una herramienta de seguridad encargada de monitorizar los eventos que ocurren en un sistema en busca de intentos de intrusión.

Se define un intento de intrusión como cualquier intento de comprometer la confidencialidad, integridad y/o disponibilidad o evitar los mecanismos de

CATEGORÍA	AMENAZA	DESCRIPCIÓN
Ataques /Abusos	Malware	Programas informáticos diseñados para realizar acciones no deseadas y no autorizadas en un sistema sin el consentimiento del usuario. Ocasionan daños, lesiones o robo de información. Poseen un impacto alto.
	Secuencias de <i>Exploit</i>	Código diseñado para aprovechar un punto vulnerable y acceder a un sistema. La detección de esta amenaza es difícil y en los entornos IoT puede tener distintos grados de impacto, desde un impacto elevado hasta un impacto grave, dependiendo de los activos que se vean afectados.
	Ataques dirigidos	Ataques diseñados con un objetivo específico lanzados durante un periodo de tiempo prolongado y llevados a cabo en numerosas fases. Su objetivo principal es permanecer ocultos para obtener la mayor cantidad de información/datos confidenciales o el mayor nivel de control posible. Aunque esta amenaza presenta un impacto medio, su detección suele ser complicada y requiere mucho tiempo.
	Denegación de Servicio Distribuido (DDoS)	Varios sistemas atacan un único objetivo para saturarlo y dejarlo inoperativo. Puede llevarse a cabo empleando varias conexiones, colapsando un canal de comunicación o reenviando las mismas comunicaciones de manera repetida.
	Falsificación de dispositivos maliciosos	Esta amenaza es difícil de detectar, puesto resulta complicado diferenciar un dispositivo falso de uno original. Estos dispositivos cuentan normalmente con <i>backdoors</i> o puertas traseras que pueden emplearse para atacar otros sistemas de ICT (Tecnologías de la Información y de las Comunicaciones, por sus siglas en inglés) en el entorno.
	Ataques a la privacidad	Estas amenazas afectan tanto a la privacidad del usuario como a la exposición de los elementos en red a personal no autorizado.
Eavesdropping / Intercepción / Secuestro	Modificación de información	En este caso, el objetivo no reside en dañar los dispositivos, sino en manipular la información para causar caos u obtener beneficios económicos.
	<i>Man in the middle</i>	Se trata de un ataque de intercepción activa en el que el atacante se posiciona en medio de la comunicación entre dos víctimas, haciéndoles creer que están hablando directamente entre sí.
	Secuestro de protocolo de comunicación IoT	Toma del control de una sesión de comunicación existente entre dos elementos de la red. El intruso tiene acceso a información, incluyendo contraseñas. El secuestro puede emplear técnicas agresivas, como forzar la desconexión o la denegación de servicio.
	Intercepción de información	Intercepción no autorizada (y, en ocasiones, modificación) de procesos de comunicación privada, como llamadas telefónicas, mensajería instantánea o correos electrónicos.
	Reconocimiento de red	Obtención pasiva de información sobre la red: dispositivos conectados, protocolo empleado, puertos abiertos, servicios en uso, etc.
	Secuestro de sesión	Robo de la conexión de datos actuando como <i>host</i> legítimo con el objetivo de robar, modificar o eliminar los datos transmitidos.
	Obtención de información	Obtención pasiva de información sobre la red: dispositivos conectados, protocolo empleado, etc.
	Reproducción de mensajes	Este ataque emplea una transmisión de datos válida de manera maliciosa enviándolos repetidamente o retrasándolos, con el propósito de dejar inoperativo el dispositivo objetivo.

Figura 1.2: Taxonomía de ataques parte 1

Caidas	Caída de red	Interrupción o fallo en el suministro de red, bien sea de manera intencionada o accidental. Dependiendo del segmento de la red afectado y del tiempo necesario para recuperar el servicio, la importancia de esta amenaza varía de alta a grave.
	Fallos de dispositivos	Amenaza de fallo o avería en los dispositivos de <i>hardware</i> .
	Fallo del sistema	Amenaza de fallo de los servicios o aplicaciones de <i>software</i> .
	Pérdida de servicios de soporte	Pérdida del acceso a servicios de soporte necesarios para el funcionamiento adecuado del sistema de información.
Daño / Pérdida (Activos TI)	Filtrado de datos / información confidencial	Se revelan datos confidenciales, de manera intencional o no, a terceros no autorizados. La importancia de esta amenaza puede variar, dependiendo del tipo de datos que se filtren.
Fallos / averías	Vulnerabilidades del <i>software</i>	Con frecuencia, los dispositivos IoT más comunes son vulnerables a causa de contraseñas débiles/por defecto, errores de <i>software</i> y errores de configuración, lo que supone un riesgo para la red. Esta amenaza suele estar vinculada con otras, como secuencias de <i>exploit</i> , y se considera una amenaza grave.
	Fallos de terceros	Errores en un elemento activo de la red ocasionados por una configuración no adecuada de un elemento que guarda relación directa con este.
Desastre	Desastre natural	Incluye desastres naturales tales como inundaciones, fuertes rachas de viento, grandes nevadas o desprendimientos de tierra, que pudiesen ocasionar un daño físico en los dispositivos.
	Desastres ambientales	Desastres en el despliegue de los entornos de equipos de IoT, causando su inoperatividad.
Ataques físicos	Modificación de dispositivos	Manipulación de dispositivos, por ejemplo, mediante malas comunicaciones de puertos, aprovechando aquellos que quedan abiertos.
	Destrucción del dispositivo (sabotaje)	Sucesos como robo del dispositivo, ataques con explosivos, vandalismo o sabotaje, que puedan dañar el dispositivo.

Figura 1.3: Taxonomía de ataques parte 2

Decálogo básico de seguridad	
1	La cultura de la ciberseguridad, la concienciación del usuario, debe ser uno de los pilares en lo que se asiente la ciberseguridad en cualquier ámbito.
2	No abrir ningún enlace ni descargar ningún fichero adjunto procedente de un correo electrónico que presente cualquier indicio o patrón fuera de lo habitual.
3	Utilizar software de seguridad, herramientas antivirus y antimalware, cortafuegos personales, herramientas de borrado seguro, etc. debe ser algo irrenunciable cuando se utiliza un sistema de las TIC.
4	Limitar la superficie de exposición a las amenazas, no solo hay que implementar medidas de seguridad que protejan el acceso a la información, sino que hay que determinar los servicios que son estrictamente necesarios.
5	Cifrar la información sensible, no hay otra alternativa.
6	Utilizar contraseñas adaptadas a la funcionalidad siendo conscientes de que la doble autenticación ya es una necesidad.
7	Hacer un borrado seguro de la información una vez que esta ya no sea necesaria o se vaya a retirar de uso el soporte en cuestión.
8	Realizar copias de seguridad periódicas, no existe otra alternativa en caso de infección de código malicioso tipo ransomware, pérdida de datos, averías del hardware de almacenamiento, borrado de información involuntaria por parte del usuario, etc.
9	Mantener actualizadas las aplicaciones y el sistema operativo es la mejor manera de evitar dar facilidades a la potencial amenaza.
10	Revisa regularmente la configuración de seguridad aplicada, los permisos de las aplicaciones y las opciones de seguridad.

Cuadro 1.1: Decálogo Básico de Seguridad

seguridad de un sistema por parte de una persona o usuario no autorizado.

1.6.2. ¿Por qué usar un IDS?

La detección de intrusiones permite proteger los sistemas de las amenazas y prevenir la ocurrencia de un incidente de seguridad. Entre las ventajas de los IDSs cabe destacar:

Prevenir problemas al disuadir a individuos hostiles

Al incrementar la posibilidad de descubrir y castigar a los atacantes, el comportamiento de algunos cambiará de forma que muchos ataques no llegarán a producirse.

Detectar ataques y otras violaciones de la seguridad que no son prevenidas por otras medidas de protección

Los atacantes pueden conseguir accesos no autorizados a los sistemas a través de las conexiones públicas, Internet. Esto a menudo ocurre cuando vulnerabilidades conocidas no son corregidas. Los fabricantes y vendedores procuran dar a conocer y corregir estas vulnerabilidades pero, especialmente en un entorno con poco recursos y conocimientos técnicos, hay situaciones en las que esto no es suficiente.

- Desconocimiento del usuario.
- Sistemas obsoletos o sin mantenimiento.
- Mantenimiento no adecuado.

Un IDS puede detectar cuando un atacante ha intentado penetrar en un sistema explotando una vulnerabilidad no corregida. De esta forma, es posible avisar al usuario para que de manera guiada proceda a corregir o mitigar dicha vulnerabilidad.

Detectar preámbulos de ataques (normalmente pruebas de red y otras actividades)

Los ciberataques normalmente presentan una serie de fases o patrones reproducibles [7]. Normalmente un ataque se produce en cinco fases las que pueden verse en la figura 1.4.



Figura 1.4: Fases de un ataque

- **Reconocimiento:** Esta etapa involucra la obtención de información (Information Gathering) con respecto a una potencial víctima que puede ser una persona u organización. Por lo general, durante esta fase se recurre a diferentes recursos de Internet como Google, entre tantos otros, para recolectar datos del objetivo. Algunas de las técnicas utilizadas en este primer paso son la Ingeniería Social, el Dumpster Diving, el sniffing.
- **Exploración:** En esta segunda etapa se utiliza la información obtenida en la fase 1 para sondar el blanco y tratar de obtener información sobre el sistema víctima como direcciones IP, nombres de host, datos de autenticación, entre otros. Entre las herramientas que un atacante puede emplear durante la exploración se encuentran el network mappers, port mappers, network scanners, port scanners, y vulnerability scanners.
- **Obtener acceso:** En esta instancia comienza a materializarse el ataque a través de la explotación de las vulnerabilidades y defectos del sistema (Flaw exploitation) descubiertos durante las fases de reconocimiento y exploración. Algunas de las técnicas que el atacante puede utilizar son ataques de Buffer Overflow, DDoS, Password filtering y Session hijacking.
- **Mantener acceso:** Una vez que el atacante ha conseguido acceder al sistema, buscará implantar herramientas que le permitan volver a

acceder en el futuro desde cualquier lugar donde tenga acceso a internet. Para ellos, suelen recurrir a utilidades como backdoors, rootkits y troyanos.

- **Borrar huellas:** Una vez que el atacante ogro obtener y mantener el acceso al sistema, intentara borrar todas las huellas que fue dejando durante la intrusión para evitar ser detectado por el profesional de seguridad o los administradores de la red. En consecuencia, buscara eliminar los archivos de registro (log) o alarmas del Sistema de Detección de Intrusos (IDS).

Un IDS permitirá alertar sobre actividad sospechosa durante la fase de exploración e incluso podrá bloquear el acceso activamente a la red por parte del atacante antes incluso de que se produzca el ataque.

Proveer información útil sobre las intrusiones que se produzcan

Incluso cuando el IDS no sea capaz de reconocer un ataque puede recoger información valiosa durante la ejecución del mismo que podrá ser utilizada como prueba en actuaciones legales o usarse para corregir fallos en la configuración de los equipos.

1.6.3. Clasificación de los IDS

[8] Generalmente existen dos grandes enfoques a la hora de clasificar a los sistemas de detección de intrusos: o bien en función de qué sistemas vigilan, o bien en función de cómo lo hacen.

Partiendo del análisis sobre qué sistemas vigilan existen dos grupos de sistemas de detección de intrusiones: los que analizan actividades de una única máquina en busca de posibles ataques, y los que lo hacen de una subred (generalmente, de un mismo dominio de colisión) aunque se emplacen en uno sólo de los hosts de la misma. Esta última puntualización es importante: un IDS que detecta actividades sospechosas en una red no tiene porqué (y de hecho en la mayor parte de casos no suele ser así) estar ubicado en todas las máquinas de esa red.

- **IDS basado en red:** Monitoriza los paquetes que circulan por la red en busca de elementos que denoten un ataque contra alguno de los sistemas ubicados en ella. El IDS puede situarse en cualquiera de los hosts o en un elemento que analice todo el tráfico. Este tipo de IDS monitorizará diversas máquinas y no una sola.
- **IDS basado en host:** Monitoriza un único sistema. El IDS es un proceso que se ejecuta en *background* buscando patrones que puedan

denotar un intento de intrusión y alertando o tomando las medidas oportunas en cada caso. Los IDS basados en host se pueden subdividir a su vez en tres subcategorías:

- Verificadores de integridad del sistema (SIV): Monitoriza los archivos de una máquina (típicamente los relativos a la configuración y archivos del sistema) en busca de posibles modificaciones no autorizadas.
- Monitores de registros (LMF): Monitorizan los archivos de *log* generados por las aplicaciones en busca de patrones que denotan un ataque o intrusión.
- Sistemas de decepción o *honeypots*: Sistemas que simulan servicios con vulnerabilidades o problemas de seguridad de forma que un atacante piense que se trata de un sistema real en el que se registran todas las acciones del atacante.

Atendiendo a la segunda clasificación de los IDS, realizada en función de cómo actúan estos sistemas se encuentran dos grandes técnicas: las basadas en detección de anomalías (*anomaly detection*) y las basadas en la detección de usos indebidos del sistema (*misuse detection*).

- **Detección de anomalías:** La base del funcionamiento de estos sistemas es suponer que una intrusión se puede ver como una anomalía del sistema, por lo que si se pudiera establecer un perfil del comportamiento habitual de los sistemas se podría detectar las intrusiones haciendo uso de la estadística: probablemente una intrusión sería una desviación excesiva de la media del perfil de comportamiento habitual. Este esquema de detección se conoce como conocimiento positivo pues se basa en el conocimiento del comportamiento normal del sistema.
- **Detección de usos indebidos:** Es posible establecer patrones para los diferentes ataques conocidos y algunas de sus variaciones. Este esquema se limita a conocer lo anormal para poderlo detectar. Se conoce como un esquema de conocimiento negativo.

1.6.4. Requisitos de un IDS

Cualquier sistema de detección de intrusos ha de cumplir algunas propiedades para poder desarrollar su trabajo correctamente.

En primer lugar, el IDS ha de ejecutarse continuamente sin nadie que esté obligado a supervisarlo, independientemente de que al detectar un problema se informe a un operador o se lance una respuesta automática, el funcionamiento habitual no debe implicar interacción con un humano. Los

sistemas de detección son mecanismos automatizados que se instalan y configuran de forma que su trabajo habitual sea transparente a los operadores del entorno informático.

Otra propiedad a tener en cuenta es la aceptabilidad o grado de aceptación del IDS. Los mecanismos de detección de intrusos han de ser aceptables para las personas que trabajan habitualmente en el entorno. Por ejemplo, no ha de introducir una sobrecarga considerable en el sistema ni generar una cantidad elevada de falsos positivos o de logs, ya que entonces llegará un momento en que nadie se preocupe de comprobar las alertas emitidas por el detector.

La tercera característica es la adaptabilidad del mismo a cambios en el entorno de trabajo. Ningún sistema puede considerarse estático por lo que los IDS deben ser capaces de adaptarse rápidamente a esos cambios.

1.6.5. Estrategias de respuesta

El IDS debe ser capaz de generar respuestas automáticas ante los intentos de ataque. Entre estas opciones puede estar el bloqueo de la dirección atacante en el cortafuegos, envío de notificaciones (SMS, correo electrónico, ...), registro en logs. Es posible que el IDS detecte como un ataque un comportamiento lícito. Nunca se puede garantizar por completo que lo que un IDS categoriza como ataque realmente lo sea.

Un correcto esquema de respuesta automática debe contemplar los siguientes puntos:

- Probabilidad de que un ataque detectado sea un falso positivo: Es necesario asignar un peso específico a cada alarma generada por el sensor para actuar en el caso de detectar un patrón de ataque claro.
- Contemplar direcciones protegidas: Es difícil determinar si el tráfico que se recibe determina un verdadero ataque cuando provenga de direcciones autorizadas. En este caso, mejor que el bloqueo automático es mejor generar alertas para que se analicen y se tomen las acciones oportunas.
- Límite en el número de respuestas por unidad de tiempo: Si se produce la detección de demasiados patrones sospechosos en un periodo de tiempo pequeño es mejor generar alertas para que se verifique si son legítimas o no. Si se generaran muchas respuestas automáticas, por ejemplo de bloqueo de dirección IP origen, en un periodo de tiempo pequeño se podría acabar produciendo una auto denegación de servicio contra potenciales usuarios legítimos.

1.7. Objetivos

Dada la importancia de la ciberseguridad y el incremento del número de ataques que día a día se están produciendo es de vital importancia incrementar las medidas de seguridad en cada uno de los aspectos y ámbitos de la vida.

En este trabajo fin de máster se aborda la posibilidad de desarrollar un sistema de detección de intrusiones portable y económico, haciendo uso de herramientas *opensource* y dotándolo de una interfaz gráfica útil y sencilla para su uso por parte del usuario final.

Capítulo 2

Análisis y Planificación

2.1. Introducción

Tal y como se ha expuesto en el capítulo 1, la realización de este proyecto se hará en base a herramientas *opensource*. Para poder alcanzar la solución óptima en la fase de diseño, es necesario elaborar antes un análisis de las diferentes opciones que existen para poder elegir en base a argumentos sólidos la mejor tecnología a usar en cada caso.

En un segundo apartado se exponen la planificación detallada para la realización del proyecto y el análisis de costes.

2.2. Análisis

Para poder abordar en profundidad el análisis de las diferentes tecnologías existentes que puedan ser susceptibles de usarse en la elaboración del presente trabajo fin de máster, éste se ha dividido en diferentes secciones atendiendo a las diferentes tecnologías base a utilizar.

2.2.1. Software IDS

Este análisis se centra en herramientas IDS basadas en red y todas las analizadas se corresponden con software *opensource* cumpliendo así con los objetivos de este proyecto.

Snort

Snort[9] es un software para la detección y prevención de intrusiones basado en reglas. Fue desarrollado en 1998 por Martin Roesch[10]. Una de las

principales ventajas del uso de Snort es la capacidad de realizar análisis del tráfico y de recogida de log de paquetes. Incorpora funcionalidad de análisis de protocolos, inspección de contenido de paquetes y varios pre procesadores. Snort es una herramienta ampliamente reconocida y aceptada para detección de gusanos, exploits, escaneos de puertos y otros intentos de intrusión. Puede ejecutarse en tres modo de trabajo diferentes: *sniffer*, captura de paquetes y herramienta para detección de intrusiones. En el modo de trabajo *sniffer*, la herramienta únicamente lee paquetes y muestra información en consola. En el modo captura, los paquetes son almacenados en el disco. En el modo de detección de intrusiones, se monitoriza el tráfico en tiempo real y lo compara con las reglas definidas por el usuario.

Snort es capaz de detectar multitud de ataques como buffer overflow, escáneres de puertos, ataques CGI, sondas SMB, intentos de fingerprinting, etc. La última versión liberada es la 2.9.11.1.

La arquitectura de Snort[11] se muestra en la figura 2.1 y está compuesta de los siguientes elementos:

- **Sniffer:** captura el tráfico de red e identifica la estructura de los paquetes. Una vez con capturados los envía a los pre procesadores.
- **Preprocesadores:** realiza ciertas acciones para determinar el tipo de paquetes y el comportamiento que tiene que tener Snort hacia ellos. Con este conjunto de protocolos, tipos de escaneo y niveles de sensibilidad definidos es posible identificar el tipo de tráfico para poder realizar una mejor toma de decisiones.
- **Motor de detección:** compara cada paquete recogido con cada una de las reglas definidas. Si el resultado es satisfactorio, el paquete es enviado hacia la salida.
- **Salida:** es posible generar un log o lanzar una alerta en base a la acción definida en la regla. Los logs pueden ser almacenados en diferentes formatos y en diferentes localizaciones.

Los requerimientos hardware recomendados para instalar Snort son al menos una CPU con dos cores, 4 GB de memoria RAM y 30 GB de disco duro.

Pros

- Distribución gratuita y opensource para múltiples sistemas operativos.
- Facilidad de elaboración de reglas para detección de intrusiones.
- Altamente flexible y dinámico en términos de ciclo de vida y soporte.

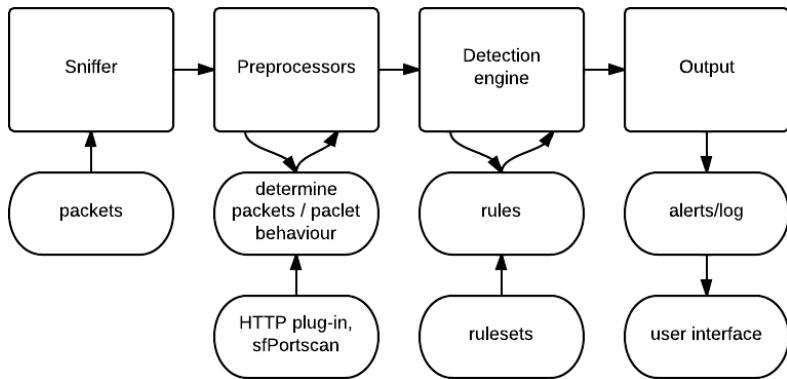


Figura 2.1: Arquitectura de Snort

- Comunidad amplia para soporte y solución de problemas.

Contras

- Carece de interfaz gráfica.
- Lentitud en el procesamiento de paquetes.
- No detecta división de firmas cuando está repartida sobre múltiples paquetes TCP.
- Alto número de falsos positivos. Requiere de un esfuerzo alto de ajuste e implementación.

Suricata

Suricata[12] es un software de código abierto, rápido y altamente robusto para la detección en tiempo real de intrusiones en red así como para la prevención de intrusiones y la gestión de seguridad en red. También proporciona herramientas para el análisis offline de capturas de tráfico. Este sistema fue desarrollado por la OISF[13]. Inspecciona el tráfico usando un potente y extenso conjunto de reglas y lenguaje de marcas. Incorpora además soporte para scripting Lua para detección de ataques complejos.

Posee formatos estándar de entrada y salida como YAML y JSON lo que facilita su integración con otras herramientas SIEM, Splunk, ELK, etc.

Pros

- Realiza el procesamiento del tráfico en las siete capas del modelo OSI mejorando la capacidad de detectar actividades maliciosas.
- Detecta y parsea automáticamente protocolos como IP, TCP, UDP, ICMP, HTTP, TLS, FTP, SMB y FTP permitiendo ejecutar reglas para todos ellos.
- Características avanzadas de procesamiento tales como multi-threading y aceleración por GPU.

Contras

- Menor comunidad en comparación con otros IDSs como Snort.
- Operación complicada.
- Alto número de falsos positivos. Requiere de un esfuerzo alto de ajuste e implementación.
- Requisitos de sistema elevados para una funcionalidad completa.

La arquitectura básica de Suricata es similar a la anteriormente explicada en el caso de Snort pero tal y como se muestra en la figura 2.2, Suricata es capaz de realizar un procesamiento en paralelo de los flujos entre los distintos cores de los que disponga el sistema. La última versión estable de Suricata es la 4.0.3 y está disponible para sistemas operativos Linux, Windows, FreeBSD y UNIX. Los requisitos mínimos recomendados de sistema para implementar Suricata son 4 GB de memoria RAM, 30 GB de almacenamiento y un procesador multicore para hacer uso del paralelismo en el análisis de flujos que implementa.

Bro

Bro[14] es un analizador de tráfico pasivo desarrollado originalmente por Vern Paxson[15] y usado para recolectar medidas o información acerca de los flujos de tráfico en una red, investigación forense, establecimiento de líneas base, etc. Bro consta de un conjunto de ficheros de log para almacenar las actividades de red como sesiones HTTP con URIs, cabeceras, tipos MIME, peticiones DNS, certificados SSL, sesiones SMTP. Además proporciona funcionalidad extra para el análisis y detección de ataques, extracción de archivos de sesiones HTTP, detección sofisticada de malware, vulnerabilidades software, ataques de fuerza bruta contra SSH y validación de la cadena de certificados.

La arquitectura interna de Bro está dividida en dos capas, el núcleo, también llamado motor de eventos y el intérprete de scripts. El núcleo reduce

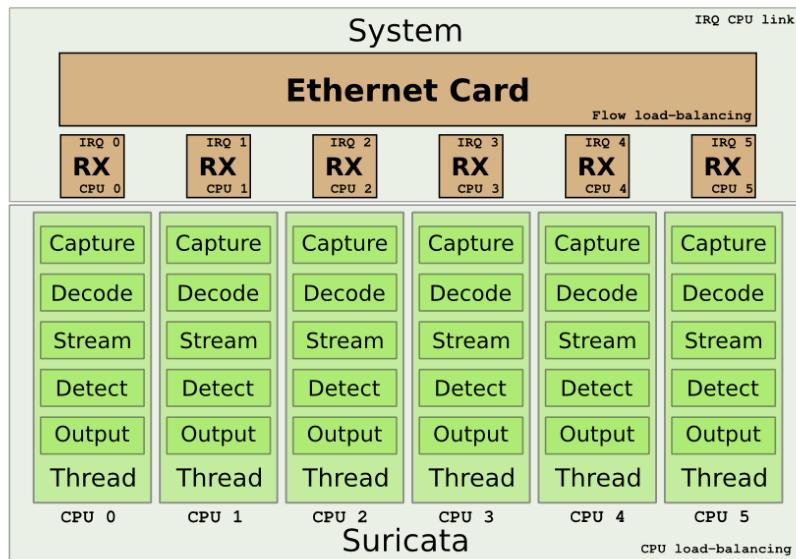


Figura 2.2: Arquitectura de Suricata

el flujo de paquetes de entrada en una serie de eventos de alto nivel. Estos eventos reflejan la actividad de la red en términos neutros, es decir, reflejan qué se ve pero no intenta explicar el porqué. Estos eventos son derivados a la segunda capa, el intérprete de scripts. Es el encargado de ejecutar un conjunto de gestores de eventos escritos en lenguaje propio de Bro. Estos scripts pueden expresar las políticas de seguridad y el tipo de acciones a tomar para diferentes tipos de actividad. También puede generar alertas en tiempo real o ejecutar software externo bajo demanda, por ejemplo, como respuesta activa frente a un ataque.

Pros

- Eficiente y flexible para adaptarse a multitud de entornos. No se resiente a la detección tradicional basada en reglas o firmas.
- Interfaces abiertas para aplicación con otras arquitecturas e intercambio de información en tiempo real.
- Posibilidad de automatización de tareas y respuesta activa frente a detecciones.

Contras

- Es necesaria experiencia en programación para poder hacer un completo manejo de Bro.

- Curva fuerte de aprendizaje.

2.2.2. Framework de desarrollo

El desarrollo de una interfaz gráfica para la gestión y configuración del sistema conlleva asociados muchos desafíos pero hay una serie de características comunes (manejo de rutas, accesos a base de datos, estructura del proyecto, seguridad, etc.) a tomar en cuenta a la hora de hacer la elección. Un framework nos ayuda a cubrir estas características de modo que los esfuerzos se centran en la lógica de la aplicación en lugar de tener que implementar todos los componentes como el acceso a base de datos o el envío de email.

En las siguientes secciones se expone el análisis de Codeigniter y Laravel como los principales frameworks de desarrollo PHP **codeigniterlaravel**.

CodeIgniter

CodeIgniter [16] se define como a un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo principal es permitir que los desarrolladores puedan realizar proyectos mucho más rápido. Esta herramienta brinda un conjunto de bibliotecas para tareas comunes, además brinda una interfaz simple y una estructura lógica para acceder a las bibliotecas.

Características

- **Versatilidad:** es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo se dispone de un acceso por FTP para enviar los archivos al servidor y donde no hay acceso a su configuración.
- **Facilidad de instalación:** solo se requiere de una cuenta FTP para subir Codeigniter al servidor y su configuración se realiza en base a la edición de un archivo de configuración. No es necesario el acceso a la línea de comandos para la instalación.
- **Flexibilidad:** es menos rígido que otros frameworks ya que define una manera de trabajar pero para algunos módulos es opcional seguirlo o no.
- **Ligereza:** permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. Los módulos o clases se pueden cambiar de manera opcional sólo cuando se vayan a utilizar realmente.

- **Documentación:** la documentación es fácil de seguir y asimilar al estar escrita en modo tutorial. En contrapartida no facilita la referencia rápida para consultas concretas.

Laravel

Laravel [17] es un framework opensource para desarrollar en PHP, con una filosofía muy clara enfocada para que el código sea lo más expresivo y elegante posible, para desarrollar aplicaciones y servicios web.

Características

- **Modular y extensible:** permite agregar lo que sea necesario para el desarrollo a través de Packalyst.
- **Micro-servicios y APIs:** posibilidad de usar Lumen como micro-framework para desarrollo de microservicios y APIs.
- **HTTP routing:** posee un sistema de enrutamiento rápido y eficiente similar al usado por Ruby on Rails.
- **HTTP middleware:** el uso de middleware permite analizar y filtrar las llamadas al servidor.
- **Caché:** sistema de caché configurable y robusto.
- **Autenticación:** incorpora autenticación de usuarios de forma nativa.
- **Integración con Stripe:** incluye lo necesario a través de Cashier para integrarlo con Stripe como sistema de cobro.
- **Tareas automatizadas:** con el API Elixir permite crear tareas de Gulp para definir el uso de pre procesadores para comprimir nuestro CSS y JavaScript.
- **Encriptación:** incorpora el uso de seguridad OpenSSL y cifrado AES-256-CBC.
- **Eventos:** sencillez para la definición, registro y escucha de eventos.
- **Paginación**
- **Object-Relational-Map (ORM)**
- **Unit Testing**
- **Cola de tareas (queues):** incorpora la opción para ejecutar procesos largos y complejos en segundo plano a través del uso de listas de tareas.

AngularJS

AngularJS [18] [19] es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

Este framework adapta y amplía el HTML tradicional para servir mejor contenido dinámico a través de un data binding bidireccional que permite la sincronización automática de modelos y vistas. Como resultado, AngularJS pone menos énfasis en la manipulación del DOM y mejora la testeabilidad y el rendimiento.

Objetivos de diseño

- Disociar la manipulación del DOM de la lógica de la aplicación. Esto mejora la capacidad de prueba del código.
- Considerar a las pruebas de la aplicación como iguales en importancia a la escritura de la aplicación. La dificultad de las pruebas se ve reducida drásticamente por la forma en que el código está estructurado.
- Disociar el lado del cliente de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.
- Guiar a los desarrolladores a través de todo el proceso del desarrollo de una aplicación: desde el diseño de la interfaz de usuario, a través de la escritura de la lógica del negocio, hasta las pruebas.
- Angular sigue el patrón MVVM (Model View View-Model) de ingeniería de software y alienta la articulación flexible entre la presentación, datos y componentes lógicos. Con el uso de la inyección de dependencias, Angular lleva servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente. En consecuencia, gran parte de la carga en el backend se reduce, lo que conlleva a aplicaciones web mucho más ligeras.

ReactJS

React [20] [21] es una biblioteca Javascript de código abierto para crear interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones en una sola página. Es mantenido por Facebook, Instagram y una comunidad de desarrolladores independientes y compañías.

React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar. React sólo maneja la interfaz de usuario en una aplicación; está construida únicamente para utilizar el patrón de diseño modelo–vista–controlador (MVC), y puede ser utilizada conjuntamente con otras bibliotecas de Javascript. También puede ser utilizado con las extensiones de React-based que se encargan de las partes no-UI (no gráficas) de una aplicación web.

2.2.3. Plataforma hardware

Raspberry Pi

Raspberry Pi es un SBC (single board computer), una computadora de placa única, es decir, una computadora completa en un sólo circuito. Ver figura 2.3. El diseño se centra en un sólo microprocesador con la RAM, E/S y todas las demás características de un computador funcional en una sola tarjeta que suele ser de tamaño reducido, y que tiene todo lo que necesita en la placa base.

Es una placa de desarrollo de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación.

Arquitectura y componentes: Raspberry Pi 3 Model B

La placa Raspberry Pi 3 Model B está construida sobre una arquitectura ARMv8-A [22], el intento de ARM por crear una arquitectura que sirva de base tanto para la futura generación de tablets y smartphones como para que sea su entrada tanto en el mercado de los servidores. ARMv8-A es totalmente compatible con las aplicaciones de ARMv7, añade 8 bits más respecto a la versión anterior (ARMv7) al direccionamiento de memoria con un total de 48 bits lo que junto con técnicas LPAE (Large Physical Address Extensions) se podrían llegar a los 256 TB de memoria.

Añade facilidad a la virtualización que puede ayudar a correr distintos sistemas al mismo tiempo sobre la misma máquina. Tiene tres conjuntos de instrucciones, A32, T32 y A64. Los dos primeros los hereda de su compatibi-

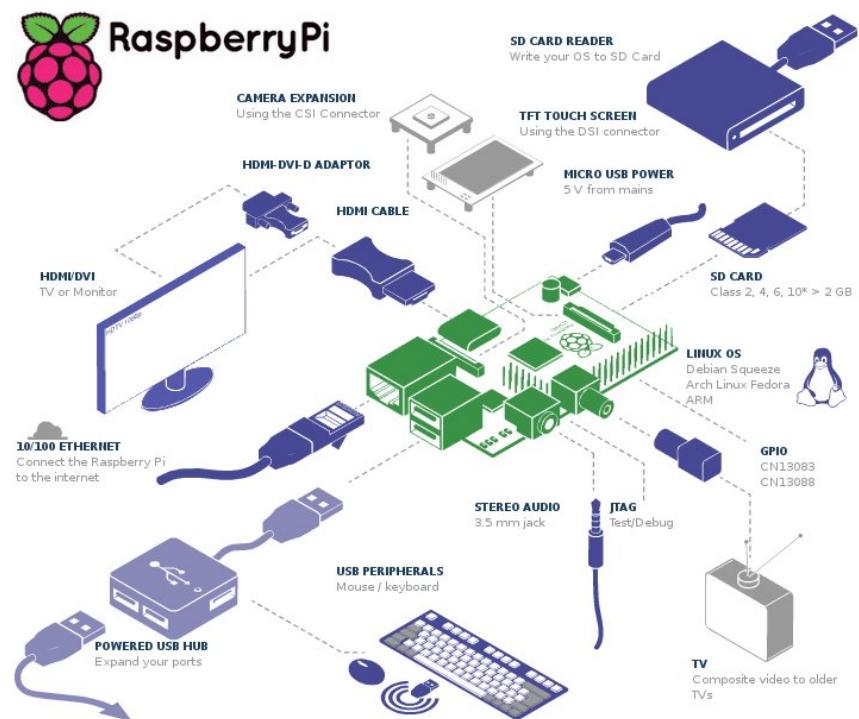


Figura 2.3: Raspberry Pi

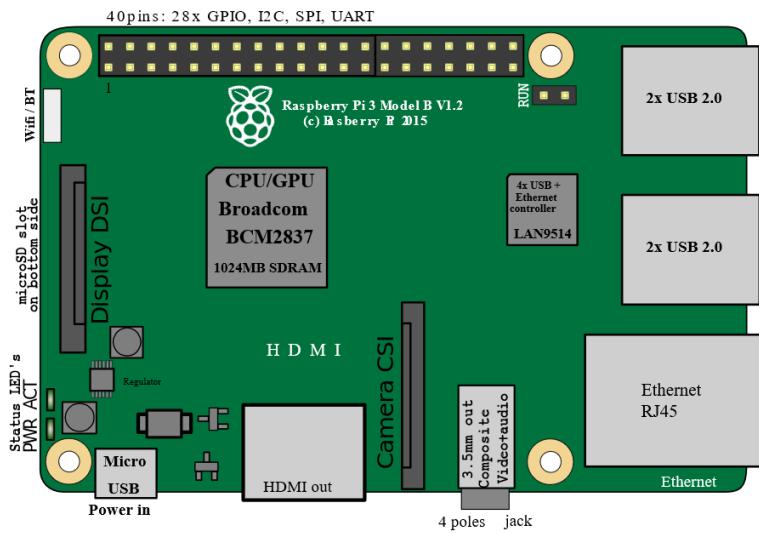


Figura 2.4: Arquitectura Raspberry Pi 3 Model B

lidad con ARMv7-A. Mejora un bloque funcional denominado NEON que ya existía en ARMv7-A pero como opcional. Este permite trabajar con datos de 8 y 16 bits como si fueran vectores. En esta versión tenemos más registros de 128 bits en concreto 32. De esta forma se pueden realizar 16 sumas de 8 bits a la vez. Así se aceleran muchas aplicaciones multimedia como son los codificadores de video o audio e incluso aplicaciones criptográficas.

Los registros de propósito general son de 64 bits, respecto a los 32 de la anterior versión mejorando la eficiencia ya que permite hacer más cosas con menos funciones. Además, añade conjuntos de instrucciones para criptografía como AES, SHA-1 y SHA-256. Estas instrucciones mejoran en 10 veces su velocidad gracias al bloque funcional NEON.

Los componentes de esta placa se pueden ver en la figura 2.4. En la tabla 2.1 se muestran las especificaciones de la placa analizada.

2.2.4. Sistema Operativo

Raspbian

Raspbian [23] es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Jessie (Debian 8.0) para la placa computadora (SBC) Raspberry Pi. Fue lanzado en Junio de 2012. Técnicamente, es un fork no oficial de Debian armhf para el procesador (CPU) de Raspberry Pi, con soporte optimizado para cálculos en coma flotante por hardware.

Al ser una distribución de GNU/Linux las posibilidades son infinitas. To-

Especificaciones Raspberry Pi 3 Model B	
Arquitectura	ARMv8-A
SOC	Broadcom BCM2837
CPU	1.2 GHz 64-bit quad-core ARM Cortex-A53
GPU	Dual Core VideoCore IV ® Multimedia Co-procesador
Memoria (SDRAM)	1 GB (compartida con GPU)
Puertos USB 2.0	4
Entrada de video	Interfaz de cámara MIPI (CSI) de 15 pines
Salida de video	HDMI rev 1.3 y 1.4. RCA compuesto (PAL y NTSC)
Salida de audio	Analógico jack de 3,5 mm. Digital
Almacenamiento	Ranura para tarjeta
Potencia	Entre 300 mA (1.5 W) en reposo y 1.34 A (6.7 W)

Cuadro 2.1: Especificaciones Raspberry 3 Model B

Características Raspbian	
Desarrollador	Raspberry Pi Foundation
Familia OS	Unix-like
Modelo desarrollo	Open source
Última release	Raspbian Stretch con PIXEL / 29/11/2017
Mercado objetivo	Raspberry Pi
Idiomas	Inglés
Método de actualización	APT
Gestor de paquetes	dpkg
Plataformas	ARM
Arquitectura Kernel	Monolítico

Cuadro 2.2: Características Raspbian

do software de código abierto puede ser recompilado en la propia Raspberry Pi para arquitectura armhf que pueda ser utilizado en el propio dispositivo. Además esta distribución, como la mayoría, contiene repositorios donde es posible descargar multitud de software. Esto hace que la Raspberry Pi expanda su funcionalidad puesto que además de servir como placa con microcontrolador básico añada funcionalidades de un PC.

Existen dos versiones de Raspbian: *stretch lite* que incorpora la mínima funcionalidad para añadir los diferentes componentes según sean necesarios y *stretch desktop*, imagen de Raspbian con entorno gráfico PIXEL, basado en LXDE y Midori como navegador web. Además contiene herramientas de desarrollo como IDLE para el lenguaje de programación Python o Scratch.



Figura 2.5: Flujo de ELK

2.2.5. Monitorización y explotación de logs

Stack ELK

ELK es un conjunto de herramientas de código abierto que se combinan para crear una herramienta de administración de registros que permite monitorizar, consolidar y analizar los logs generados en múltiples servidores. Estas herramientas tienen como objetivo resolver algunos inconvenientes que surgen del trabajo y la gestión de logs como puede ser la detección de incidencias en tiempo real, almacenamiento de gran cantidad de información, tiempo de respuesta y escalabilidad.

Las herramientas que conforman el stack ELK son Elasticsearch, Logstash y Kibana.

- **ElasticSearch:** es un motor de búsquedas y análisis que se encarga del almacenamiento de los datos ya optimizados por la indexación. Es un sistema distribuido, tolerante a fallos y puede ser desplegado en esquemas de alta disponibilidad.
- **Logstash:** lee los datos de diferentes fuentes simultáneamente, los trata en la medida que sea necesaria para que se almacenen en Elasticsearch.
- **Kibana:** motor de visualización de los datos almacenados en Elasticsearch permitiendo dar forma a sus datos de manera personalizada.

La figura 2.5 muestra el flujo de ELK donde Logstash obtiene todos los mensajes de logs, transformándolos según requiramos, para luego ser almacenados en Elasticsearch permitiendo así visualizarlos y manejarlos con Kibana.

Entre los problemas que trata de solventar Logstash se encuentran:

- **Falta de consistencia:** Dificultad para realizar búsquedas entre los distintos formatos de logs que generan los diferentes dispositivos, es decir, cada dispositivo tiene su propio formato de logs.

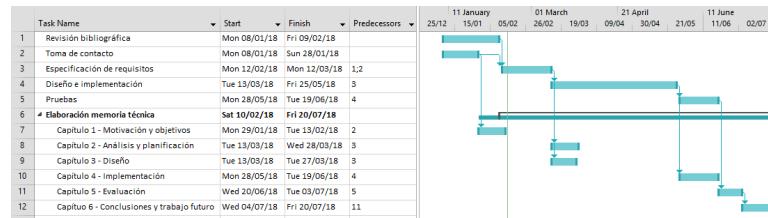


Figura 2.6: Diagrama de Gantt planificación del proyecto

- **Formato de tiempo:** Cada log puede tener un tipo de formato de tiempo diferente por ejemplo: Oct 04 12:15:21, 020289 07:23:24, 150260505, entre otros.
- **Descentralización:** Significa que los logs se encuentran repartidos en todos los distintos servidores que tengamos disponibles y para encontrar algún log en específico se tendría que buscar en cada servidor hasta encontrar dicho log. Es sencillo realizar esta tarea cuando se tienen pocos servidores, pero cuando se trata de 60 servidores o más la tarea es bastante compleja.
- **Falta de conocimiento:** Existen usuarios finales que utilizan servicios, en caso de que le ocurra alguna falla necesitan monitorizar dicho servicio para indagar que le sucedió, pero probablemente nunca han trabajado con logs y desconocen la ruta de los logs, como se actualizan, y puede que no entiendan lo que expresa una línea de log.

2.3. Planificación

En este apartado se describen las principales tareas de las que se compone el trabajo fin de máster incluyendo el desarrollo temporal de las fases del proyecto y quedando recogidas en el diagrama de Gantt mostrado en la figura 2.6.

A continuación se describen brevemente cada una de estas fases.

- **Revisión bibliográfica:** en esta fase se pretende realizar una investigación sobre las diferentes herramientas y sistemas así como posibles implementaciones similares que se hayan desarrollado. Esta recopilación de información se utiliza como fuente de inspiración y referencia.
- **Toma de contacto con herramientas:** durante esta fase se configurarán diferentes entornos virtualizados sobre los que instalar, analizar y evaluar las diferentes herramientas.

- **Especificación de requisitos:** en esta fase se describen todos los requisitos de diseño y funcionalidades a implementar.
- **Diseño e implementación:** en esta fase se realiza el diseño completo del sistema tanto a nivel hardware, software y de comunicaciones.
- **Pruebas:** verificación del correcto funcionamiento del sistema implementado y el software desarrollado.
- **Elaboración de memoria técnica:** realizar la documentación de cada uno de los aspectos relacionados con el trabajo fin de máster.

2.3.1. Recursos utilizados

En esta sección se describen todos los recursos que han sido empleados durante el desarrollo del proyecto clasificando cada uno de ellos en: humanos, hardware y software.

Recursos humanos

- D. Javier Tallón Guerri
- D. Antonio Miguel Ramírez Oliva

Recursos hardware

- Raspberry Pi 3 Model B
- Ordenador Portátil
- Fuente de alimentación
- Tarjeta SD
- Switch HP Procurve J9079A
- Cable de red UTP categoría 5E
- Router TP-Link TL-WR841N

Recursos software

- VirtualBox
- Debian 9.3
- Raspbian

- Bro
- Netsniff
- SSMTP
- Loki
- CriticalStack
- PHP Laravel
- AngularJS
- ELK + Logstash

2.3.2. Estimación de costes

Recursos humanos

La duración total estimada para la realización del proyecto según la planificación temporal mostrada en la figura 2.6 es de 28 semanas. Esta planificación se ha realizado atendiendo a una jornada laboral estándar de 5 días por semana y con una dedicación diaria del 25 % de una jornada laboral de 8 horas. Bajo esta premisa, la dedicación del alumno para la realización de este trabajo fin de máster será de 280 horas.

Con respecto a la dedicación del tutor del mismo, se estima necesario un trabajo de 30 horas dedicadas a las diferentes tutorías, revisión y correcciones necesarias y resolución de dudas.

De acuerdo a los honorarios de un Graduado en Ingeniería de Telecomunicaciones según la Junta General del Colegio Oficial de Ingeniero de Telecomunicación, los honorarios de un Ingeniero de Telecomunicación, son libres y responden al libre acuerdo entre el profesional y el cliente. Dada esta situación, deben definirse según una serie de elementos como costes del ingeniero, desplazamientos, etcétera. Teniendo en cuenta estos elementos se han fijado unos honorarios de 30 €/hora netos para el alumno y para el tutor. Teniendo en cuenta la cantidad de horas establecida para la realización del proyecto, así como los honorarios fijados para los diferentes recursos humanos, la estimación de costes en este sentido vienen recogidos en el cuadro 2.3.

Recursos hardware

Para la evaluación del coste asociado a los recursos hardware a utilizar se toma el precio medio de mercado de cada componente a utilizar. En el cuadro 2.4 se muestra un resumen de ellos.

Estimación de costes relativos a recursos humanos			
Concepto	Coste/Hora	Cantidad	Total
Trabajo alumno	30 €/h	280 horas	8400 €
Trabajo tutor	30 €/h	30 horas	900 €
Total			9300 €

Cuadro 2.3: Estimación de costes relativos a recursos humanos

Estimación de costes relativos a recursos hardware				
Concepto	Coste unitario	Amortización	Horas uso	Prorrteado
Raspberry Pi 3 Model B	37 €	36 meses	90 horas	0.13 €
Ordenador portátil	1200 €	36 meses	280 horas	11.23 €
Fuente de alimentación	10 €	36 meses	90 horas	0.035 €
Tarjeta SD	24 €	36 meses	90 horas	0.084 €
Switch HP Procurve J9079A	115 €	36 meses	90 horas	0.40 €
Cable de red UTP categoría 5E	1 €	36 meses	90 horas	0.010 €
Router TP-Link TL-WR841N	20 €	36 meses	90 horas	0.070 €
Total			12 €	

Cuadro 2.4: Estimación de costes relativos a recursos hardware

Recursos software

En cuanto a los recursos software empleados, todos los programas y herramientas utilizadas disponen de licencias gratuitas o se han usado licencias de prueba, por lo que en este caso es posible considerar que el coste asociado a este tipo de recursos es nulo.

Presupuesto total del proyecto

La estimación total del coste del proyecto se puede calcular como suma de las estimaciones de los costes calculados para cada apartado.

Estimación de costes totales	
Costes	Estimación
Costes recursos humanos	9300 €
Costes recursos hardware	12 €
Costes recursos software	0 €
Total	9312 €

Cuadro 2.5: Estimación de costes totales

Capítulo 3

Diseño

En el presente capítulo se expone todo el diseño del sistema realizado, las herramientas y los diferentes módulos que serán instalados junto con su integración entre ellos y el diseño y funcionalidades a añadir en la interfaz web para controlar el sistema que será desarrollada.

3.1. Diseño de la conectividad física para la recogida de datos

El primer punto a tener en cuenta en el diseño es el de la arquitectura de red a utilizar y el método para obtener el tráfico para su análisis en búsqueda de patrones o conexiones maliciosas. Para conseguir esto se opta por una solución SPAN, también conocido como port mirror. Con esta solución, una copia del tráfico que circula por un puerto del switch es replicado en otro. Es la solución elegida en base a su bajo coste, la posibilidad de configuración remota y la capacidad de recoger el tráfico interior al switch.

Teniendo en cuenta lo anteriormente expuesto se podrá conectar el sistema a cualquier puerto del switch que esté configurado como puerto SPAN del puerto por el que circula el tráfico que se quiere analizar. En la implementación realizada para este trabajo fin de máster se ha optado por analizar el tráfico que circula en la conexión entre el router y el proveedor de servicios de internet ya que de esta forma y dado que el router también tiene integrado el punto de acceso WiFi es posible capturar todo el tráfico de entrada y salida de nuestra red. El detalle del esquema de conectividad que habrá que implementar se muestra en la figura 3.1.

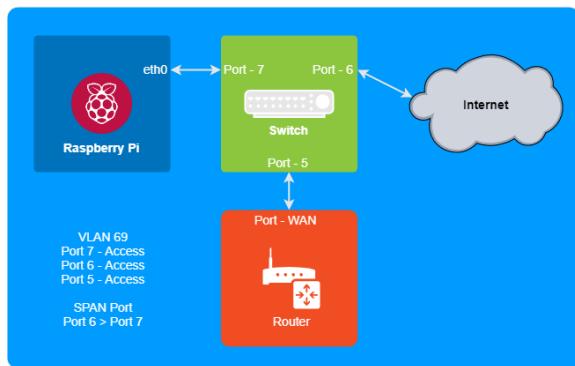


Figura 3.1: Diseño de la conectividad física para la recogida de datos

3.2. Diseño de la arquitectura de red y de gestión

Aunque la placa utilizada para la implementación del sistema (Raspberry Pi) posee de salida de video tipo HDMI, es necesario dotarla de conectividad para que sea accesible tanto para la gestión y acceso remoto del sistema como para el envío y recogida de logs por parte de un servidor centralizado. Aprovechando la conectividad de la tarjeta WiFi integrada en la placa se procederá a conectarla a una red LAN inalámbrica con conexión a internet a través desde la cual se podrá acceder al equipo y además se creará una red privada virtual a través de la cual será posible acceder a ella desde ubicaciones remotas sin conexión directa y que se conecte con el servidor de recogida de logs. Las ventajas de hacer uso de una conexión privada para las conexiones remotas al dispositivo en lugar de hacer NAT de puertos sobre un acceso a internet son más que evidentes pues de esta forma se evita la exposición de los servicios necesario a Internet con el ahorro consecuente de escaneos de puertos, intentos de conexión no autorizados, etc. Una visión completa de este esquema se puede encontrar en la figura 3.2.

3.3. Diseño de las herramientas y módulos de seguridad

El diseño de las diferentes herramientas que compondrán el sistema IDS y los diferentes módulos a integrar es de vital importancia ya que constituirán el motor de seguridad del sistema.

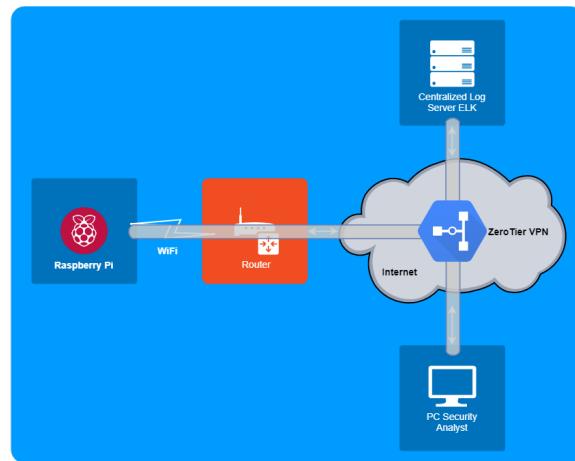


Figura 3.2: Diseño de la arquitectura de red y de gestión

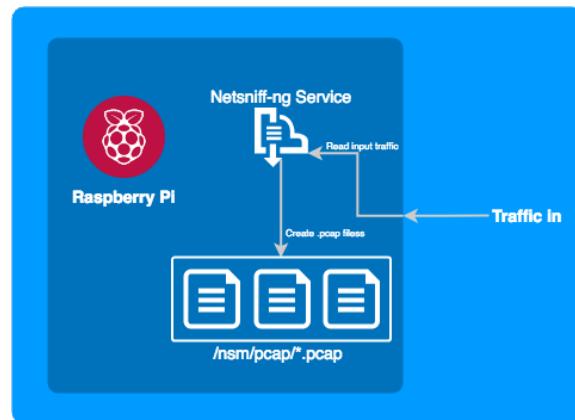


Figura 3.3: Captura del tráfico de red

3.3.1. Captura del tráfico de red

Uno de los puntos a cubrir con las herramientas a desplegar es la recogida de capturas de red del tráfico que recibirá el sistema. Esto es interesante por el motor principal Bro solo proporciona información agregada acerca de conexiones y estadísticas lo que nos permite añadir cierto contexto al análisis pero a costa de perder los detalles del tráfico. Para solventar esto y poder disponer del detalle del tráfico, a través de **netsniff-ng** se debe de configurar un servicio en el que se recojan, en archivos de 100 MB, todo el tráfico que circule por el sistema. Dichas capturas estarán en formato *.pcap*. El esquema de este servicio se puede encontrar en la figura 3.3.

Es posible poder dotar al sistema de una cantidad considerable de almacenamiento utilizando una tarjeta micro sd de mayor tamaño pero, a fin de

cuentas, es finita. Será necesario crear un script que se ejecute de manera recurrente para asegurar que la ocupación de estas capturas de tráfico no exceda los 25 GB. Esta es la configuración por defecto pero será configurable a través de la interfaz web de gestión.

3.3.2. Análisis del tráfico con Bro e integración con CriticalStack

Como base principal del sistema IDS se utilizará **bro** junto con sus herramientas adicionales **broctl**, **bro-common** y **bro-aux**. Bro será el encargado de analizar todo el tráfico recibido y de generar los logs correspondientes para su posterior explotación. Dichos logs, para una más fácil integración en el sistema centralizado de recogida de logs, deberán de estar en formato *json*.

Una de las características esenciales de Bro es la posibilidad de poder configurar y usar módulos adicionales para ampliar su funcionalidad. Habrá que retocar la configuración de Bro para incluir el módulo de CriticalStack para poder detectar conexiones maliciosas en base a los feeds a los que es posible suscribirse en su propia web. Otro módulo a incluir es el **extract** que permite guardar una copia de los ficheros que sea capaz de detectar dentro de los flujos de tráfico que analiza. También se añadirá al script que asegura que no se llena el espacio de almacenamiento una función para limpiar los ficheros extraídos más antiguos y asegurar que no se ocupan más de 2.5 GB.

Para asegurar que Bro está continuamente en ejecución se configurará una tarea recurrente que haciendo uso de **broctl** se encargue de relanzar el servicio. También se generará una tarea recurrente para la actualización de las firmas de CriticalStack y el relanzado de Bro para que aplique los cambios. En la interfaz web se podrá ver la fecha de la última actualización así como poder lanzar esta tarea de forma manual. Un resumen de esta funcionalidad se incluye en la figura 3.4.

3.3.3. Análisis de malware sobre los ficheros extraídos

Para poder asegurar una mejor seguridad proporcionada por este sistema se instalará **Loki** como motor antimalware para el análisis de los archivos que Bro haya extraído de los flujos de tráfico basado en reglas YARA. YARA es una herramienta de código abierto para la identificación de malware la cual utiliza una gran variedad de técnicas. Su principal característica es su flexibilidad. Además, es de gran ayuda en situaciones de respuesta a incidentes, en las cuales tanto las herramientas como el tiempo, suelen ser limitados.

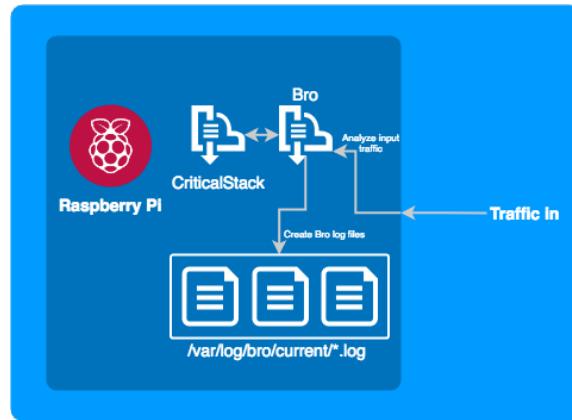


Figura 3.4: Análisis del tráfico con Bro e integración CriticalStack

3.3.4. Estructura de directorios

En esta sección se describe la estructura de directorios a implementar en el sistema y dónde estará ubicado cada tipo de archivos de los que conformarán este sistema de detección de intrusos portátil.

```
/nsm
|-- bro
|   |-- extracted
|   |-- logs
|-- Loki
|-- pcap
|-- scripts
    |-- cleanup
    |-- scan
    |-- update
```

Script 3.1: Estructura de directorios del sistema

Como se muestra en la referencia 3.1 la estructura de directorios a crear partirá de una carpeta `nsm` ubicada en la raíz del sistema de archivos dentro de la cual se crearán las carpetas `bro` para guardar los ficheros que se extraigan y con un enlace a la ruta donde se almacenan los logs en tiempo real de Bro, `Loki` donde estará instalado Loki junto con las reglas para los diferentes indicadores de compromiso y reglas YARA, `pcap` para almacenar las capturas de tráfico del servicio Netsniff-ng y `scripts` en donde se encontrarán los scripts para ejecutar el limpiado de la carpeta `extracted` y `pcap`, el script para ejecutar el análisis sobre los ficheros extraídos y el script para actualizar tanto las firmas de CriticalStack como de Loki.

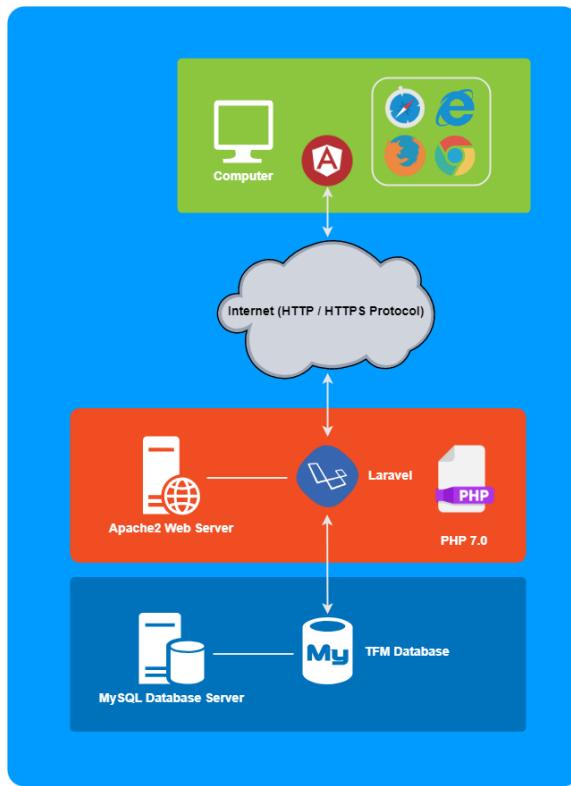


Figura 3.5: Arquitectura LAMP para la interfaz web

3.3.5. Arquitectura para la interfaz gráfica

La interfaz gráfica de gestión correrá sobre un servidor Apache2 y estará desarrollada usando el framework de desarrollo Laravel con lo que hará uso de PHP para la generación del contenido dinámico de la interfaz web. Para los usuarios del sistema se usará un motor de base de datos MySql sobre el que crear una base de datos para este proyecto y en donde se encontrarán las tablas necesarias, entre ellas la de los usuarios para la interfaz web. Se usará una arquitectura LAMP estándar como la mostrada en la figura 3.5

3.3.6. Diseño de la interfaz gráfica

Dentro del diseño de la interfaz gráfica se usarán distintos componentes y frameworks. Para la base de la aplicación se usará Laravel como framework de desarrollo sobre el que desplegar la aplicación y sobre el que configurar una API para la consulta e interacción desde el front-end hacia el back-end. En lo relativo al front-end se desarrollará usando la herramienta *Blade* integrada en Laravel junto con *AngularJS* para la actualización de valores

Ruta	Controlador y función que ejecuta
GET /login	PagesController@showLogin
POST /login	LoginAppController@handleLogin
POST /logout	LoginAppController@logout
GET /home	UsersController@getHome
GET /dashboard	DashboardController@getDashboard
GET /broconfig	UsersController@getBroconfig
GET /kibana	UsersController@getKibana
GET /loginregister	UsersController@getLoginRegisters

Cuadro 3.1: Rutas definidas para la navegación web, controladores y funciones que ejecutan

dinámicos sobre la interfaz web sin la necesidad de recargar la página. Los estilos y disposición de la página así como para el resto de los elementos visuales se utilizará *Bootstrap*.

El uso del framework de desarrollo Laravel automatiza el uso de modelos vista-controlador. En la tabla 3.1 se muestran las rutas definidas en el fichero `/routes/web.php` que definen el árbol de navegación por las distintas secciones de la web.

A excepción de las rutas `/login` tanto para GET como para POST, las demás hacen uso del middleware `auth` para controlar que sólo usuarios autenticados tienen acceso a estas rutas. Dentro de la llamada que genera el contenido de cada sección se comprueba el nivel de privilegios del usuario y en función de él se le permite acceder a una información o a otra.

Diseño de la página de login

La página de inicio de la aplicación será un formulario web con una disposición similar al mostrado en la figura 3.6 en donde además del logotipo aparecerán dos campos para introducir el nombre de usuario y la contraseña.

Diseño de la página Home

Tras pasar este menú de autenticación se accede a la página de bienvenida o *Home* del sistema. En dicha página deberá de aparecer en la parte superior de la página el menú de navegación con los enlaces a las distintas secciones junto con nombre del usuario que ha iniciado sesión junto con un botón para poder hacer logout. El contenido de la página constará de un banner en la parte superior dando la bienvenida junto con un texto con la descripción de las distintas funcionalidades y secciones que tiene esta interfaz de gestión. En la parte derecha de la página aparecerá un menú vertical con los datos e

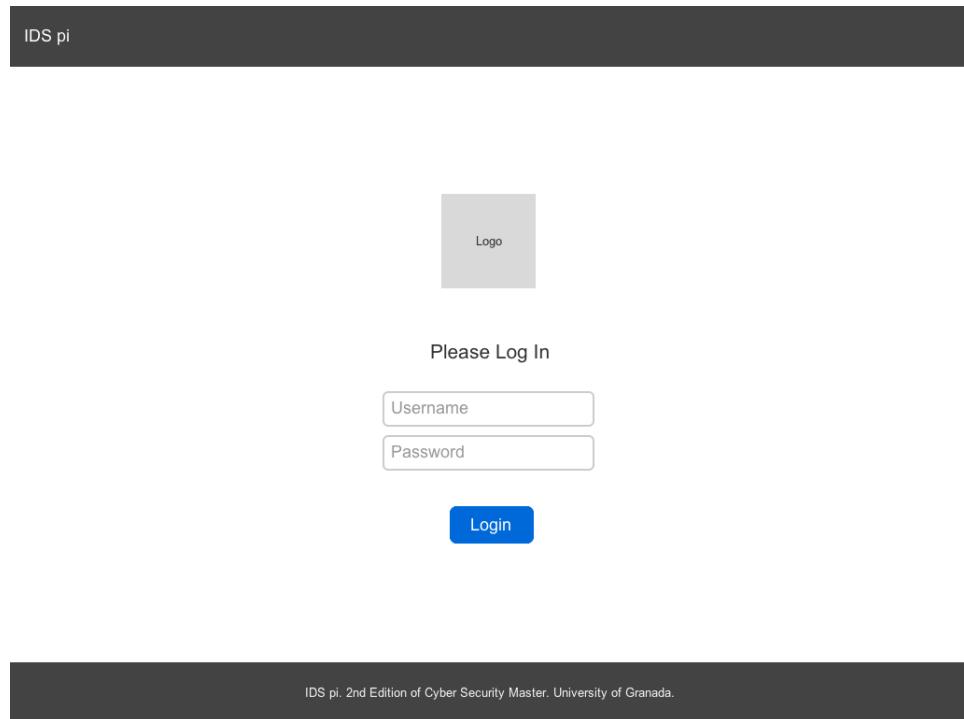


Figura 3.6: Wireframe de la página de login

información de contacto para el soporte de la aplicación. Se puede observar un diseño esquemático en la figura 3.7.

Diseño de la página Dashboard

Haciendo uso del menú de navegación de la parte superior de la página es posible moverse entre las distintas secciones. Una de ellas es *Dashboard* y en ella aparece de un vistazo la información del estado actual del sistema. Según se refleja en la figura 3.8, en primer lugar deberán de aparecer tres medidores tipo reloj para marcar el porcentaje de utilización tanto del disco, la CPU como la memoria. Debajo de estos aparecerá la información acerca del número de procesos en ejecución y el número de cores del sistema.

En la parte inferior de la página se mostrarán dos tablas desplegables, una con la lista de todos los procesos en ejecución del sistema y otra con todas las conexiones de red. A través de *AngularJS* la información de esos medidores y tablas se actualizará de manera automática cada segundo y se refrescará en pantalla. Para ello se configurarán las siguientes rutas en la API del sistema lo que desencadenará la ejecución de las funciones escritas en PHP para devolver estos valores. Las llamadas a la API solo serán accesibles para usuarios logueados en el sistema.

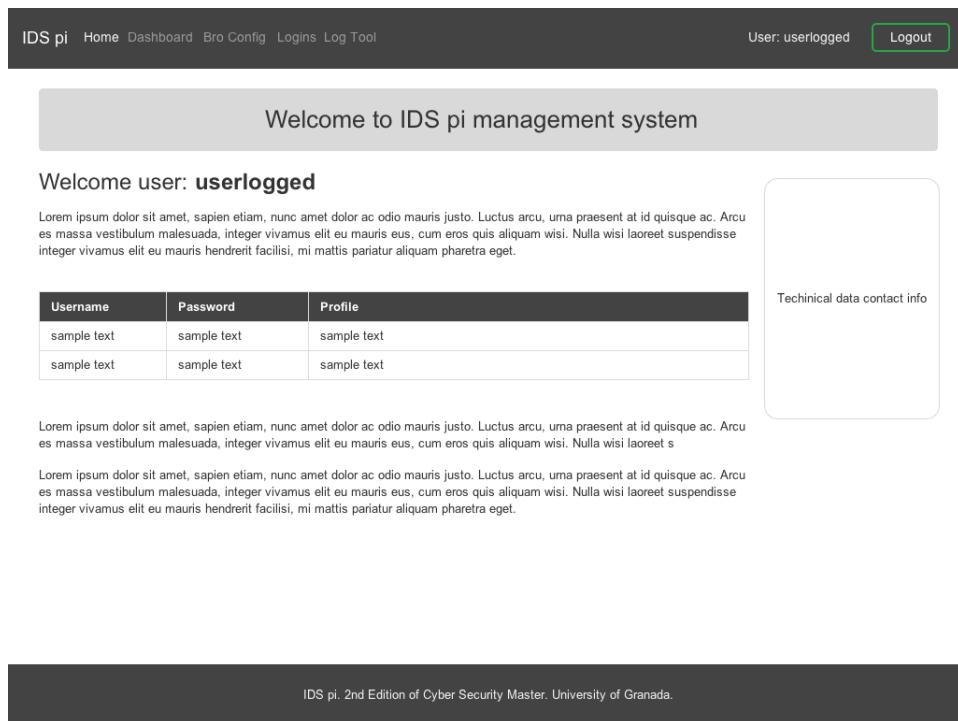


Figura 3.7: Wireframe de la página Home

En la tabla 3.2 se recoge la lista de las rutas que definen la API para obtener la información para visualizar en los dashboards junto con la función que ejecuta en el controlador. Esta API está protegida por el middleware `auth` para permitir sólo peticiones de usuarios autenticados en el sistema.

Una vez con la API definida y los métodos implementados para recopilar la información que devuelven, la nomenclatura de los métodos es bastante autodescriptiva, desde el front-end se ejecutará una llamada cada segundo de manera asíncrona a la URL `/api/dashboard/full` devolviendo un array con todos los valores necesarios. A partir de aquí, estos valores se representan haciendo uso de las bibliotecas *AngularJS* para los diferentes tipos de visualización de datos de DevExtreme, desarrolladas por DevExpress, [24]. El código fuente que se encarga de todas estas representaciones es `/public/js/gauges.js`.

Diseño de la página Bro Config

La sección de la interfaz web *Bro Config* se encargará de representar la configuración del sistema IDS implementado con Bro y Critical Stack. En la parte superior aparecerá la configuración de hostname y hora del sistema. Debajo el estado de Bro, las redes locales configuradas y la última fecha

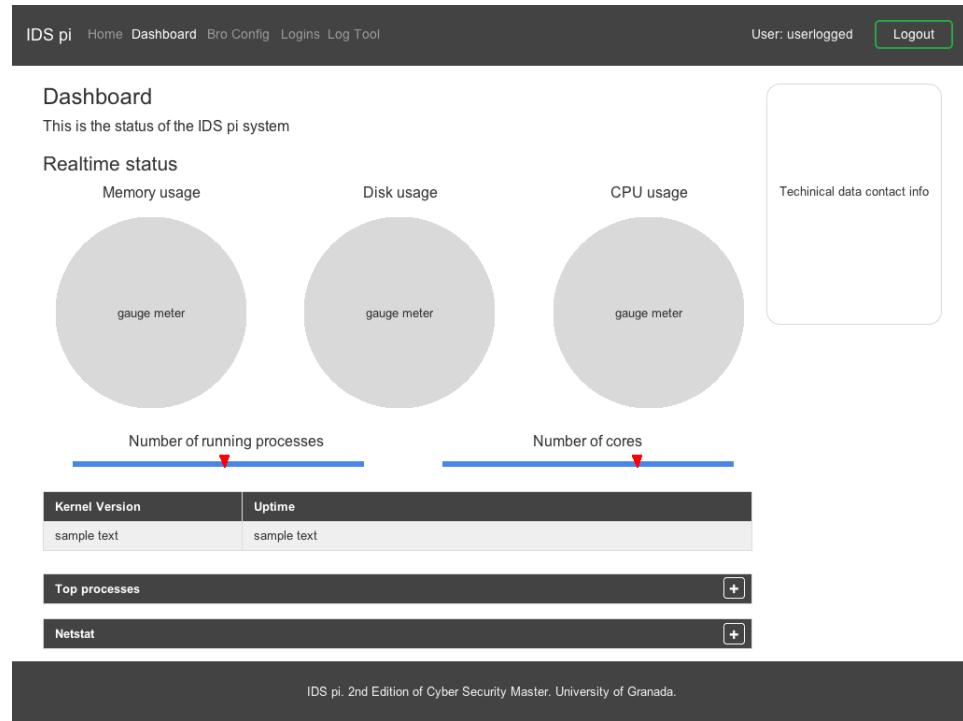


Figura 3.8: Wireframe de la página Dashboard

Ruta	Controlador y función que ejecuta
GET /api/dashboard/memory	DashboardController@getMemoryUsagePercent
GET /api/dashboard/disk	DashboardController@getDiskUsage
GET /api/dashboard/cpu	DashboardController@getCPUUsagePercent
GET /api/dashboard/numproc	DashboardController@getNumberOfProcesses
GET /api/dashboard/kernel	DashboardController@getKernelVersion
GET /api/dashboard/numcores	DashboardController@getNumberOfCores
GET /api/dashboard/top	DashboardController@getTop
GET /api/dashboard/netstat	DashboardController@getNetstat
GET /api/dashboard/full	DashboardController@getFullApi

Cuadro 3.2: Rutas definidas para la API Dashboard, controladores y funciones que ejecutan

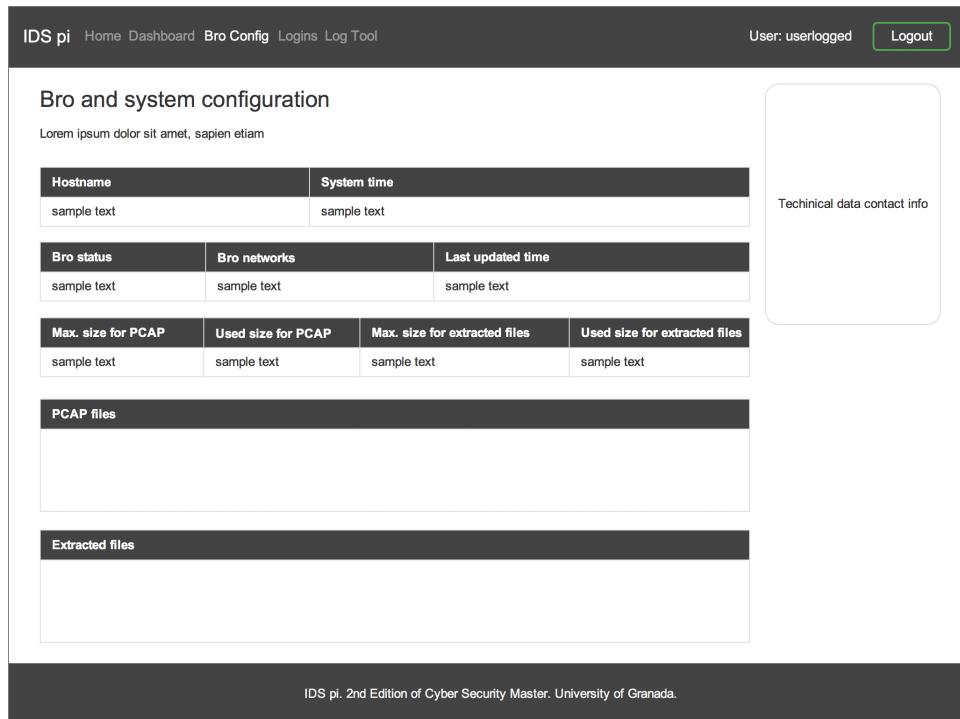


Figura 3.9: Wireframe de la página Bro Config

Ruta	Controlador y función que ejecuta
GET /apibro/broconfig/fullsystem	BroConfigController@getFullSystem
GET /apibro/broconfig/time	BroConfigController@getApiTime

Cuadro 3.3: Rutas definidas para la API Bro Config, controladores y funciones que ejecutan

de actualización y a continuación el resumen del espacio asignado para el almacenamiento de PCPAPs y de archivos extraídos. En la parte final de la vista dos exploradores de archivos para poder descargar los PCAPs o los ficheros extraídos para poder realizar un análisis en profundidad.

Para alimentar al front-end de los valores necesarios para la correcta representación de los datos se implementarán una serie de métodos en el controlador *BroConfigController* al que se accede a través de la siguiente llamada a la API 3.3.

En la tabla 3.4 se muestran los métodos que se llamarán dentro del método *BroConfigController@getFullSystem*.

Método	Valor devuelto
getHostname()	Hostname del sistema
getTime()	Hora del sistema
getUpdateTime()	Fecha de la actualización de firmas
getMaxPcapSize()	Tamaño máximo para guardar PCAPs
getMaxPcapSizeUsed()	Tamaño usado por PCAPs
getMaxFileExtractedSize()	Tamaño máximo para guardar ficheros
getMaxFileExtractedSizeUsed()	Tamaño usado por los ficheros
getBroNetworks()	Redes locales configuradas en Bro
getBroStatus()	Estado de Bro

Cuadro 3.4: Métodos llamados desde la función `getFullSystem`

Diseño de la página Logins

Dentro de este menú se mostrará una tabla con los accesos, o los intentos de acceso, que haya habido a la interfaz de gestión, junto con la información de la fecha y la hora cuando se produjeron y la acción realizada. El usuario *admin* deberá poder visualizar todos los que hayan ocurrido mientras que el usuario *readonly* sobre podrá visualizar los suyos propios. Un diseño esquemático puede verse en la figura 3.10.

Diseño de la página Log Tool

En esta visualización, dentro del menú y del entorno de navegación de la interfaz se incluirá un *iframe* en donde se provea de un acceso directo a Kibana como interfaz para la explotación de los logs y de un enlace para poder abrirlo en una ventana nueva. El diseño aparece reflejado en la figura 3.11.

3.3.7. Recogida de logs, envío y tratamiento

Todos los logs del sistema y del servidor Apache2 se recogerán sobre la ruta `/var/log`. Haciendo uso de una instancia de Filebeat junto con los módulos *system* y *apache2* se enviarán a la instancia de *Elasticsearch* que estará en ejecución sobre el servidor central de logs. Posteriormente, haciendo uso de Kibana y los diferentes dashboards y visualizaciones que estos módulos traen incorporados por defecto se podrá explotar esta información.

Los logs en tiempo real de Bro, a los que será posible acceder a través de un enlace desde `/var/log/bro/current` se enviarán, en formato *json*, a la instancia de Logstash en el servidor de recogida de logs. Logstash deberá de añadir la información de geo localización de direcciones IP y parsear

Figura 3.10: Wireframe de la página Logins

el campo *message* para extraer todos sus campos a campos individuales dentro del json. Una vez estos logs sean tratados se enviarán a Elasticsearch para su indexación y posteriormente a través de Kibana se podrán crear las visualizaciones y dashboards que puedan ser requeridos.

Esta arquitectura se ejemplifica de manera gráfica en la figura 3.12.

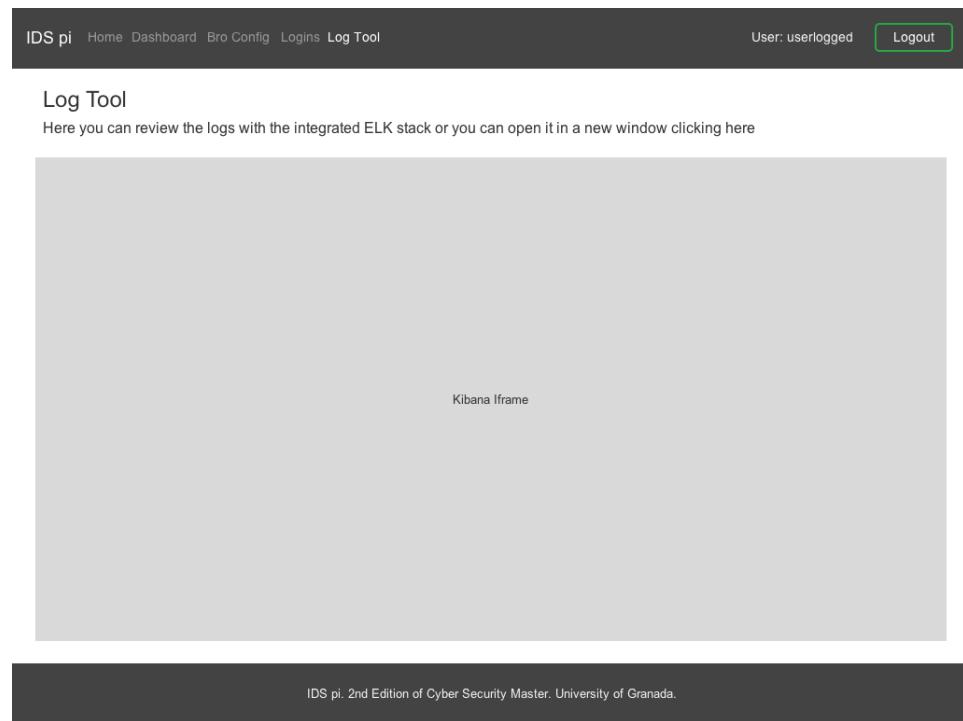


Figura 3.11: Wireframe de la página Log Tool

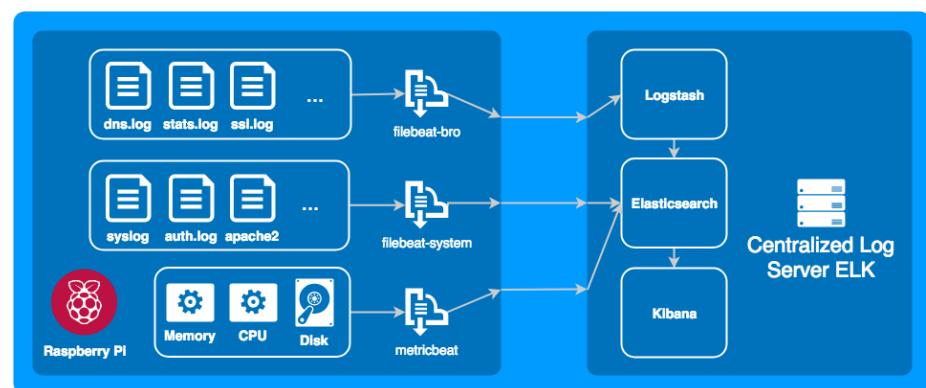


Figura 3.12: Arquitectura para la recogida de logs, envío y tratamiento

Capítulo 4

Implementación

4.1. Introducción

Una vez completadas las fases anteriores se comienza con el despliegue e instalación de todas las herramientas desarrolladas y con la realización de las diferentes pruebas de las distintas funcionalidades así como la recogida y tratamiento de logs.

4.2. Instalación de Raspberry Pi

4.2.1. Instalación del sistema operativo

Para la instalación del sistema se ha utilizado la última versión disponible de Raspbian Stretch Lite, la versión April 2018 que ha sido liberada el 18/04/2018 con una versión de kernel 4.14. Se puede descargar en la página de Raspberry Pi Foundation a través del siguiente enlace [25].

Una vez descargado y descomprimido el software el siguiente paso es formatear la tarjeta micro SD en la que será instalado a través del software SD Memory Card Formatter 4.1 de la SD Association que se encuentra [26] tanto para sistemas Windows como MAC. Es posible también realizar el formateo en otros sistemas operativos y con otras herramientas. Basta con introducir la tarjeta SD en el PC y pulsar sobre el botón Format.

El siguiente paso 4.2 es instalar la imagen de Raspbian descargada en la tarjeta micro SD que se utilizará para arrancar la Raspberry Pi. En sistemas Windows se puede hacer con el software Win 32 Disk Imager [27] en el que simplemente hay que indicarle la imagen a cargar y la unidad de la tarjeta y pulsar sobre el botón Write.

De manera similar puede hacerse con Etcher, software de código abierto



Figura 4.1: SD Memory Card Formatter



Figura 4.2: Win 32 Disk Imager

desarrollado por resin.io para flashear imágenes de sistemas operativos en tarjetas SD o dispositivos USB. Tiene la ventaja de estar disponible para sistemas Windows, Linux y Mac.

A continuación basta con introducir la tarjeta micro SD en la ranura para tal efecto en la Raspberry Pi y proceder a su arranque 4.3. El usuario por defecto es `pi` y la contraseña `raspberry`. El primer paso deberá de ser modificar la contraseña por defecto por una segura haciendo uso del comando `passwd`.

4.2.2. Configuración de la electrónica de red

De acuerdo a lo visto en el capítulo 3, para la configuración de la electrónica de red se creará una segunda VLAN en el switch para posteriormente configurar el port mirror sobre las interfaces a usar.

Para ello, tras acceder a la interfaz de configuración del switch elegido, figura 4.4, dentro del apartado de configuración VLAN se crea una adicional, en este caso se ha configurado la 69, figura 4.5, y se configuran los tres puertos que se usarán en la conexión WAN del router dentro de dicha VLAN, figura 4.6.

El último paso es configurar el port mirroring, figura 4.7, para que todo

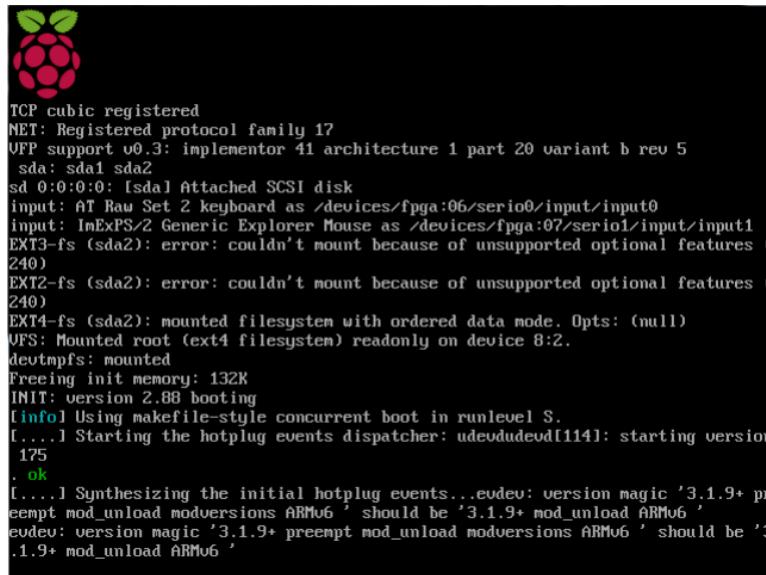


Figura 4.3: Pantalla de inicio de Raspbian

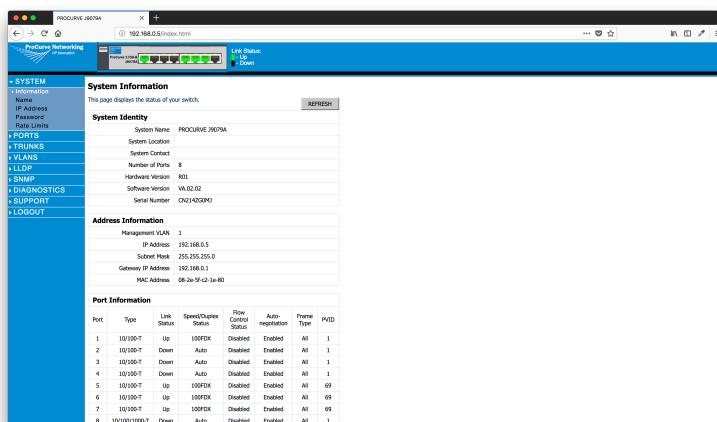


Figura 4.4: Menú de inicio de configuración del switch HP

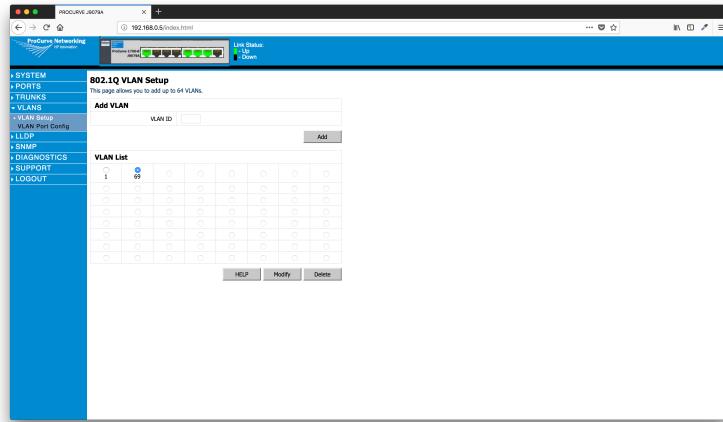


Figura 4.5: Menú de configuración de VLAN en el switch HP

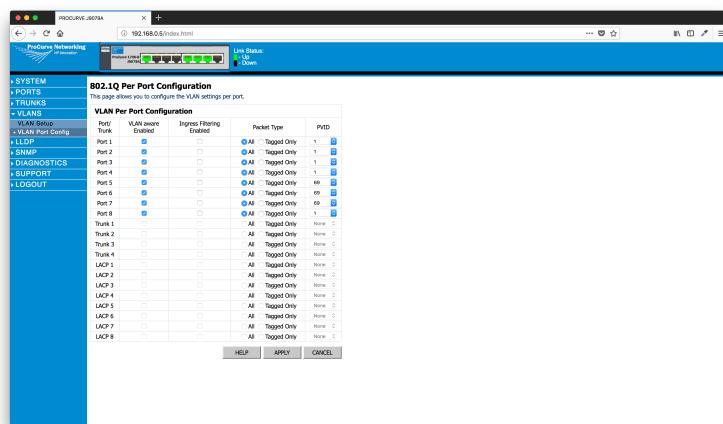


Figura 4.6: Menú de asignación de puertos por VLAN en el switch HP

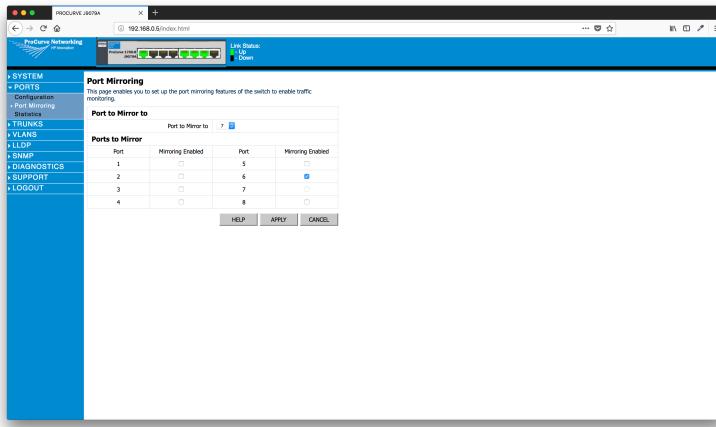


Figura 4.7: Menú de configuración de port mirroring en el switch HP

lo que se envíe o reciba por el puerto 6 que es el que proporciona la conexión a internet se copie sobre el puerto 7 que es el que se conectará con la Raspberry. El puerto 5 será el que esté conectado en la interfaz WAN del router. De esta forma ha sido posible reflejar el tráfico que circula por la conexión entre el router y el ISP y enviarlo a la Raspberry Pi para su análisis.

4.2.3. Configuración inicial del sistema operativo

En este punto se dispone de un sistema completo Raspbian GNU/Linux 9.4 (stretch) por lo que la administración y configuración del mismo no dista mucho de cualquier sistema Linux basado en Debian. De manera adicional, esta distribución incorpora el comando `raspi-config`, ver figura 4.8, que debe de ser ejecutado a través de `sudo` para elevar privilegios y que permite configurar el sistema de una forma más amena. Los detalles de configuración carecen de mayor interés ya que el objetivo principal de este trabajo no es la configuración exhaustiva de este sistema sino su uso como una sonda IDS. De todas formas de manera resumida podemos indicar que a través del menú 2 Network Options permite la configuración WiFi del sistema para dotarlo de conectividad internet, en el menú 4 Localisation Options permite configurar la codificación empleada en el sistema, se recomienda utilizar `en_US.UTF-8`, así como configurar la zona horaria y la distribución del teclado. Siguiendo con el repaso en el menú 5 Interfacing Options es posible iniciar el servidor SSH para poder tener acceso remoto al sistema a través de la red. Debido al uso intensivo y a las necesidades del almacenamiento que se va a realizar una vez el sistema esté puesto en marcha es necesario expandir el sistema de ficheros de tal forma que se emplee la totalidad de almacenamiento de la tarjeta SD. Para ello basta con seleccionar el sub menú A1 dentro del menú

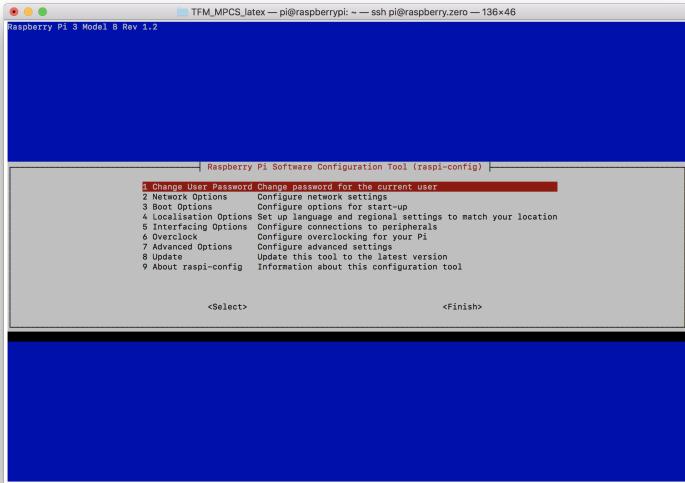


Figura 4.8: Pantalla de configuración de Raspbian

7 Advanced Options.

Una vez llegado hasta aquí es recomendable dentro de la misma utilidad ejecutar la opción de actualización de la herramienta. Una vez que finalice es posible salirse a la línea de comandos navegando hacia el menú Exit. Una vez ahí el siguiente paso es actualizar el sistema a través de la ejecución de `sudo apt-get update`, `sudo apt-get upgrade` y `sudo apt-get dist-upgrade`.

Ahora que se dispone de un sistema recién instalado y actualizado la siguiente tarea es la configuración de la red privada virtual para tener acceso al sistema desde cualquier parte con conexión a Internet y a través de la cual también se realizará el envío de los logs al servidor Elasticsearch. Para este caso se ha decidido por el uso de una red privada virtual a nivel de capa de aplicación que permite desplegar una red LAN virtual en base a redes definidas por software en la que todos los dispositivos enrolados en ella se pueden ver entre sí. Al ser una aplicación no es necesario la apertura de puertos ni realizar ninguna configuración específica en la red para permitir los flujos de tráfico necesarios. Dicho software es ZeroTier [28]. El script recogido en 4.1 muestra un ejemplo de instalación de este software y de la conexión a la red la cual es identificada a través de un ID único. En 4.9 se muestra una captura del portal de configuración de esta red.

```
curl -s https://install.zerotier.com/ | sudo bash
sudo zerotier-cli join ID_Network
```

Script 4.1: Script instalación ZeroTier

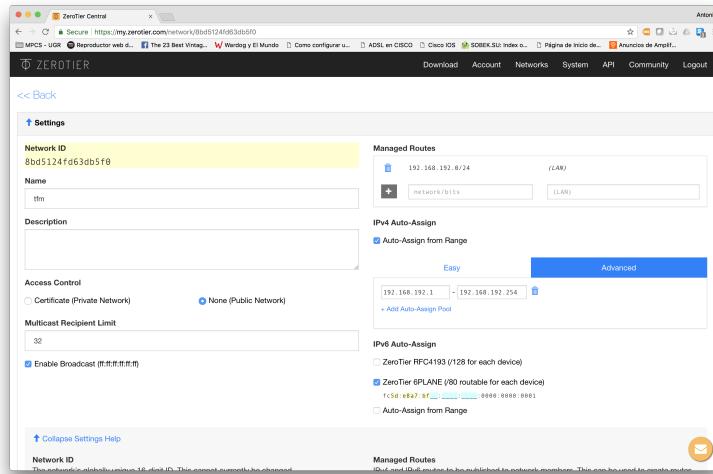


Figura 4.9: Pantalla de configuración de ZeroTier

Una vez conectado a la red aparece un nuevo adaptador de red virtual en la Raspberry sobre el que se le asigna una dirección por DHCP en el segmento de red configurado en el portal de ZeroTier tal y como se muestra en la figura 4.10.

4.2.4. Instalación de las herramientas de seguridad

En este apartado se pretende dar un repaso completo de todas las herramientas y funcionalidades que se han implementado en el sistema, las indicaciones de uso y cómo se relacionan entre ellas. Como base se ha utilizado el proyecto **foxhound-nsm** realizado por *sneakymonk3y* que puede encontrarse en GitHub [29] y en particular la petición de push realizada por *gebhard73*. También se añaden los scripts necesarios que se han realizado para la instalación.

Instalación de base de datos GeoLite

El primer paso, script 4.2, es añadir la base de datos de GeoLite para poder posicionar las direcciones IP que se detecten en el sistema. Para ello basta con ejecutar con permisos de **sudo** el siguiente script.

```
#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"
#export DEBIAN_FRONTEND=noninteractive
if [ "$EUID" -ne 0 ]
```

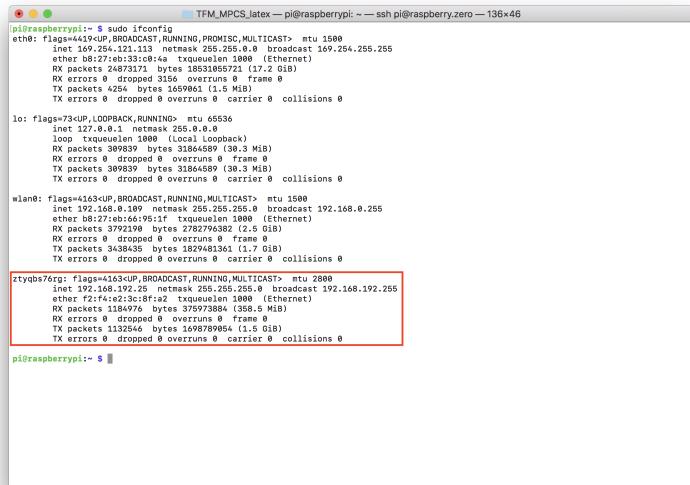


Figura 4.10: Visualización de la interfaz virtual creada con ZeroTier

```

then echo "Please run as root"
exit 1
fi

function Info {
echo -e -n '\e[7m'
echo "$@"
echo -e -n '\e[0m'
}

function Error {
echo -e -n '\e[41m'
echo "$@"
echo -e -n '\e[0m'
}

function install_geoip()
{
Info "Installing GEO-IP"
wget http://geolite.maxmind.com/download/geoip/database/
GeoLiteCity.dat.gz
wget http://geolite.maxmind.com/download/geoip/database/
GeoLiteCityv6-beta/GeoLiteCityv6.dat.gz
gunzip GeoLiteCity.dat.gz
gunzip GeoLiteCityv6.dat.gz
mv GeoLiteCity* /usr/share/GeoIP/
ln -s /usr/share/GeoIP/GeoLiteCity.dat /usr/share/GeoIP/
GeoIPCity.dat
ln -s /usr/share/GeoIP/GeoLiteCityv6.dat /usr/share/GeoIP/
GeoIPCityv6.dat
}

install_geoip

```

Script 4.2: Script instalación GeoIP

Instalación de paquetes y creación de carpetas

El segundo paso, script 4.3, es la creación de toda la estructura de carpetas donde almacenar toda la información que el sistema generará y la instalación de los paquetes requeridos.

```
#!/usr/bin/env bash
_scriptDir="$(dirname $(readlink -f $0))"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
then echo "Please run as root"
exit 1
fi

function Info {
echo -e -n '\e[7m'
echo "$@"
echo -e -n '\e[0m'
}

function Error {
echo -e -n '\e[41m'
echo "$@"
echo -e -n '\e[0m'
}

Info "Creating directories"
mkdir -p /nsm
mkdir -p /nsm/pcap/
mkdir -p /nsm/scripts/
mkdir -p /nsm/bro/
mkdir -p /nsm/bro/logs
mkdir -p /nsm/bro/extracted/
if [ ! -d /opt/ ]; then
mkdir -p /opt/
fi
ln -s /nsm/bro/logs /var/log/bro

function install_packages()
{
Info "Installing Required .debs"
apt-get update && apt-get -y install cmake make gcc g++ flex bison
libpcap-dev libssl-dev python-dev swig zlib1g-dev ssmtp htop vim
libgeoip-dev ethtool git tshark tcpdump nmap mailutils python-pip
autoconf libtool pkg-config libnacl-dev libncurses5-dev libnet1-
dev libcli-dev libnetfilter-conntrack-dev liburcu-dev

if [ $? -ne 0 ]; then
Error "Error. Please check that apt-get can install
needed packages."

```

```

        exit 2;
    fi
Info "Required -debs installed"
}

install_packages

```

Script 4.3: Script creación de carpetas e instalación de .debs

Configuración de la interfaz eth0

A continuación se procede a la configuración de la interfaz ethernet que será la que esté conectada al puerto del switch que estará funcionando en modo port mirror a través de la cual se realizará la captura de paquetes. De forma resumida se deshabilita IPv6, se configura la interfaz en modo promiscuo y se configura una MTU de 9000 bytes a través de un script auxiliar llamado *nic.sh* que se invoca cada vez que se levanta la interfaz, y se fuerza a que no tenga dirección IP a través de la asignación de la IP 0.0.0.0 para que si hay algún DHCP en la red ignore los paquetes.

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
    then echo "Please run as root"
    exit 1
fi

function Info {
    echo -e -n '\e[7m'
    echo "$@"
    echo -e -n '\e[0m'
}

function Error {
    echo -e -n '\e[41m'
    echo "$@"
    echo -e -n '\e[0m'
}

function config_net_ipv6()
{
Info "Disabling IPv6"
    if [ `grep 'net.ipv6.conf.all.disable_ipv6 = 1' /etc/sysctl.conf | wc -l` -eq 0 ] ; then
        echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf
    fi
    if [ `grep 'ipv6.disable_ipv6=1' /boot/cmdline.txt | wc -l` -eq 0 ] ; then
        sed -i '1 s/$/ ipv6.disable_ipv6=1/' /boot/cmdline.txt
    fi
Info "Sysctl"
}

```

```

        sysctl -p
}

config_net_ipv6

```

Script 4.4: Deshabilitar IPv6

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
then echo "Please run as root"
exit 1
fi

function Info {
echo -e -n '\e[7m'
echo "$@"
echo -e -n '\e[0m'
}

function Error {
echo -e -n '\e[41m'
echo "$@"
echo -e -n '\e[0m'
}

function config_net_opts()
{
Info "Configuring network options"
cd $_scriptDir
cp nic.sh /etc/network/if-up.d/interface-tuneup
chmod +x /etc/network/if-up.d/interface-tuneup
ifconfig eth0 down && ifconfig eth0 up
}

config_net_opts

```

Script 4.5: Configuración de la interfaz en modo promiscuo

```

#!/bin/bash
for i in rx tx gso gro; do ethtool -K eth0 $i off; done;
ifconfig eth0 promisc
ifconfig eth0 mtu 9000
exit 0

```

Script 4.6: Auxiliar nic.sh

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
then echo "Please run as root"

```

```

    exit 1
fi

function Info {
  echo -e -n '\e[7m'
  echo "$@"
  echo -e -n '\e[0m'
}

function Error {
  echo -e -n '\e[41m'
  echo "$@"
  echo -e -n '\e[0m'
}

function config_eth0()
{
Info "Configuring eth0"
cat >> /etc/dhcpcd.conf <<EOF

# IDSPi for promiscuous mode
static
interface eth0
static ip_address=0.0.0.0
EOF
}

config_eth0

```

Script 4.7: Deshabilita la configuración por DHCP a través de eth0

Instalación de NetSniff y creación del servicio asociado

Para poder adquirir muestras del tráfico que circula por la red se instala en el sistema el paquete NetSniff para capturar el tráfico y almacenarlo dentro de la ruta `/nsm/pcap/` en archivos de 100 MB. Tras instalar NetSniff-nf se crea un servicio para que se inicie en el arranque y se copia un script adicional llamado `cleanup.sh` que comprueba que el tamaño de los archivos ubicados en las rutas `/nsm/pcap` y `/nsm/bro/extracted` no supera el tamaño configurado, por defecto 25 GB para los pcap y 2.5 GB para los archivos extraídos eliminando los más antiguos.

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit 1
fi

function Info {
  echo -e -n '\e[7m'
  echo "$@"
  echo -e -n '\e[0m'
}
```

```

    echo -e -n '\e[0m'
}

function Error {
    echo -e -n '\e[41m'
    echo "$@"
    echo -e -n '\e[0m'
}

function install_netsniff()
{
Info "Installing Netsniff-NG PCAP"
    touch /etc/netsniff
    apt-get -y install netsniff-ng
    cd $_scriptDir
    cp cleanup.sh /nsm/scripts/cleanup
    chmod +x /nsm/scripts/cleanup
}

function create_service_netsniff()
{
Info "Creating Netsniff-NG service"
echo "[Unit]
Description=Netsniff-NG PCAP
After=network.target

[Service]
ExecStart=/usr/sbin/netsniff-ng --in eth0 --out /nsm/pcap/ --bind-cpu
    3 -s --interval 100MiB
Type=simple
EnvironmentFile=/etc/netsniff

[Install]
WantedBy=multi-user.target" > /etc/systemd/system/netsniff-ng.service
    systemctl enable netsniff-ng
    systemctl daemon-reload
    service netsniff-ng start
}

install_netsniff
create_service_netsniff

```

Script 4.8: Instalar y configurar servicio Netsniff

```

#!/bin/sh

TopSizePCAP=25000000
TopSizeExtract=2500000

removePcap() {
    local usedPCAP
    usedPCAP='du /nsm/pcap/ | awk '{ print $1 }''"
    while [ ${usedPCAP} -gt ${TopSizePCAP} ]; do
        find /nsm/pcap -type f -printf '%T+ %p\n' | sort | head -n1 | awk
            '{print $2}' | xargs rm -v
        usedPCAP='du /nsm/pcap/ | awk '{ print $1 }''"
    done
}

```

```

removeExtracted() {
    local usedExtracted
    usedExtracted='du /nsm/bro/extracted/ | awk '{ print $1 }','
    while [ ${usedExtracted} -gt ${TopSizeExtract} ]; do
        find /nsm/bro/extracted -type f -printf '%T+ %p\n' | sort | head
        -n1 | awk '{print $2}' | xargs rm -v
        usedExtracted='du /nsm/bro/extracted/ | awk '{ print $1 }','
    done
}

removePcap
removeExtracted

exit 0

```

Script 4.9: Borrado de .pcaps

Sincronización horaria

Uno de los puntos más importantes dentro de cualquier despliegue, especialmente en aquellos que integren datos de múltiples fuentes distribuidas, es que estén sincronizadas en el tiempo entre ellas para poder realizar una buena correlación de eventos. Para ello se instala un cliente NTP y se configura como servidor horario hora.rediris.es.

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
    then echo "Please run as root"
    exit 1
fi

function Info {
    echo -e -n '\e[7m'
    echo "$@"
    echo -e -n '\e[0m'
}

function Error {
    echo -e -n '\e[41m'
    echo "$@"
    echo -e -n '\e[0m'
}

echo "Please enter your ntp server (leave blank for defaults)"
read ntp_server

# ntp_server = "hora.rediris.es"

function config_ntp()
{
if [ "${ntp_server}" == "" ]; then
    Info "No ntp server set, skipping."

```

```

else
    Info "Configuring NTP"
    sed -i.bak 's/^\pool /# pool /' /etc/ntp.conf
    sed -i 's/^server /# server /' /etc/ntp.conf
    echo "## added by IDSpi:" >> /etc/ntp.conf
    echo "server $ntp_server" >> /etc/ntp.conf
fi
}

config_ntp

```

Script 4.10: Sincronización horaria NTP

Instalación de herramientas IDS

Como motor IDS sobre el que desarrollar el resto de funcionalidades se instalará Bro junto con sus herramientas complementarias bro-aux, bro-common y su herramienta de gestión broctl.

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
    then echo "Please run as root"
    exit 1
fi

function Info {
    echo -e -n '\e[7m'
    echo "$@"
    echo -e -n '\e[0m'
}

function Error {
    echo -e -n '\e[41m'
    echo "$@"
    echo -e -n '\e[0m'
}

function install_bro()
{
Info "Installing Bro"
    apt-get -y install bro broctl bro-common bro-aux
}

install_bro

```

Script 4.11: Instalación de Bro

Análisis de malware con Loki, IOCs y reglas YARA

Como se ha visto en el capítulo 3, el sistema almacenará los archivos que encuentre dentro de los flujos de tráfico y procederá a analizarlos en busca de muestras de malware. Para este análisis se utilizará Loki. Los logs con los resultados obtenidos se almacenarán en `/nsm/Loki`.

```
#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
then echo "Please run as root"
exit 1
fi

function Info {
echo -e -n '\e[7m'
echo "$@"
echo -e -n '\e[0m'
}

function Error {
echo -e -n '\e[41m'
echo "$@"
echo -e -n '\e[0m'
}

function install_loki()
{
    Info "Installing YARA packages"
    Info "Installing PIP LOKI Packages"
        pip install psutil
        pip install yara-python
        pip install gitpython
        pip install pylzma
        pip install netaddr
    Info "Installing LOKI"
        git clone https://github.com/Neo23x0/Loki.git /nsm/
        Loki
        git clone https://github.com/Neo23x0/signature-base/
        git /nsm/Loki/signature-base/
        echo "export PATH=/nsm/Loki:$PATH" >> /etc/profile
        chmod +x /nsm/Loki/loki.py
        echo "export PYTHONPATH=$PYTHONPATH:/nsm/Loki" >> /etc
        /profile
    echo "
#!/bin/sh
/usr/bin/python /nsm/Loki/loki.py --noprocscan --dontwait --
    onlyrelevant -p /nsm/bro/extracted -l /nsm/Loki/log
" \ > /nsm/scripts/scan
chmod +x /nsm/scripts/scan
}

install_loki
```

Script 4.12: Instalación de Loki

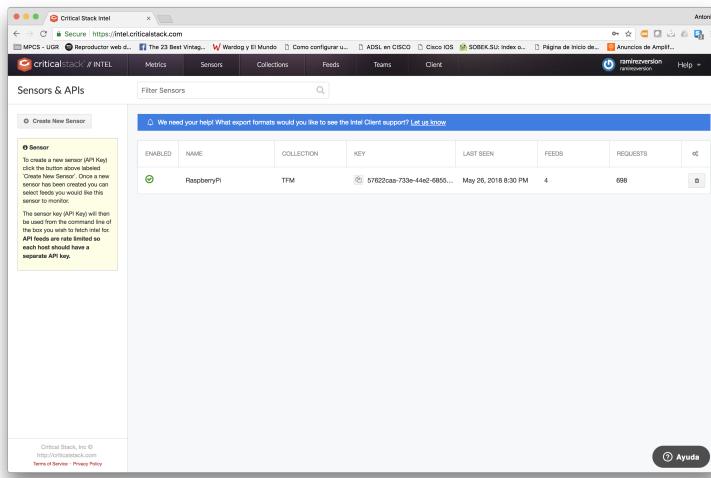


Figura 4.11: Creación de un sensor en Critical Stack

Motor de firmas malware Intel Critical Stack

El primer paso para hacer uso de los feed de malware de Intel Critical Stack [30] hay que registrarse en el propio portal de gestión y una vez dentro dar de alta un nuevo sensor como se observa en la figura 4.11. Una vez creado, dentro del mismo aparece la clave de la Api que posteriormente se usará en la instalación del agente. También es posible configurar los feeds de información relativa a malware y dominios maliciosos en el menú Collections tal y como se observa en la figura 4.12. Con estos pasos completados ya es posible proceder a la instalación del la aplicación en la placa de desarrollo.

El script 4.13 muestra el proceso de instalación de esta herramienta y cómo adicionalmente se crea un script llamado `update` que almacenado en `/nsm/scripts/` permite actualizar los feeds acerca de dominios maliciosos de Critical Stack así como las reglas YARA que utiliza Loki para su análisis.

```
#!/usr/bin/env bash
_scriptDir="$(dirname $(readlink -f $0))"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit 1
fi

function Info {
  echo -e -n '\e[7m'
  echo "$@"
  echo -e -n '\e[0m'
}
```

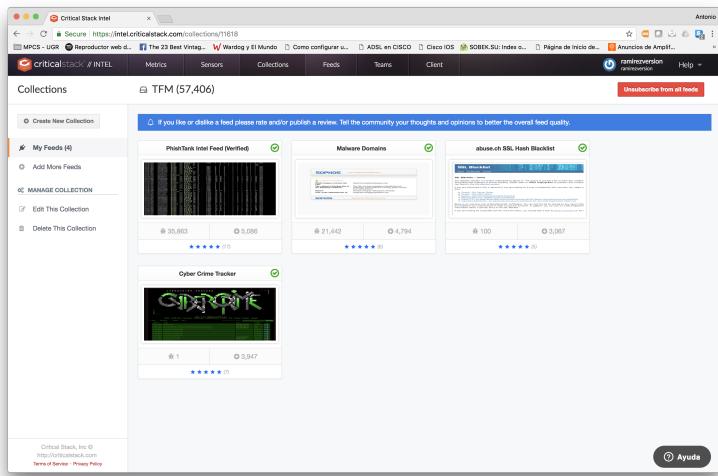


Figura 4.12: Suscripción a feeds en Critical Stack

```

function Error {
    echo -e -n '\e[41m'
    echo "$@"
    echo -e -n '\e[0m'
}

echo "Please enter your Critical Stack API Key (sensor): "
read api

function install_criticalstack()
{
Info "Installing Critical Stack Agent"
    wget --no-check-certificate https://intel.criticalstack.com/
          client/critical-stack-intel-arm.deb
    dpkg -i critical-stack-intel-arm.deb
        chown critical-stack:critical-stack /usr/share/bro/
          site/local.bro
    sudo -u critical-stack critical-stack-intel config --
          set bro.path=/usr/bin/bro
    sudo -u critical-stack critical-stack-intel config --
          set bro.include.path=/usr/share/bro/site/local.bro
    sudo -u critical-stack critical-stack-intel config --
          set bro.broctl.path=/usr/bin/broctl
    sudo -u critical-stack critical-stack-intel api $api
    sudo -u critical-stack critical-stack-intel list
    sudo -u critical-stack critical-stack-intel pull
    #Deploy and start BroIDS
    export PATH="/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/
      bin:/usr/local/bro/bin:$PATH"
    echo "Deploying and starting BroIDS"

    sudo echo "critical-stack ALL=(ALL) NOPASSWD: ALL" > /etc/sudoers.
      d/99-critical-stack
        broctl deploy
        broctl cron enable
  
```

```

#Create update script
echo "
echo \"#### Pulling feed update ####\
sudo -u critical-stack critical-stack-intel pull
echo \"#### Applying the updates to the bro config ####\
broctl check
broctl install
echo \"#### Restarting bro ####\
broctl restart
cd /nsm/Loki/
python ./loki.py --update
" \ > /nsm/scripts/update
        sudo chmod +x /nsm/scripts/update
}

install_criticalstack

```

Script 4.13: Instalación de Critical Stack

El desarrollo y soporte de Critical Stack para ARM está discontinuado por lo que es posible que con esa configuración las cadenas de texto con las políticas descargadas en la ruta `/opt/critical-stack/frameworks/intel/feeds.bro` no se carguen y que cada vez que se regenera con cada actualización se vuelve erróneo. Para solucionarlo se añaden las siguientes líneas al final del fichero de configuración de Bro `/etc/bro/site/local.bro`.

```

sudo echo "@load policy/frameworks/intel/seen
@load policy/frameworks/intel/do_notice" >> /etc/bro/site/local.bro

```

Script 4.14: Ajuste de Critical Stack para funcionar en Raspberry Pi

Tras esto es posible lanzar una actualización manual del sistema para recargar la configuración y actualizar los feeds de Critical Stack `sudo /nsm/scripts/update`.

Instalación de script auxiliares de Bro

Para completar y complementar el sistema, es posible ampliar las funcionalidad de Bro a través de scripts auxiliares. Concretamente se han seleccionado dos de ellos. El primero permite, haciendo uso de la base de datos GeoLite de posicionamiento, a través de direcciones IP, extraer el país y la ciudad en determinado casos a través de su dirección IP. El segundo proporciona las herramientas necesarias para almacenar los archivos y fichero que viajen sin cifrar por la red y guardar una copia. Se añade también una línea al final del fichero de configuración `/etc/bro/broctl.cf` para indicar que se guarden los logs en formato json.

```

#!/usr/bin/env bash
_scriptDir="$(dirname `readlink -f $0`)"

```

```

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
    then echo "Please run as root"
    exit 1
fi

function Info {
    echo -e -n '\e[7m'
    echo "$@"
    echo -e -n '\e[0m'
}

function Error {
    echo -e -n '\e[41m'
    echo "$@"
    echo -e -n '\e[0m'
}

function install_bro_reporting()
{
Info "Bro Reporting Requirements"
    pip install colorama
#PYSUBNETREE
    pip install pysubnettree
#IPSUMDUMP
    cd /opt/
    wget http://www.read.seas.harvard.edu/~kohler/ipsumpdump/
        ipsumpdump-1.85.tar.gz
    tar -zxf ipsumpdump-1.85.tar.gz
    cd ipsumpdump-1.85/
    ./configure && make && make install
}

function config_bro_scripts()
{
Info "Configuring BRO scripts"
    #PULL BRO SCRIPTS
    cd /usr/share/bro/site/
    if [ -d /usr/share/bro/site/bro-scripts/ ]; then
        rm -rf /usr/share/bro/site/bro-scripts/
    fi
    mkdir -p /usr/share/bro/site/bro-scripts
    git clone https://github.com/sneakymonk3y/bro-scripts.git
    echo "@load bro-scripts/geoip" >> /usr/share/bro/site/local.
        bro
    echo "@load bro-scripts/extract" >> /usr/share/bro/site/local.
        .bro

    #configure Bro for JSON log output
    echo "### Added for json log output" >> /etc/bro/broctl.cfg
    echo "broargs=-e 'redef LogAscii::use_json=T;'" >> /etc/bro/broctl.
        cfg

    broctl deploy
}

install_bro_reporting
config_bro_scripts

```

Script 4.15: Scripts auxiliares de Bro

Configuración de tareas recurrentes

Como último paso en la instalación se configuran una serie de tareas recurrentes a través del planificador cron del sistema. En particular se configura para que a cada múltiplo entero de 5 minutos se ejecute el comando `broctl cron` que se encarga de revisar los nodos configurados para que en caso de hayan tenido algún crash volverlos a levantar automáticamente. El script de actualización de Critical Stack y YARA se ejecuta en el minuto 30 una vez cada hora mientras que el script de limpiado de los .pcap guardados es ejecutado cada 5 minutos.

```
#!/usr/bin/env bash
_scriptDir="$(dirname $(readlink -f $0))"

#export DEBIAN_FRONTEND=noninteractive

if [ "$EUID" -ne 0 ]
  then echo "Please run as root"
  exit 1
fi

function Info {
  echo -e -n '\e[7m'
  echo "$@"
  echo -e -n '\e[0m'
}

function Error {
  echo -e -n '\e[41m'
  echo "$@"
  echo -e -n '\e[0m'
}

#CRON JOBS
echo "0,5,10,15,20,25,35,40,45,50,55 * * * * root /usr/bin/broctl cron
" >> /etc/crontab
echo "*/5 * * * * root /nsm/scripts/cleanup" >> /etc/crontab
echo "30 * * * * root /nsm/scripts/update" >> /etc/crontab
#echo "*/5 * * * * root python /nsm/scripts/scan" >> /etc/crontab
```

Script 4.16: Tareas recurrentes y configuración de cron

4.2.5. Compilación y configuración de Filebeat y Metricbeat

Para poder hacer el envío al servidor ELK tantos de los logs del sistema (Syslog, Auth, Apache2) y de los logs generados por Bro se hace uso de Filebeat. De igual forma es posible recoger los datos de la monitorización del estado de la sonda (CPU, procesos, memoria) al servidor central de Elasticsearch y Kibana a través de Metricbeat. En este punto surge un problema puesto que las últimas versiones del software a utilizar no son compatibles de manera nativa sobre arquitecturas ARM por lo que hay que

realizar una compilación cruzada. Los detalles se encuentran en la sección 4.2.5.

Compilación Filebeat/Metricbeat

Para la compilación de este software se han seguido las referencias que pueden encontrarse en [31] y [32].

El primer paso es instalar el intérprete de Go o en su defecto usar el docker. Ésta última ha sido la opción elegida y en concreto se ha realizado con el siguiente contenedor.

```
sudo docker run -it --rm -v `pwd`:/build golang:latest /bin/bash
```

Script 4.17: Arranque del docker golang

Una vez dentro del docker se ha utilizado el script 4.18 para compilar Filebeat. No se incluye el código pero de manera similar se realizan los mismos pasos para la compilación de Metricbeats.

```
# -----
# Step 1) Build
# NOTE: the git checkout version needs to match the elastic search API
#       version
# -----
elastic_version="6.2.4"

echo "-----"
echo " $(date)"
echo " Downloading source..."
echo "-----"
go get github.com/elastic/beats
cd /go/src/github.com/elastic/beats/filebeat/
git checkout "v${elastic_version}"
echo "-----"
echo " $(date)"
echo " Building source..."
echo "-----"
GOARCH=arm go build
cp filebeat /build
cd /build

# -----
# Step 2) Download the tar
# The url contains the version number like this: "https://artifacts.
#       elastic.co/downloads/beats/filebeat/filebeat-6.1.1-linux-x86.tar.
#       gz"
# -----


# then download the linux tar from:
download_url="https://artifacts.elastic.co/downloads/beats/filebeat/
#       filebeat-${elastic_version}-linux-x86.tar.gz"
echo "-----"
echo " $(date)"
echo " Downloading filebeat tarball from:"
```

```
echo " ${download_url}"
echo "-----"
curl $download_url -o download.tar

# -----
# Step 3) Untar, modify, tar
# Drop the filebeat binary into the new tar....
# -----
echo "-----"
echo " $(date)"
echo " Adding the filebeat binary to the tar..."
echo "-----"
mkdir workdir
tar -xf download.tar -C workdir --strip-components=1
cp filebeat workdir/filebeat
cd workdir
tar -zcf ../../pibeats-${elastic_version}.tar.gz .
cd ..
echo "-----"
echo " $(date)"
echo " Clearning up..."
echo "-----"
rm -rf filebeat
rm -rf workdir
rm -rf download.tar

echo "-----"
echo " $(date)"
echo " COMPLETE! Copy pibeats.tar.gz to raspberry pi!"
echo ""
echo " Something like this?"
echo " scp ./pibeats.tar.gz username@pi_address:/home/username/"
echo "-----"
```

Script 4.18: Compilación de Filebeat

En este punto ya es posible descomprimir los dos archivos con extensión .tar.gz y copiarlos a la ruta donde se van a ejecutar. Durante esta instalación se han ubicado dentro de /opt/filebeat y /opt/metricbeat. La configuración de toda la suite de elasticsearch se realiza en base a ficheros de texto con extensión .yml.

Configuración de Filebeat

Para el envío de los logs del sistema con Filebeat es posible usar los módulos System y Apache2 que ya tiene integrados. Para ello, los logs tienen que ser enviados directamente al servidor Elasticsearch, sin poder pasar por Logstash antes, pero para poder añadir geo localización a las direcciones IP recogidas por Bro, los logs tienen que ser tratados por el servidor Logstash previamente. Sin embargo, Filebeat solo permite la configuración de un destino de los logs por lo que para poder hacer uso de ambas funcionalidades se configuran dos instancias de Filebeat cada una con su configuración correspondiente para enviar los logs a Elasticsearch o Logstash según co-

rresponda. Para la configuración de Filebeat basta con copiar el archivo de configuración en la ruta deseada (por comodidad se han situado en la misma carpeta del ejecutable).

Para que Filebeat funcione es necesario que el propietario de todos los archivos sea root por lo que se ejecuta el siguiente comando para lograrlo
`sudo chown -R root:root /opt/filebeat/*.`

Para el envío de los logs del sistema, la configuración de los módulos de Filebeat se encuentra en la ruta `/opt/filebeat/modules.d`. En los siguientes scripts (4.19 y 4.20) se recoge la configuración para los módulos `system` `/opt/filebeat/modules.d/system.yml` y `apache2` `/opt/filebeat/modules.d/apache2.yml` de filebeat.

```
- module: apache2
  # Access logs
  access:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/apache2/access*"]

  # Error logs
  error:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/apache2/error*"]
```

Script 4.19: Configuración del módulo `system` de Filebeat

```
- module: apache2
  # Access logs
  access:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/apache2/access*"]

  # Error logs
  error:
    enabled: true
    # Set custom paths for the log files. If left empty,
    # Filebeat will choose the paths depending on your OS.
    var.paths: ["/var/log/apache2/error*"]
```

Script 4.20: Configuración del módulo `apache2` de Filebeat

El fichero de configuración 4.21 es el utilizado para la instancia que recoge los logs del sistema y del servidor Apache. Se recogen todos los logs del sistema y se configuran y habilitan los módulos `system` y `apache2`. Finalmente se hace el envío al servidor de Elasticsearch.

```
#= Filebeat prospectors =
filebeat.prospectors:
- type: log

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

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/boot*
    - /var/log/daemon*
    - /var/log/debug*
    - /var/log/dpkg*
    - /var/log/faillog*
    - /var/log/kern*
    - /var/log/lastlog*
    - /var/log/messages*
    - /var/log/user*
    - /var/log/Xorg*

  exclude_files: [ '\.gz$' ]

#= Filebeat modules =
filebeat.config.modules:
  # Glob pattern for configuration loading
  enable: true
  path: /opt/filebeat2/modules.d/*.yml

# Kibana =
# Starting with Beats version 6.0.0, the dashboards are loaded via the
# Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  host: "kibana.zero:80"

# Outputs =
# Configure what outputs to use when sending the data collected by the
# beat
#-- Elasticsearch output -
output.elasticsearch:
  hosts: "kibana.zero:9200"
```

Script 4.21: Configuración de Filebeat (Sistema y Apache2)

Para la instancia que se encarga de recoger los logs de Bro, la configuración se muestra en el script 4.22. Para cada archivo de log se añade un valor en el campo tag para facilitar su búsqueda en Kibana, se añaden: un campo Bro con el valor del tipo de log y dos campos para que posteriormente Logstash rellene los valores de geolocalización y se configura para indicarle que son logs en formato JSON. En el envío de los logs a Logstash se configura el índice a configurar para posteriormente crearlo en Kibana.

```
#= Filebeat prospectors =
```

```
filebeat.prospectors:

# Bro current logs

- type: log
  enabled: true
  paths: ["/var/log/bro/current/conn.log"]
  tags: ["bro-conn"]
  fields:
    bro: conn
    geobro_resp:
    geobro_orig:
    fields_under_root: true
    json.keys_under_root: true
    json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/dns.log"]
  tags: ["bro-dns"]
  fields:
    bro: dns
    geobro_resp:
    geobro_orig:
    fields_under_root: true
    json.keys_under_root: true
    json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/files.log"]
  tags: ["bro-files"]
  fields:
    bro: files
    geobro_resp:
    geobro_orig:
    fields_under_root: true
    json.keys_under_root: true
    json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/http.log"]
  tags: ["bro-http"]
  fields:
    bro: http
    geobro_resp:
    geobro_orig:
    fields_under_root: true
    json.keys_under_root: true
    json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/notice.log"]
  tags: ["bro-notice"]
  fields:
    bro: notice
    geobro_resp:
    geobro_orig:
    fields_under_root: true
    json.keys_under_root: true
```

```
  json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/ssl.log"]
  tags: ["bro-ssl"]
  fields:
    bro: ssl
    geobro_resp:
    geobro_orig:
  fields_under_root: true
  json.keys_under_root: true
  json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/stats.log"]
  tags: ["bro-stats"]
  fields:
    bro: stats
    geobro_resp:
    geobro_orig:
  fields_under_root: true
  json.keys_under_root: true
  json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/software.log"]
  tags: ["bro-software"]
  fields:
    bro: software
    geobro_resp:
    geobro_orig:
  fields_under_root: true
  json.keys_under_root: true
  json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/x509.log"]
  tags: ["bro-x509"]
  fields:
    bro: x509
    geobro_resp:
    geobro_orig:
  fields_under_root: true
  json.keys_under_root: true
  json.message_key: message

- type: log
  enabled: true
  paths: ["/var/log/bro/current/intel.log"]
  tags: ["bro-intel"]
  fields:
    bro: intel
    geobro_resp:
    geobro_orig:
  fields_under_root: true
  json.keys_under_root: true
  json.message_key: message
```

```

#= Outputs =
#- Elasticsearch output -
output.logstash:
  hosts: "kibana.zero:5044"
  index: "brobeat-%{+yyyy.MM.dd}"

```

Script 4.22: Configuración de Filebeat (Bro)

Configuración de Metricbeat

Para la configuración de Metricbeat se sigue un proceso similar al descrito en la sección 4.2.5.

En primer lugar se cambia el propietario de los archivos de Metricbeat

```
sudo chown -R root:root /opt/metricbeat/*.
```

En este caso se ha optado solo por el envío de los datos de monitorización del sistema. El fichero de configuración `/opt/metricbeat/metricbeat.yml` se indica en el script 4.23.

```

#= Modules configuration =
metricbeat.config.modules:
  # Glob pattern for configuration loading
  path: /opt/metricbeat/modules.d/*.yml
  # Set to true to enable config reloading
  reload.enabled: false
  # Period on which files under path should be checked for changes
  #reload.period: 10s

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

#= Kibana =
# Starting with Beats version 6.0.0, the dashboards are loaded via the
# Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:
  # Kibana Host
  host: "kibana.zero:80"

# Configure what output to use when sending the data collected by the
# beat.
#- Elasticsearch output -
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["kibana.zero:9200"]

```

Script 4.23: Configuración de Metricbeat

La configuración del módulo `system` para metricbeat 4.24 debe estar en la ruta `/opt/metricbeat/modules.d/system.yml`.

```

- module: system
  period: 10s
  metricsets:
    - cpu
    - load
    - memory
    - network
    - process
    - process_summary
    #- core
    #- diskio
    #- socket
  processes: ['.*']
  process.include_top_n:
    by_cpu: 5      # include top 5 processes by CPU
    by_memory: 5   # include top 5 processes by memory

- module: system
  period: 1m
  metricsets:
    - filesystem
    - fsstat
  processors:
  - drop_event.when.regexp:
      system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib)(\$|/)' 

- module: system
  period: 15m
  metricsets:
    - uptime

```

Script 4.24: Configuración del módulo system de Metricbeat

Instalación de visualizaciones y dashboards pre configuradas en Kibana

Dentro del paquete que se ha compilado de Filebeat y Metricbeat hay incluidas visualizaciones y dashboards por defecto con los indicadores más útiles ya pre configuradas para la monitorización de apache y de sistemas Linux. Para instalarlas es necesario que el servidor Kibana esté configurado y en ejecución. El proceso para configurarlo se muestra en la sección 4.3.1. Una vez instalado y arrancado basta con ejecutar los siguientes comandos teniendo en cuenta encontrarse dentro de las rutas correspondientes de los ejecutables de filebeat y metricbeat. Esto sólo es válido para la instancia de Filebeat que recoge los logs del sistema.

```

cd /opt/filebeat/
sudo ./filebeat -c filebeat.yml setup -e
cd /opt/metricbeat/
sudo ./metricbeat -c metricbeat.yml setup -e

```

Script 4.25: Instalación en Kibana de visualizaciones y dashboards

Con la opción `-c` se indica el fichero de configuración a cargar, la opción `setup` se procede a instalar las visualizaciones y dashboards en el servidor Kibana configurado en el fichero de configuración y la opción `-e` imprime en pantalla los mensajes durante el proceso, útil para revisar el proceso por si se producen errores.

Configuración de los servicios Filebeat y Metricbeat

Para iniciar ambas instancias de Filebeat y la de Metricbeat de forma sencilla e incorporarlo para que arranque automáticamente durante el arranque del sistema se configuran dos servicios.

```
#!/usr/bin/env bash

#Create service config for filebeat-system
echo "[Unit]"
Description=filebeat-system
Documentation=https://www.elastic.co/guide/en/beats/filebeat/current/
    index.html
Wants=userwork-online.target
After=network-online.target

[Service]
ExecStart=/opt/filebeat2/filebeat -c /opt/filebeat2/filebeat.yml
Restart=always

[Install]
WantedBy=multi-user.target
" > /usr/lib/systemd/system/filebeat-system.service

#Create service config for filebeat-bro
echo "[Unit]"
Description=filebeat-bro
Documentation=https://www.elastic.co/guide/en/beats/filebeat/current/
    index.html
Wants=userwork-online.target
After=network-online.target

[Service]
ExecStart=/opt/filebeat/filebeat -c /opt/filebeat/filebeat.yml
Restart=always

[Install]
WantedBy=multi-user.target
" > /usr/lib/systemd/system/filebeat-bro.service

#Create service config for metricbeat
echo "[Unit]"
Description=metricbeat
Documentation=https://www.elastic.co/guide/en/beats/metricbeat/current/
    /index.html
Wants=userwork-online.target
After=network-online.target

[Service]
ExecStart=/opt/metricbeat/metricbeat -c /opt/metricbeat/metricbeat.yml
Restart=always
```

```
[Install]
WantedBy=multi-user.target
" > /usr/lib/systemd/system/metricbeat.service

#Run services during startup
systemctl enable filebeat-system
systemctl enable filebeat-bro
systemctl enable metricbeat

#Launch services
systemctl start filebeat-system
systemctl start filebeat-bro
systemctl start metricbeat
```

Script 4.26: Configuración de los servicios asociados a Filebeat y Metricbeat

4.2.6. Instalación de la interfaz gráfica web

A través de los siguientes puntos se especifican los pasos y el procedimiento seguido para la instalación de los diferentes sistemas y software sobre los que se sustenta la interfaz gráfica web. Dicha interfaz ha sido desarrollada sobre los siguientes componentes.

- Laravel versión 5.5.40
- PHP versión 7.0.27-0+deb9u1
- Apache 2 versión 2.4.25
- MySql versión 10.1.23-MariaDB-9+deb9u1

Instalación de los componentes

En el script 4.27 se muestra el proceso de instalación del software necesario para dar soporte a la interfaz web.

```
#!/usr/bin/env bash

sudo apt-get update
sudo apt-get upgrade

sudo apt-get install apache2
sudo apt-get install mysql-server

sudo apt-get install php7.0 php7.0-mcrypt php7.0-xml php7.0-gd php7.0-
    opcache php7.0-mbstring php7.0-mysql libapache2-mod-php7.0
    phpmyadmin php7.0-mbstring php-gettext

sudo phpenmod mcrypt
sudo phpenmod mbstring
```

```
sudo a2enmod ssl
sudo a2enmod rewrite

sudo systemctl restart apache2
```

Script 4.27: Instalación de los componentes necesarios para la interfaz web

Configuración de los componentes para la interfaz web

En primer lugar se crea un usuario para phpmyadmin para la base de datos ejecutando los comandos del script 4.28.

```
#!/usr/bin/env bash

sudo mysql --user=root mysql

CREATE USER 'phpmyadmin'@'localhost' IDENTIFIED BY 'some_pass';
GRANT ALL PRIVILEGES ON *.* TO 'phpmyadmin'@'localhost' WITH GRANT
OPTION;
FLUSH PRIVILEGES;
```

Script 4.28: Creación del usuario phpmyadmin para la base de datos

A continuación hay que editar el archivo `/etc/dbconfig-common/phpmyadmin.conf` para añadir el usuario y la contraseña anteriormente configuradas.

```
# dbc_dbuser: database user
#       the name of the user who we will use to connect to the
#       database.
dbc_dbuser='phpmyadmin'

# dbc_dbpass: database user password
#       the password to use with the above username when connecting
#       to a database, if one is required
dbc_dbpass='some_pass'
```

Script 4.29: Modificación del fichero de configuración de phpmyadmin

El último paso para la configuración es la edición de los ficheros de configuración de apache que se encuentran en `/etc/apache2/sites-available` para que las web por defecto apunten hacia la carpeta que contendrá los ficheros. En el script 4.30 se muestra el contenido del archivo `000-default.conf` mientras que en el script 4.31 está el del archivo `default-ssl.conf`.

```
<VirtualHost *:80>
    ServerAdmin hoyvin.mayvin@protonmail.com
    DocumentRoot /var/www/tfm/tfm2/public

    <Directory /var/www/tfm/tfm2>
        AllowOverride All
    </Directory>
```

```

# Available loglevels: trace8, ..., trace1, debug, info,
# notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For
# example the
# following line enables the CGI configuration for this host
# only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

Script 4.30: Fichero 000-default.conf

```

<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin hoyvin.mayvin@protonmail.com
    DocumentRoot /var/www/tfm/tfm2/public

    <Directory /var/www/tfm/tfm2>
      AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    #   SSL Engine Switch:
    #   Enable/Disable SSL for this virtual host.
    SSLEngine on

    #   A self-signed (snakeoil) certificate can be created by
    #   installing
    #   the ssl-cert package. See
    #   /usr/share/doc/apache2/README.Debian.gz for more info.
    #   If both key and certificate are stored in the same file, only
    #   the
    #   SSLCertificateFile directive is needed.
    SSLCertificateFile /etc/ssl/private/server.crt
    SSLCertificateKeyFile /etc/ssl/private/server.key

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>

  </VirtualHost>
</IfModule>

```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Script 4.31: Fichero default-ssl.conf

No se incluye en esta documentación la instalación de los certificados autofirmados generados. En este punto hay que habilitar el site default-ssl en el servidor de apache y proceder a su reinicio. Esto se realiza ejecutando `sudo a2ensite default-ssl` y a continuación `sudo systemctl restart apache`.

Importación de la base de datos

Para crear la base de datos necesaria para la gestión de los usuarios y de guardar el log de los logins que realizan se hace ejecutando el script sql 4.32. Se puede realizar bien a través de consola de comandos o a través de la interfaz web phpmyadmin.

```
-- phpMyAdmin SQL Dump
-- version 4.6.6deb4
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Jun 16, 2018 at 11:47 AM
-- Server version: 10.1.23-MariaDB-9+deb9u1
-- PHP Version: 7.0.27-0+deb9u1

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

-- Database: 'tfm'
--
CREATE DATABASE IF NOT EXISTS 'tfm' DEFAULT CHARACTER SET utf8mb4
    COLLATE utf8mb4_general_ci;
USE 'tfm';

-----
-- Table structure for table 'login_registers'
--

CREATE TABLE 'login_registers' (
    'id' int(10) UNSIGNED NOT NULL,
    'username' varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
    'date' date NOT NULL,
    'time' time NOT NULL,
    'action' mediumtext COLLATE utf8mb4_unicode_ci NOT NULL,
    'created_at' timestamp NULL DEFAULT NULL,
    'updated_at' timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- 
-- Table structure for table 'migrations'
-- 

CREATE TABLE 'migrations' (
  'id' int(10) UNSIGNED NOT NULL,
  'migration' varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
  'batch' int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- 
-- Dumping data for table 'migrations'
-- 

INSERT INTO 'migrations' ('id', 'migration', 'batch') VALUES
(19, '2014_10_12_000000_create_users_table', 1),
(20, '2014_10_12_100000_create_password_resets_table', 1),
(21, '2018_03_17_224126_create_login_registers', 1);

-- 
-- Table structure for table 'password_resets'
-- 

CREATE TABLE 'password_resets' (
  'username' varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
  'token' varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
  'created_at' timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- 
-- Table structure for table 'users'
-- 

CREATE TABLE 'users' (
  'id' int(10) UNSIGNED NOT NULL,
  'username' varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
  'password' varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
  'super' tinyint(1) NOT NULL,
  'remember_token' varchar(100) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  'created_at' timestamp NULL DEFAULT NULL,
  'updated_at' timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- 
-- Dumping data for table 'users'
-- 

INSERT INTO 'users' ('id', 'username', 'password', 'super', 'remember_token', 'created_at', 'updated_at') VALUES
(1, 'admin', '$2y$10$E2t/Ggihja8DNRYx80ArHex2eU9VTrv9te3lHSLAtYMFMVUnUXRt0', 1, 'DqRLiBWl0iqT0ouSHhzupkKQUvx1F55Mi1tWwZDWVjMdba7HcBLQBGFqNhUt', NULL, NULL),
(2, 'readonly', '$2y$10$yHrez6y/S50QdeZ52mfE50.mhUENEDkR0/KxASyZzvtr1VBA/SA06', 0, NULL, NULL, NULL);
```

```

-- 
-- Indexes for dumped tables
-- 

-- 
-- Indexes for table `login_registers`
-- 
ALTER TABLE `login_registers`
  ADD PRIMARY KEY (`id`);

-- 
-- Indexes for table `migrations`
-- 
ALTER TABLE `migrations`
  ADD PRIMARY KEY (`id`);

-- 
-- Indexes for table `password_resets`
-- 
ALTER TABLE `password_resets`
  ADD KEY `password_resets_username_index` (`username`);

-- 
-- Indexes for table `users`
-- 
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `users_username_unique` (`username`);

-- 
-- AUTO_INCREMENT for dumped tables
-- 

-- 
-- AUTO_INCREMENT for table `login_registers`
-- 
ALTER TABLE `login_registers`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
  =18;
-- 
-- AUTO_INCREMENT for table `migrations`
-- 
ALTER TABLE `migrations`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
  =22;
-- 
-- AUTO_INCREMENT for table `users`
-- 
ALTER TABLE `users`
  MODIFY `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT
  =3;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

Script 4.32: Creación de la base de datos y tablas necesarias

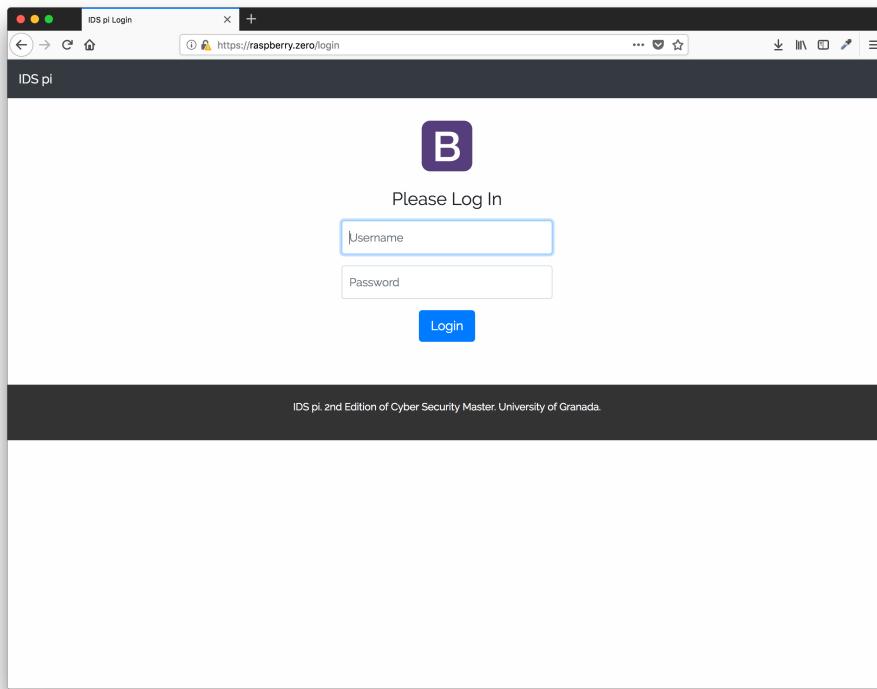


Figura 4.13: Página de login de la interfaz web del sistema

Instalación de la interfaz Web

El último punto para poder hacer uso de la interfaz web desarrollada es copiar los archivos a la ruta configurada en el servidor apache y una vez hecho ya es posible acceder desde un navegador siempre y cuando haya conectividad IP con la raspberry que lo alberga. La configuración por defecto permite el acceso tanto a través de http como de https.

Uso de la interfaz web

La página por defecto, mostrada en la figura 4.13, consiste en un formulario con un campo para introducir el usuario y la contraseña. Por defecto está configurado el usuario con privilegios totales sobre el sistema `admin` con contraseña `admin` y el usuario de solo lectura `readonly` con contraseña `readonly`.

Tras acceder al sistema por defecto se navega a la sección **Home** (ver figura 4.14) y aparece en la parte superior la barra de navegación, a la derecha los datos de contacto del autor y se muestra una explicación de las diferentes secciones y apartados que tiene la interfaz.

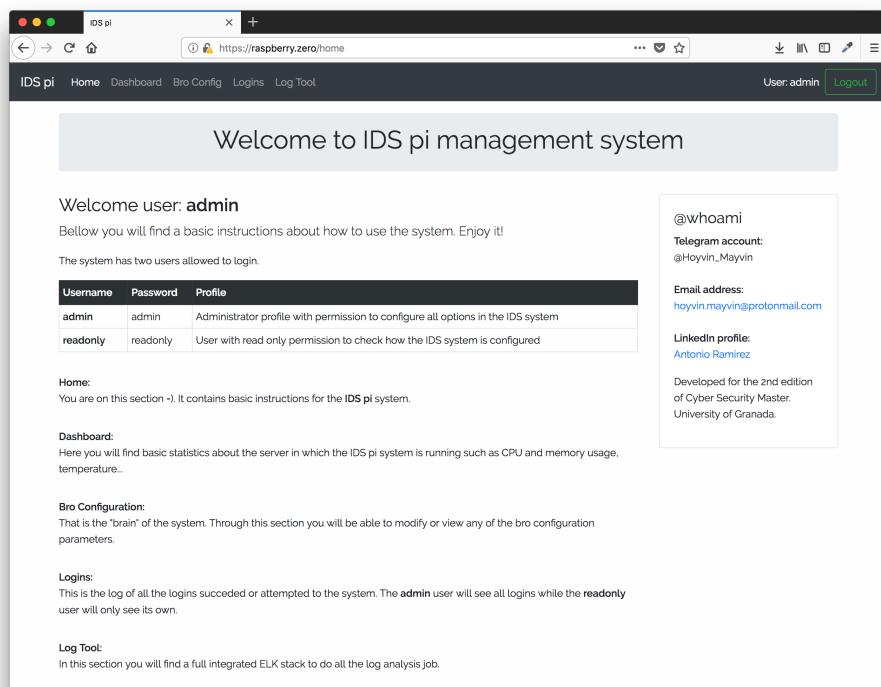


Figura 4.14: Menú Home de la interfaz web

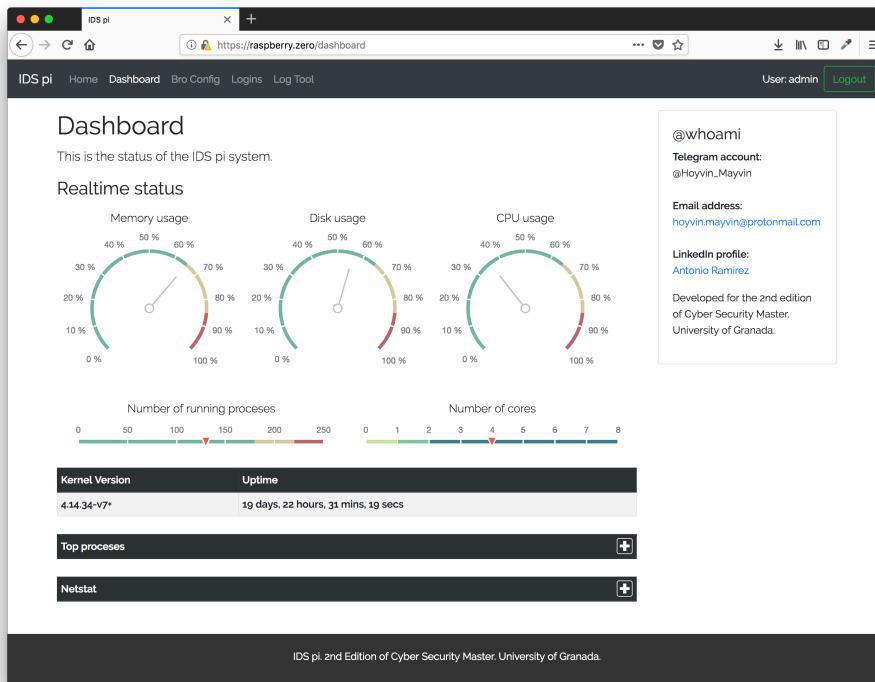


Figura 4.15: Menú Dashboard de la interfaz web

La siguiente sección es **Dashboard** en donde se han desarrollado unos medidores para conocer el estado de la memoria, la CPU y el disco del sistema, el número de procesos en ejecución, el número de cores del sistema. Se muestra la figura 4.15.

Además, en esta sección se han incluido dos desplegables para poder conocer la lista de los procesos en ejecución y el estado de las conexiones de red. Se pueden ver ejemplos en las figuras 4.16 y 4.17.

El menú **Bro Config** es en el que se muestran todas las opciones de configuración relativas al sistema IDS implementado con Bro, Loki y Critical Stack. Muestra el hostname y la fecha y hora del equipo, el estado de la ejecución de Bro, las redes locales configuradas para su análisis, la fecha de actualización de las firmas de Loki y de Critical Stack. Además muestra la información del tamaño reservado y utilizado para los PCAPs y para los ficheros extraídos.

Además, incorpora dos navegadores de ficheros a través de los cuales es posible descargar las muestras para un análisis en detalle.

Se muestra la visualización de este menú en la figura 4.18.

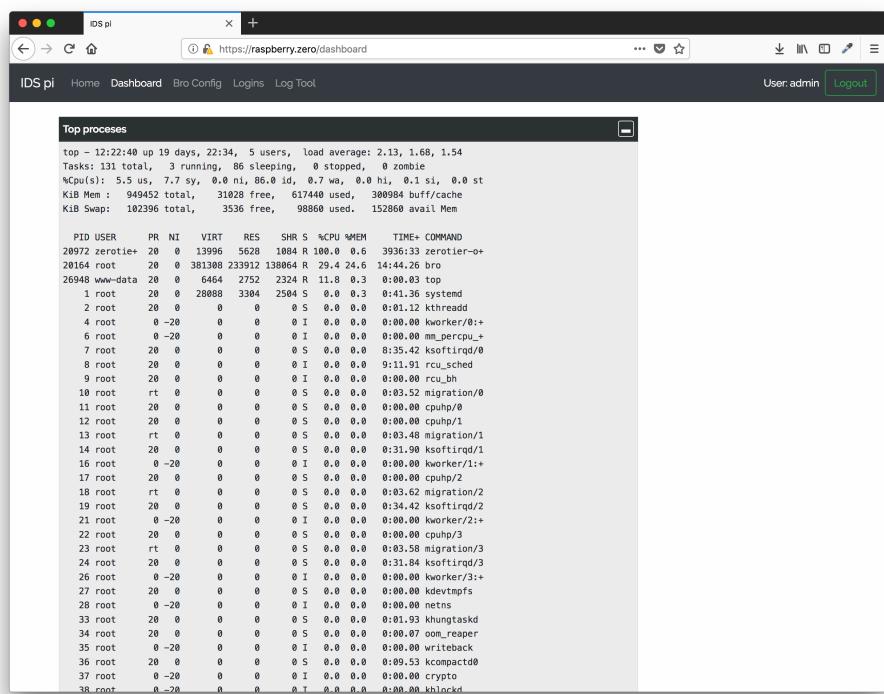
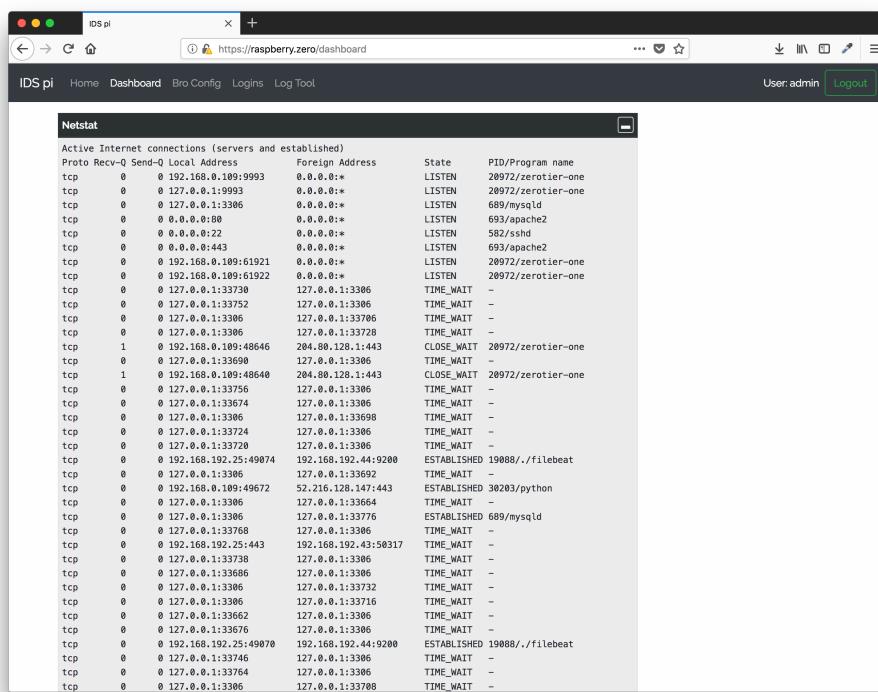


Figura 4.16: Detalle de los procesos en ejecución del menú Dashboard de la interfaz web



Active Internet connections (servers and established)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	192.168.0.109:9993	0.0.0.0:*	LISTEN 28972/zerotier-one
tcp	0	0	127.0.0.1:19993	0.0.0.0:*	LISTEN 28972/zerotier-one
tcp	0	0	127.0.0.1:3386	0.0.0.0:*	LISTEN 689/mysqld
tcp	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN 693/apache2
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN 582/sshd
tcp	0	0	0.0.0.0:443	0.0.0.0:*	LISTEN 693/apache2
tcp	0	0	192.168.0.109:61921	0.0.0.0:*	LISTEN 28972/zerotier-one
tcp	0	0	192.168.0.109:61922	0.0.0.0:*	LISTEN 28972/zerotier-one
tcp	0	0	127.0.0.1:33738	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:33752	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:3386	127.0.0.1:33706	TIME_WAIT -
tcp	0	0	127.0.0.1:3386	127.0.0.1:33728	TIME_WAIT -
tcp	1	0	192.168.0.109:48646	284.88.128.1:443	CLOSE_WAIT 28972/zerotier-one
tcp	0	0	127.0.0.1:33690	127.0.0.1:3306	TIME_WAIT -
tcp	1	0	192.168.0.109:48648	284.88.128.1:443	CLOSE_WAIT 28972/zerotier-one
tcp	0	0	127.0.0.1:33756	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:33674	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:3386	127.0.0.1:33696	TIME_WAIT -
tcp	0	0	127.0.0.1:33724	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:33728	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	192.168.192.25:49074	192.168.192.44:9280	ESTABLISHED 19088//filebeat
tcp	0	0	127.0.0.1:3386	127.0.0.1:33692	TIME_WAIT -
tcp	0	0	192.168.0.109:49672	52.216.128.147:443	ESTABLISHED 30203/python
tcp	0	0	127.0.0.1:3386	127.0.0.1:33664	TIME_WAIT -
tcp	0	0	127.0.0.1:3386	127.0.0.1:33776	ESTABLISHED 689/mysqld
tcp	0	0	127.0.0.1:33768	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	192.168.192.25:443	192.168.192.43:59317	TIME_WAIT -
tcp	0	0	127.0.0.1:33738	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:33666	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:3386	127.0.0.1:33732	TIME_WAIT -
tcp	0	0	127.0.0.1:3386	127.0.0.1:33716	TIME_WAIT -
tcp	0	0	127.0.0.1:33662	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:33676	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	192.168.192.25:49070	192.168.192.44:9280	ESTABLISHED 19088//filebeat
tcp	0	0	127.0.0.1:33746	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:33764	127.0.0.1:3306	TIME_WAIT -
tcp	0	0	127.0.0.1:3306	127.0.0.1:3308	TIME_WAIT -

Figura 4.17: Detalle de las conexiones de red del menú Dashboard de la interfaz web

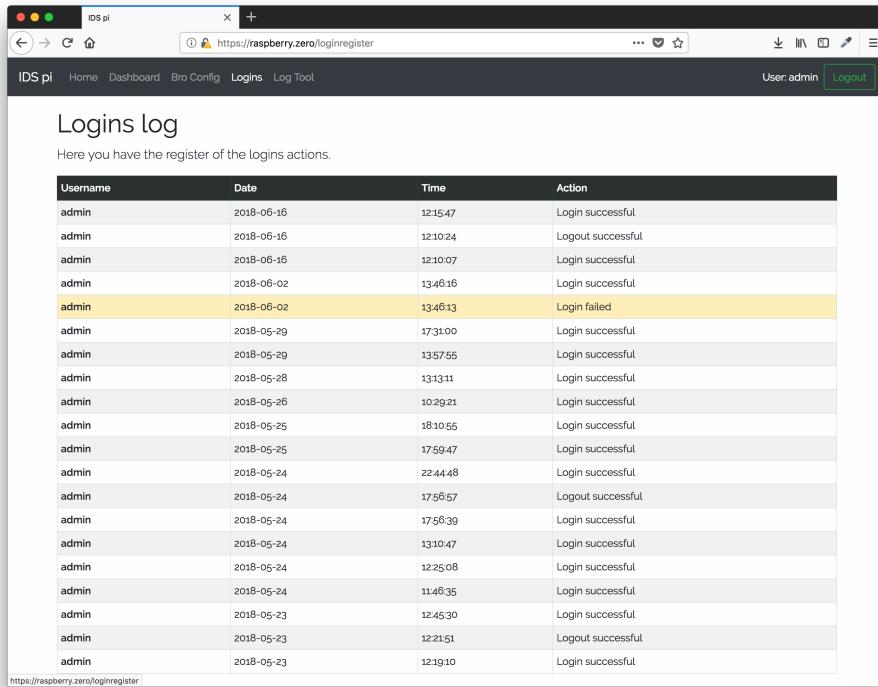
The screenshot shows the 'Bro and system configuration' page of the IDS pi web interface. The top navigation bar includes 'IDS pi', 'Home', 'Dashboard', 'Bro Config', 'Logins', 'Log Tool', 'User: admin', and 'Logout'. The main content area is titled 'Bro and system configuration' and contains the following sections:

- System Time:** 29/07/18 - 15:57:05
- Bro Status:** running, IP: 192.168.188.160/32, 84.236.188.195/32, Last Update: Jul 29 15:30
- PCAP Files:** A list of PCAP files with their names, last modified times, and sizes:
 - dump-1532856665.pcap (2018-07-29 11:53, 13M)
 - dump-1532853043.pcap (2018-07-29 11:30, 104M)
 - dump-1532845845.pcap (2018-07-29 10:30, 106M)
 - dump-1532838636.pcap (2018-07-29 08:30, 106M)
- Extracted Files:** A list of extracted files with their names, last modified times, and sizes:
 - HTTP-F72b63mT9RK8OWmWk.ocsp-response (2018-07-29 11:40, 471)
 - HTTP-Fse183L479Dgaw8q.json (2018-07-29 11:09, 4.8K)
 - HTTP-FjBQfP2k6aBL1nYjZ.json (2018-07-29 11:09, 5.2K)

Right sidebar:

- @whoami**
- Telegram account:** @Hoyvin_Mayvin
- Email address:** hoyvin.mayvin@protonmail.com
- LinkedIn profile:** Antonio Ramirez
- Developer note:** Developed for the 2nd edition of Cyber Security Master, University of Granada.

Figura 4.18: Menú Bro Config de la interfaz web



Username	Date	Time	Action
admin	2018-06-16	12:15:47	Login successful
admin	2018-06-16	12:10:24	Logout successful
admin	2018-06-16	12:10:07	Login successful
admin	2018-06-02	13:46:16	Login successful
admin	2018-06-02	13:49:13	Login failed
admin	2018-05-29	17:31:00	Login successful
admin	2018-05-29	13:57:55	Login successful
admin	2018-05-28	13:13:11	Login successful
admin	2018-05-26	10:29:21	Login successful
admin	2018-05-25	18:10:55	Login successful
admin	2018-05-25	17:59:47	Login successful
admin	2018-05-24	22:44:48	Login successful
admin	2018-05-24	17:56:57	Logout successful
admin	2018-05-24	17:56:39	Login successful
admin	2018-05-24	13:10:47	Login successful
admin	2018-05-24	12:25:08	Login successful
admin	2018-05-24	11:46:35	Login successful
admin	2018-05-23	12:45:30	Login successful
admin	2018-05-23	12:21:51	Logout successful
admin	2018-05-23	12:19:10	Login successful

Figura 4.19: Menú Logins de la interfaz web

En la sección **Logins** se muestra la lectura de la base de datos del registro de accesos al sistema de tal forma que el usuario **admin** puede ver todos los producidos y el usuario **readonly** solo puede visualizar los correspondientes a él mismo. Figura 4.19.

El último de los menús de la interfaz web es el llamado **Log Tool**. Este menú, mostrado en la figura 4.20, simplemente contiene un iframe en el que se ha incluido la interfaz web del servidor Kibana y además se ha creado un hiper vínculo para que se abra en otra ventana del navegador. Los detalles sobre los logs recogidos y la explotación de estos se encuentra en la sección 4.4.2.

4.3. Instalación del servidor Logstash, Elasticsearch y Kibana

En las siguientes secciones se explica el proceso de instalación de un servidor ELK en un servidor Ubuntu 18.04 que será el encargado de recibir los logs generados por la sonda para su posterior tratamiento y explotación.

90 4.3. Instalación del servidor Logstash, Elasticsearch y Kibana

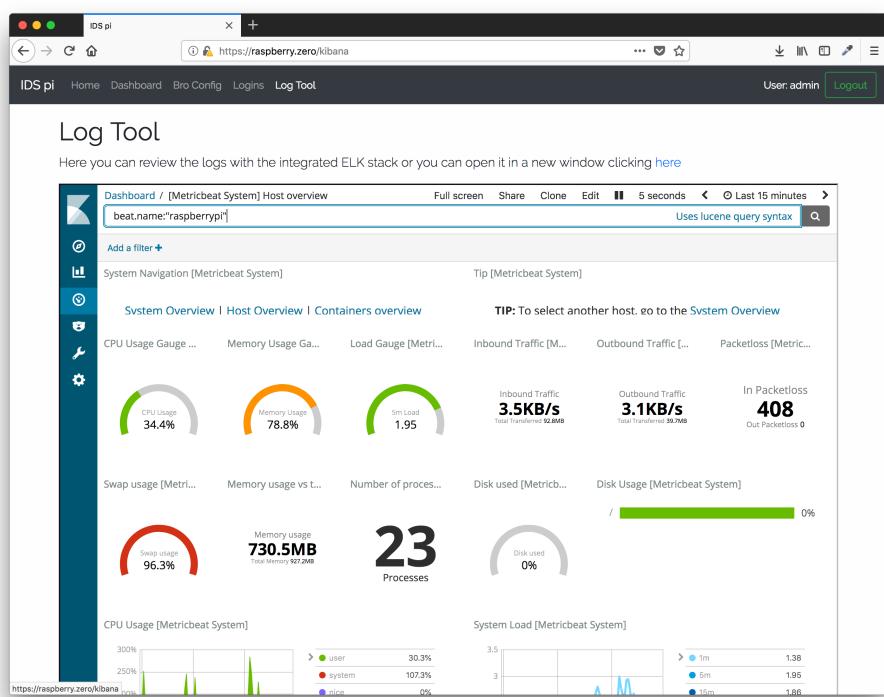


Figura 4.20: Menú Log Tool de la interfaz web

Se ha realizado usando el procedimiento descrito en [33] como referencia.

El primer paso del proceso es la instalación de las dependencias y la instalación del servidor web Nginx. Se debe tener en cuenta que Logstash no soporta la versión 10 de Java por lo que hay que instalar la versión 8.

```
sudo apt install openjdk-8-jre apt-transport-https wget nginx
```

Script 4.33: Instalación de dependencias para ELK

El siguiente paso es importar la clave GPG de Elastic al servidor.

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

Script 4.34: Instalación de clave GPG

A continuación se crea un archivo en `/etc/apt/sources.list.d/elastic.list` y pegar el siguiente contenido deb `https://artifacts.elastic.co/packages/6.x/apt stable main`. Guardar el archivo y actualizar apt `sudo apt update`.

4.3.1. Instalación y configuración de Elasticsearch y Kibana

Elasticsearch y Kibana

La instalación se realiza directamente desde apt con el comando `sudo apt install elasticsearch kibana`. Una vez instalado hay que editar el fichero de configuración de Kibana `/etc/kibana/kibana.yml` para indicar cual es el servidor. Simplemente hay que buscar la siguiente línea y descomentárla.

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

Script 4.35: Configuración Kibana

Para habilitar el geo posicionamiento en base a las direcciones IP hay que instalar en el servidor Elasticsearch los siguientes paquetes.

```
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install ingest-geoip
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install ingest-user-agent
```

Script 4.36: Configuración Kibana

Seguidamente se configuran los servicios para su arranque automático y se inician.

92 4.3. Instalación del servidor Logstash, Elasticsearch y Kibana

```
sudo systemctl enable kibana
sudo systemctl restart kibana
sudo systemctl enable elasticsearch
sudo systemctl start elasticsearch
```

Script 4.37: Configuración y arranque de servicios elasticsearch y Kibana

Nginx

Aunque la instancia de Kibana es directamente accesible con una navegador web a través del puerto 5601 se utiliza Nginx para presentar esta web y poder controlar, y, en caso necesario securizar estos accesos. En la ruta `/etc/nginx/sites-available` se crea un nuevo archivo con el nombre del site a publicar y la configuración mostrada en el script 4.38.

```
server {
    listen 80;
    listen 443 ssl;

    ssl_certificate /etc/ssl/private/domain.crt;
    ssl_certificate_key /etc/ssl/private/domain.key;

    server_name kibana.zero;

    location / {
        proxy_pass http://localhost:5601;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Script 4.38: Configuración de Nginx

Una vez hecho se crea un enlace simbólico y se elimina el sitio por defecto.

```
sudo rm /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/kibana /etc/nginx/sites-
enabled/kibana
sudo systemctl enable nginx
sudo systemctl restart nginx
```

Script 4.39: Configuración del enlace simbólico para Nginx

4.3.2. Instalación y configuración de Logstash

La instalación de Logstash se realiza directamente desde el gestor de paquetes APT, `sudo apt install logstash`. Una vez instalado se copia el

siguiente archivo de configuración 4.40 en la ruta `/etc/logstash/conf.d/`.

```
input {
  beats {
    port => 5044
  }
}

filter {

  if [bro] == "conn" or [bro] == "dns" or [bro] == "http" or [bro] ==
  "notice" or [bro] == "ssl" or [bro] == "intel" {
    geoip {
      source => "id.resp_h"
      target => "geobro_resp"
      add_field => [ "[geobro_resp][coordinates]", "%{[geobro_resp][longitude]}" ]
      add_field => [ "[geobro_resp][coordinates]", "%{[geobro_resp][latitude]}" ]
    }
    geoip {
      source => "id.orig_h"
      target => "geobro_orig"
      add_field => [ "[geobro_orig][coordinates]", "%{[geobro_orig][longitude]}" ]
      add_field => [ "[geobro_orig][coordinates]", "%{[geobro_orig][latitude]}" ]
    }
    mutate {
      convert => [ "[geobro_resp][coordinates]", "float" ]
      convert => [ "[geobro_orig][coordinates]", "float" ]
    }
  }

  if [bro] == "files"{
    geoip {
      source => "rx_hosts"
      target => "geobro_resp"
      add_field => [ "[geobro_orig][coordinates]", "%{[geobro_orig][longitude]}" ]
      add_field => [ "[geobro_orig][coordinates]", "%{[geobro_orig][latitude]}" ]
    }
    geoip {
      source => "tx_hosts"
      target => "geobro_orig"
      add_field => [ "[geobro_orig][coordinates]", "%{[geobro_orig][longitude]}" ]
      add_field => [ "[geobro_orig][coordinates]", "%{[geobro_orig][latitude]}" ]
    }
    mutate {
      convert => [ "[geobro_resp][coordinates]", "float" ]
      convert => [ "[geobro_orig][coordinates]", "float" ]
    }
  }
}
```

```

output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "brobeat-%{+yyyy.MM.dd}"
    document_type => "%{[@metadata][type]}"
  }
}

```

Script 4.40: Configuración de Logstash

Hay que retocar la configuración del servicio como se indica en 4.41 que se crea por defecto (`/etc/systemd/system/logstash.service`) para que cargue este fichero de configuración en el que se incorpora la posición GPS de las direcciones IP que se encuentran en los logs de Bro.

```

[Unit]
Description=logstash

[Service]
Type=simple
User=logstash
Group=logstash
# Load env vars from /etc/default/ and /etc/sysconfig/ if they exist.
# Prefixing the path with '-' makes it try to load, but if the file
# doesn't
# exist, it continues onward.
EnvironmentFile=-/etc/default/logstash
EnvironmentFile=-/etc/sysconfig/logstash
ExecStart=/usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/
  logstash.conf
Restart=always
WorkingDirectory=/
Nice=19
LimitNOFILE=16384

[Install]
WantedBy=multi-user.target

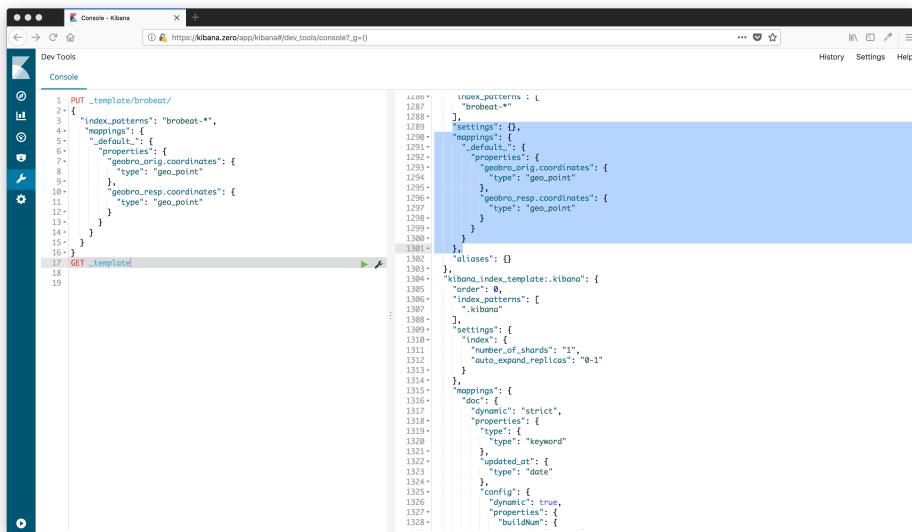
```

Script 4.41: Configuración del servicio de Logstash

Para hacer uso del tipo de datos Geopoint es necesario añadir un mapping al servidor de Elasticsearch. Para ello se pueden utilizar las herramientas de desarrolladores incluidas en el portal. Se muestra un ejemplo de cómo configurar esto en la figura 4.21.

4.4. Gestión y análisis de logs

Una vez con el sistema instalado y configurado acorde a lo expuesto en las secciones anteriores el motor IDS instalado en la Raspberry Pi debe estar reportando los logs que genera junto con los generados por el sistema y las métricas de su estado al servidor central ELK.



```

PUT _template/brobeat/
{
  "index_patterns": "brobeat-*",
  "mappings": {
    "_default_": {
      "properties": {
        "geobro_orig_coordinates": {
          "type": "geo_point"
        },
        "geobro_resp_coordinates": {
          "type": "geo_point"
        }
      }
    }
  }
}

{
  "index_patterns": "brobeat-*",
  "settings": {
    "mappings": {
      "_default_": {
        "properties": {
          "geobro_orig_coordinates": {
            "type": "geo_point"
          },
          "geobro_resp_coordinates": {
            "type": "geo_point"
          }
        }
      }
    }
  },
  "aliases": {}
}

{
  "kibana_index_template": {
    "order": 0,
    "index_patterns": [
      "brobeat-*"
    ],
    "settings": {
      "index": {
        "number_of_shards": "1",
        "auto_expand_replicas": "0-1"
      }
    },
    "mappings": {
      "_doc": {
        "dynamic": "strict",
        "properties": {
          "type": {
            "type": "keyword"
          },
          "updated_at": {
            "type": "date"
          }
        }
      },
      "config": {
        "dynamic": true,
        "properties": {
          "buildNum": {
            "type": "string"
          }
        }
      }
    }
  }
}

```

Figura 4.21: Creación de un mapping para tipos de dato Geopoint

4.4.1. Creación de índices

Si se han seguido los pasos enumerados en la sección 4.2.5 se habrán creado en el servidor Elasticsearch dos índices por defecto: `filebeat-*` y `metricbeat-*` usados para indexar los logs enviados por la instancia de Filebeat del sistema y las métricas de Metricbeat. Para poder indexar los índices enviados por Bro hay que crear un nuevo índice cuyo nombre coincida con el indicado en el archivo de configuración de Filebeat para Bro y en el de Logstash. El índice configurado es `index => \"brobeat-\%\{+yyyy.MM.dd\}\\"` por lo que el índice a crear debe de ser `brobeat-*`. En la figura 4.22 se muestra un ejemplo de cómo se haría la configuración.

Para que se oculten algunos mensajes de alerta en Kibana hay que elegir un índice por defecto. Se puede elegir cualquiera de ellos y en la figura 4.23 se muestra cómo sería este proceso.

Ahora es posible, dentro del menú `Discover` de Kibana poder encontrar todos los logs que se están recibiendo y seleccionar los que vayan a ser tratados. La figura 4.24 lo muestra.

4.4.2. Visualización y explotación de logs del sistema

Como antes de lanzar Filebeat y Metricbeat para recoger logs y métricas del sistema se realizó un proceso de instalación sobre el servidor Kibana de los dashboards y visualizaciones que trae por defecto (ver sección 4.2.5) es

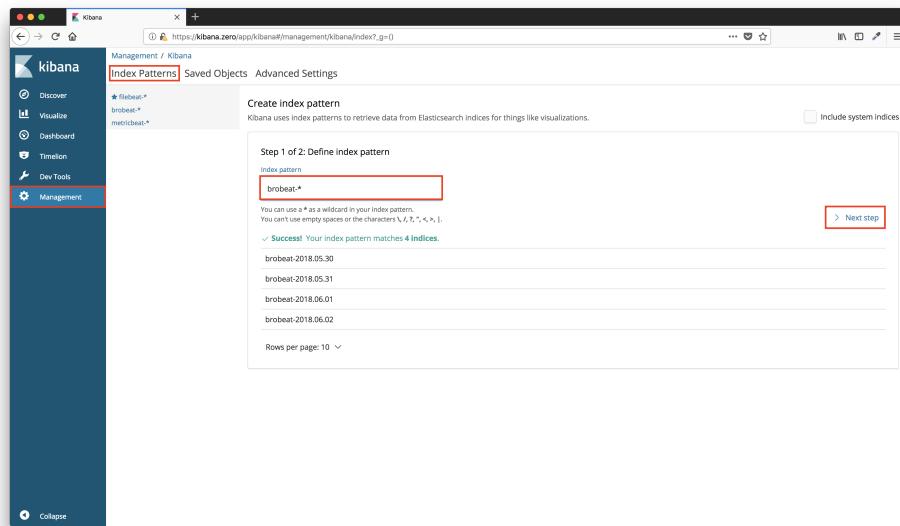


Figura 4.22: Creación de índices en Elasticsearch para logs Bro

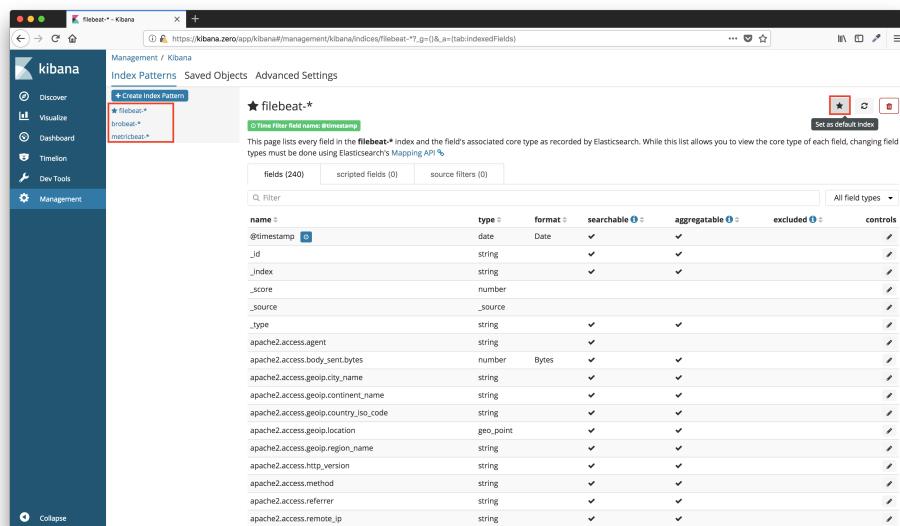


Figura 4.23: Selección de un índice por defecto en Elasticsearch

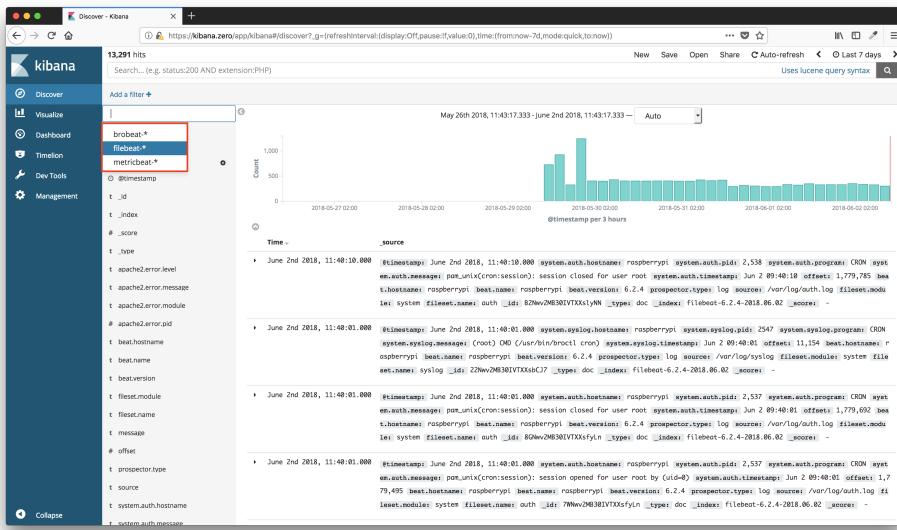


Figura 4.24: Elección de los diferentes índices en Kibana

possible hacer uso de ellas para explotar la información más significativa.

Metricbeat

Con Metricbeat se obtienen una medición del estado de la sonda y estadísticas del rendimiento. Para obtener una visión general del estado del sistema se puede acceder al dashboard [\[Metricbeat System\] Host overview](#) que muestra una visualización del consumo de CPU, de memoria, de los paquetes entrantes y salientes del sistema, del top de procesos por CPU y memoria y del tráfico total gestionado por cada una de las interfaces. Figuras 4.25, 4.26 y 4.27.

Filebeat

Con el servicio `filebeat-system` configurado se envían al servidor Elasticsearch el conjunto de todos los logs del sistema y haciendo uso de los dashboards y visualizaciones ya pre cargados es sencillo obtener, por ejemplo, las estadísticas de acceso al servidor Apache2 que gestiona la interfaz web, accesos SSH, estadísticas de syslog a través del dashboard [\[Filebeat Apache2\] Access and error logs](#).

En la figura 4.28 se muestra el resumen de los accesos al sistema de gestión del IDS, el tipo de navegador, y, si estuviera publicado en internet se mostrarían las diferentes ubicaciones desde donde de accede.

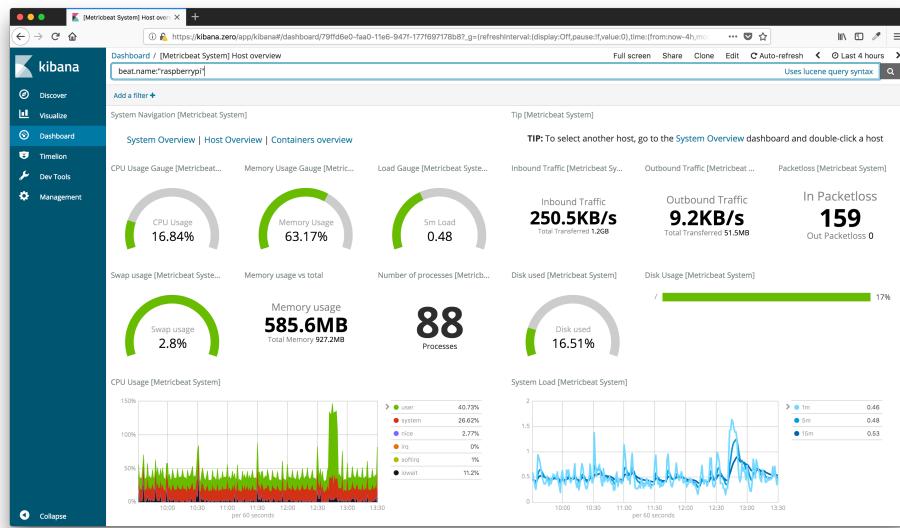


Figura 4.25: Dashboard de Metricbeat con el estado del sistema (1)



Figura 4.26: Dashboard de Metricbeat con el estado del sistema (2)

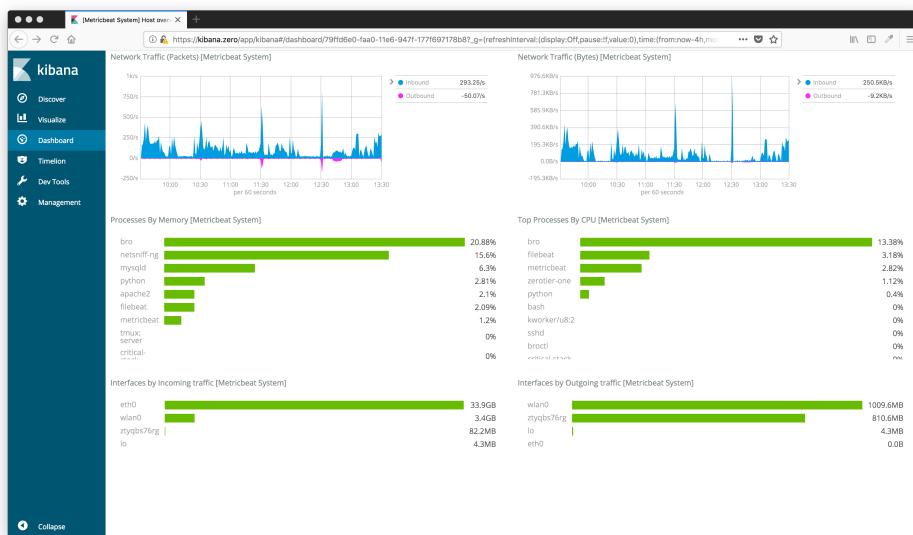


Figura 4.27: Dashboard de Metricbeat con el estado del sistema (3)

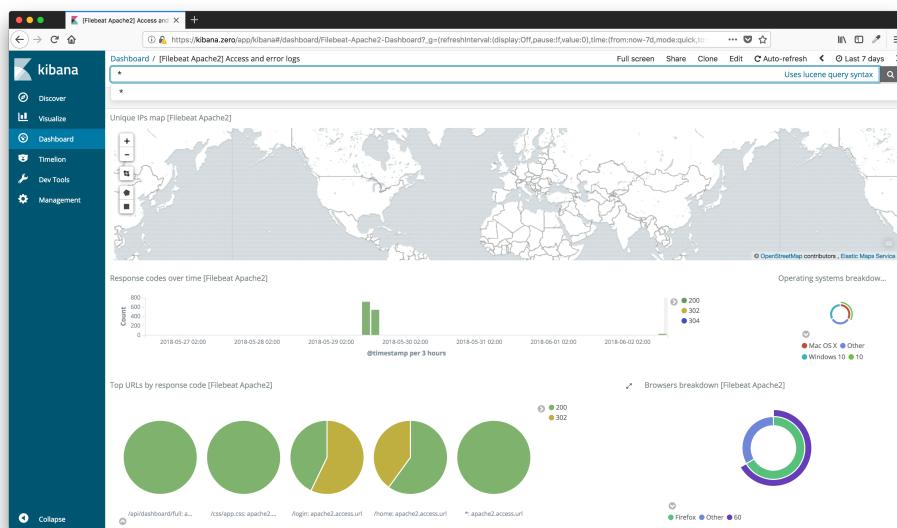


Figura 4.28: Dashboard del resumen de accesos al servidor Apache2

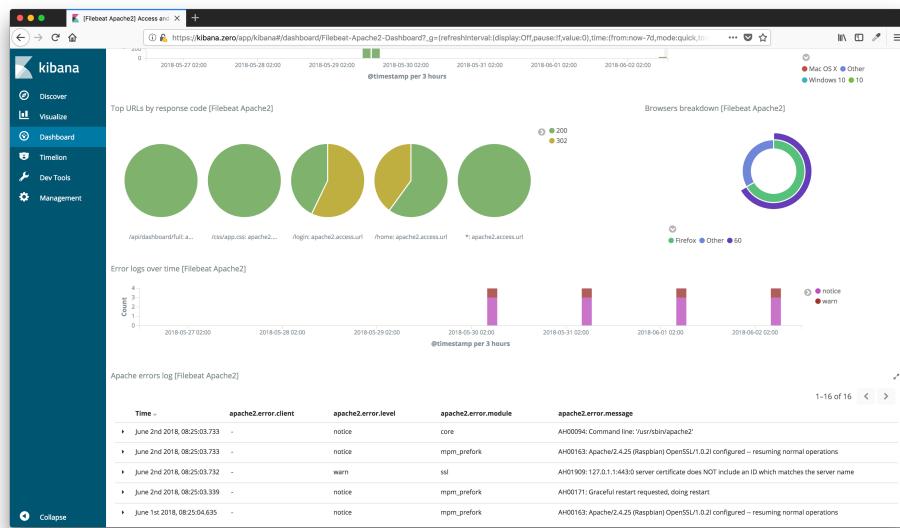


Figura 4.29: Dashboard del log de errores del servidor Apache2

En la parte inferior de este mismo dashboard se visualiza el registro de errores, su distribución en el tiempo y sus últimas entradas en el log de errores. Ver figura 4.29.

Navegando hasta el dashboard **[Filebeat System] Syslog dashboard**, figura 4.30, se obtiene un resumen de los eventos de syslog por tiempo, una distribución de qué procesos son los que mayor tiempo llevan en ejecución y una visualización de las entradas en el syslog en donde es fácilmente observar si las tareas programadas periódicas se están ejecutando correctamente o si por el contrario pudiera haber algún problema con ellas y anticiparse a posibles problemas, como por ejemplo, el llenado del almacenamiento del sistema.

Otro de los dashboards interesantes a monitorizar sobre el sistema es el de los accesos por SSH, figura 4.31. El nombre de este panel de control es **[Filebeat System] SSH login attempts** y de un solo vistazo es posible controlar los diferentes acceso por SSH que ha habido al sistema, los usuarios con un mayor índice de logins fallidos y una visión completa del log de accesos.

Para finalizar esta sección, otra de las pantallas interesantes de revisar es el de los comandos SUDO ejecutados los cuales aparecen ordenados por el usuario que los creó. Un ejemplo se muestra en la figura 4.32.

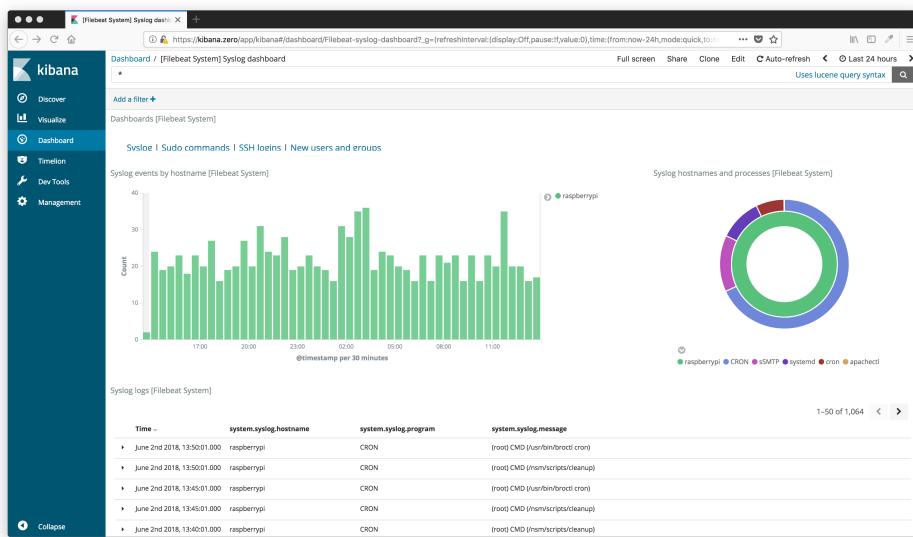


Figura 4.30: Dashboard de syslog

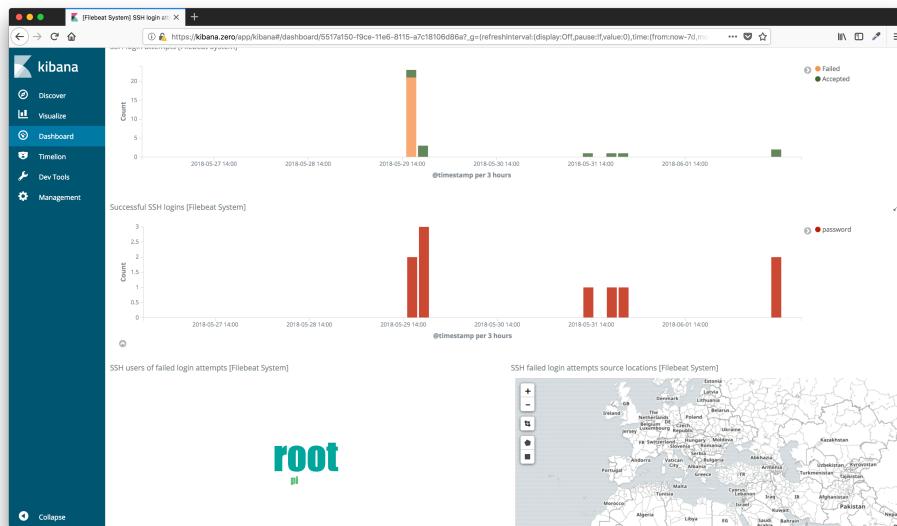


Figura 4.31: Dashboard de accesos SSH al sistema

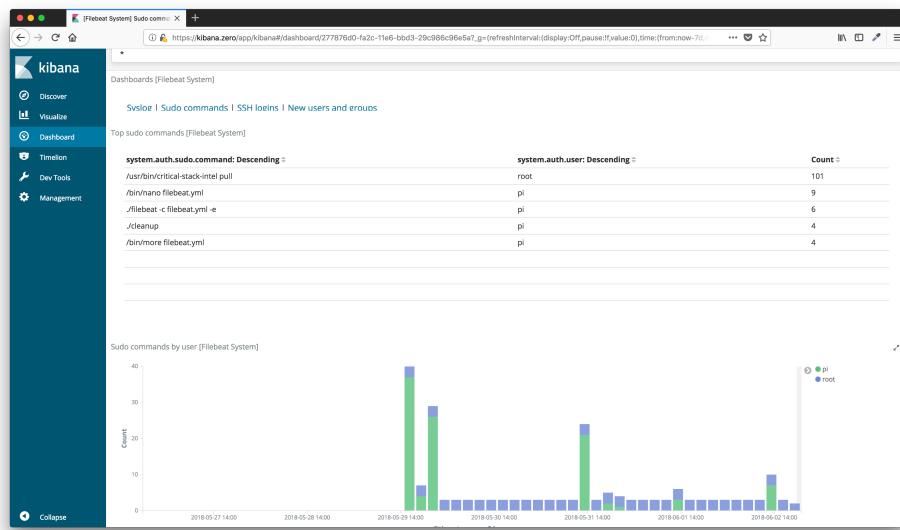


Figura 4.32: Dashboard de comandos lanzados con SUDO en el sistema

4.4.3. Visualización y explotación de logs del IDS

Dentro del conjunto de todos los logs que genera Bro, en los siguientes apartados se muestra con más detalle algunos ejemplos de cómo poder extraer información y en particular cómo poder tratarla en relación a la seguridad.

Logs bro:intel

El primer, y más importante log, que proporciona información relativa a la seguridad es el log `/var/log/bro/current/intel.log`. Este log es generado por el motor Criticalstack integrado en Bro y guarda los accesos hacia las direcciones IP o dominios que se encuentren dentro de la base de datos de firmas que mantiene Critical Stack (ver figura 4.33) y que se puede encontrar en `/opt/critical-stack/frameworks/intel/master-public.bro.dat`.

En la figura 4.34 se muestra un ejemplo de la recepción de una entrada de estos logs. De manera resumida, si la configuración del sensor en la web de Critical Stack ha sido correcta y se han creado las suscripciones a feeds con buena reputación y de confianza, cada mensaje que llegue de este log deberá de crear automáticamente una investigación para determinar el origen del acceso y las razones de él.

A continuación, en el script 4.42, se adjunta una muestra de una entrada del log de Critical Stack recogido por Bro en formato json. En el campo

```
TFM_MPICS_latex - pi@raspberrypi: ~ - ssh pi@raspberrypi.zero — 136x48
[pi@raspberrypi: ~ $ sudo more /opt/critical-stack/frameworks/intel/master-public.bro.dat
filed indicator indicator_type meta.source meta.no_notice
128.199.138.74 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
165.227.62.147 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
toolsathomes.com Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
younr.kro.kr Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
15.186.86.182 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
all.davidwirlkins.com Intel:DOMAIN from http://hosts-file.net/exp.txt via intel.criticalstack.com F
linkwo.com Intel:DOMAIN from http://hosts-file.net/exp.txt via intel.criticalstack.com F
hjhmgbxyiin5lkk1.1612zt.top Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
185.227.82.56 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
ahuqk5q4v3nzb.dhs4sn.com Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
F aqamezuce.com Intel:DOMAIN from http://hosts-file.net/exp.txt via intel.criticalstack.com F
home.lauritiaeavaiting.com Intel:DOMAIN from http://hosts-file.net/exp.txt via intel.criticalstack.com F
178.27.74.12 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
193.25.118.281 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
yigusa.sle5.com Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
92.121.45.118 Intel:ADDR from http://rules.emerginthreets.net/blockrules/emerging-bottc.rules via intel.criticalstack.com F
F 37.143.11.165 Intel:ADDR from http://rules.emerginthreets.net/blockrules/emerging-bottc.rules via intel.criticalstack.com F
F 163.172.142.92 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
185.222.202.13 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
188.166.118.93 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
2008:1407:ccfe:9201::0000:0000:0000:0000: Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
arvisiion.com.co Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
unloc148trpseofft.ugqky.bid Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
F nduwmw.com Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
2008:1408:ac7:4700:0000:0000:0000:4a21 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
fqqf3ugq7m2z6ou.see0x8top Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
F www.xstar.co Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
216.59.16.175 Intel:ADDR from http://rules.emerginthreets.net/blockrules/emerging-bottc.rules via intel.criticalstack.com F
F 158.49.184.188 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
F 199.111.win Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
haysts.linkpc.net Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
185.228.101.21 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
2001:8648:2ff2:127d::a007:fe03:1841 Intel:ADDR from https://www.dan.me.uk/torlist/ via intel.criticalstack.com F
cicer0-dropbox.tk Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
xiaoyuer001.f332202.org Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
mefbbd.com Intel:DOMAIN from https://dshield.org/feeds/suspiciousdomains_low.txt via intel.criticalstack.com F
192.241.241.94 Intel:ADDR from http://rules.emerginthreets.net/blockrules/emerging-bottc.rules via intel.criticalstack.com F
F freak.chasingyourdream.com Intel:DOMAIN from http://hosts-file.net/exp.txt via intel.criticalstack.com F
```

Figura 4.33: Base de datos de firmas de Critical Stack

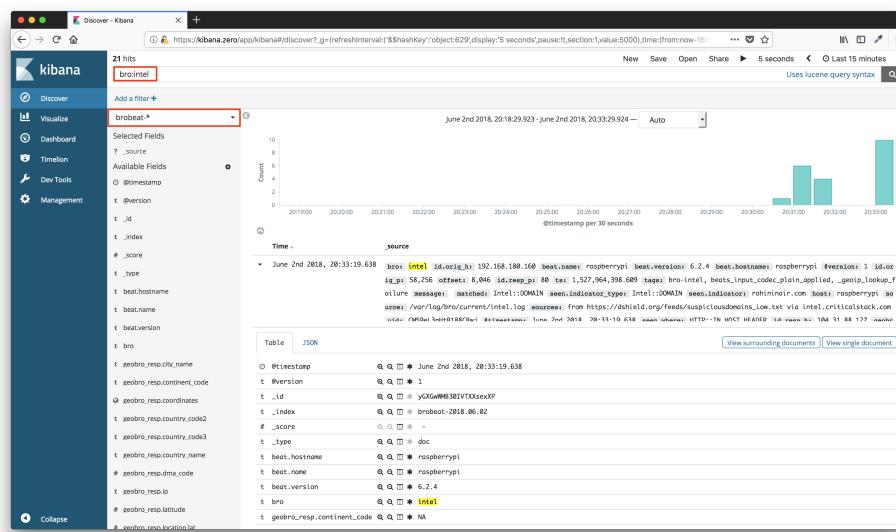


Figura 4.34: Ejemplo de log INTEL recibido en Kibana

`_source` se muestran los detalles del sistema o la sonda que recogió la alerta, útil en el caso de que existan varias desplegadas monitorizando diferentes puntos de la red, en los campos `seen` se encuentran almacenados los detalles del dominio o la dirección IP identificada como insegura y el campo `seen.where` proporciona más detalles del tipo de conexión establecida.

```
{
  "_index": "brobeat-2018.06.02",
  "_type": "doc",
  "_id": "yGXGwWMB30IVTXXsexXP",
  "_score": 1,
  "_source": {
    "id.orig_h": "192.168.180.160",
    "beat": {
      "name": "raspberrypi",
      "version": "6.2.4",
      "hostname": "raspberrypi"
    },
    "@version": "1",
    "id.orig_p": 58256,
    "offset": 8046,
    "id.resp_p": 80,
    "ts": 1527964398.609445,
    "geobro_orig": {},
    "tags": [
      "bro-intel",
      "beats_input_codec_plain_applied",
      "_geoip_lookup_failure"
    ],
    "message": "",
    "matched": [
      "Intel::DOMAIN"
    ],
    "seen.indicator_type": "Intel::DOMAIN",
    "seen.indicator": "rohininair.com",
    "host": "raspberrypi",
    "source": "/var/log/bro/current/intel.log",
    "sources": [
      "from https://dshield.org/feeds/suspiciousdomains_Low.txt via
      intel.criticalstack.com"
    ],
    "uid": "CM59eL3pHtB188CBwj",
    "@timestamp": "2018-06-02T18:33:19.638Z",
    "seen.where": "HTTP::IN_HOST_HEADER",
    "id.resp_h": "104.31.88.127",
    "geobro_resp": {
      "country_code3": "US",
      "location": {
        "lon": -97.822,
        "lat": 37.751
      },
      "coordinates": [
        -97.822,
        37.751
      ],
      "country_code2": "US",
      "latitude": 37.751,
      "ip": "104.31.88.127",
      "longitude": -97.822,
      "country_name": "United States",
    }
  }
}
```

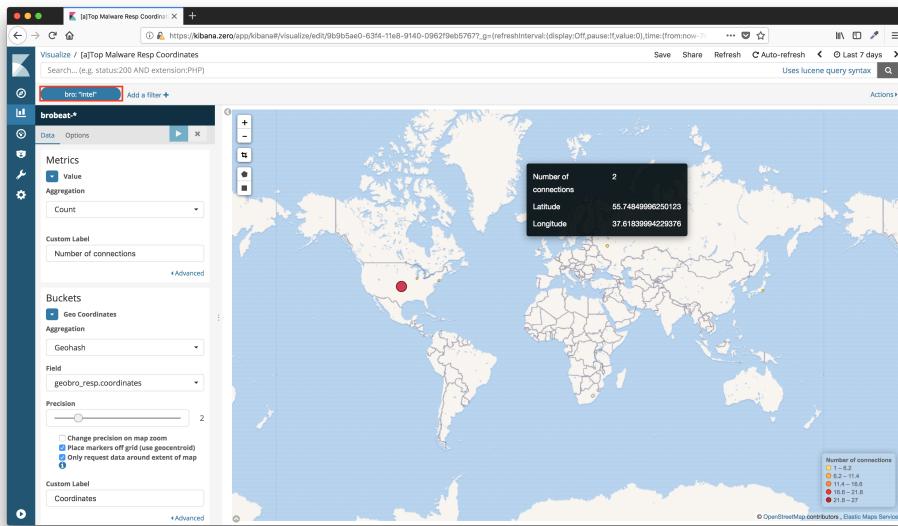


Figura 4.35: Geo posicionamiento por coordenadas de IPs en el log de Critical Stack

```

    "continent_code": "NA"
},
"prospector": {
    "type": "log"
},
"seen.node": "bro",
"bro": "intel"
},
"fields": {
    "@timestamp": [
        "2018-06-02T18:33:19.638Z"
    ]
}
}

```

Script 4.42: Ejemplo de log Intel en formato json

Los valores dentro de los campos *geobro* permiten geo posicionar la dirección IP insegura a la que se ha accedido para posteriormente poder hacer una representación en base a su posición exacta tal y como se muestra en la figura 4.35.

De forma más genérica se puede hacer representando el país, tal y como se ve en la figura 4.36.

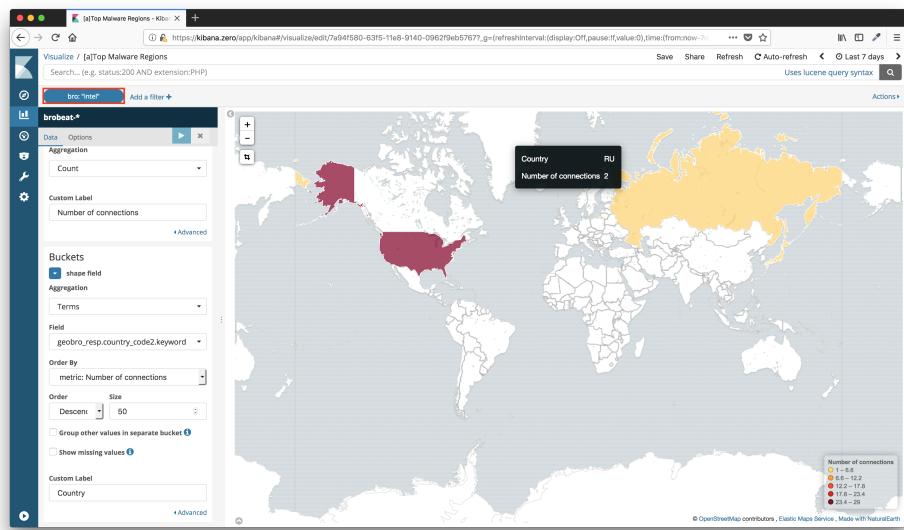


Figura 4.36: Geo posicionamiento por países de IPs en el log de Critical Stack

Logs bro:conn

En el log `/var/log/bro/current/conn.log` se crea una entrada por cada conexión que Bro reconoce. Este log puede ser muy útil a la hora de revisar los destinos u orígenes de las conexiones y en particular para este caso la geo posición de las direcciones IP proporciona una información muy valiosa. Además de la marca de tiempo de cuando se produjo la conexión y los detalles del sistema que generó el log, añade información sobre el protocolo usado, los puertos involucrados en la comunicación y la información de geoposicionamiento. En el script 4.43 se muestra una captura en formato json de una entrada en este log.

```
{  
  "_index": "brobeat-2018.06.03",  
  "_type": "doc",  
  "_id": "h2csxWMB30IVTXXsq5RY",  
  "_score": 1,  
  "_source": {  
    "id.orig_h": "192.168.180.160",  
    "beat": {  
      "name": "raspberrypi",  
      "version": "6.2.4",  
      "hostname": "raspberrypi"  
    },  
    "resp_pkts": 1,  
    "@version": "1",  
    "missed_bytes": 0,  
    "resp_ip_bytes": 87,
```

```
"id.orig_p": 9993,
"offset": 398619,
"id.resp_p": 9993,
"history": "Dd",
"ts": 1528021366.924418,
"duration": 0.076126,
"tunnel_parents": [],
"geobro_orig": {},
"tags": [
    "bro-conn",
    "beats_input_codec_plain_applied",
    "_geoip_lookup_failure"
],
"message": "",
"host": "raspberrypi",
"resp_bytes": 59,
"source": "/var/log/bro/current/conn.log",
"local_resp": false,
"proto": "udp",
"conn_state": "SF",
"uid": "CKZaH74Yoj14AIrPoc",
"orig_pkts": 1,
"@timestamp": "2018-06-03T10:23:47.991Z",
"resp_cc": "FR",
"local_orig": true,
"id.resp_h": "107.191.46.210",
"geobro_resp": {
    "country_code3": "FR",
    "coordinates": [
        2.3548,
        48.9342
    ],
    "latitude": 48.9342,
    "postal_code": "93200",
    "region_name": "Seine-Saint-Denis",
    "timezone": "Europe/Paris",
    "longitude": 2.3548,
    "location": {
        "lon": 2.3548,
        "lat": 48.9342
    },
    "city_name": "Saint-Denis",
    "country_code2": "FR",
    "ip": "107.191.46.210",
    "region_code": "93",
    "country_name": "France",
    "continent_code": "EU"
},
"prospector": {
    "type": "log"
},
"orig_bytes": 137,
"bro": "conn",
"orig_ip_bytes": 165
},
"fields": {
    "@timestamp": [
        "2018-06-03T10:23:47.991Z"
    ]
}
```

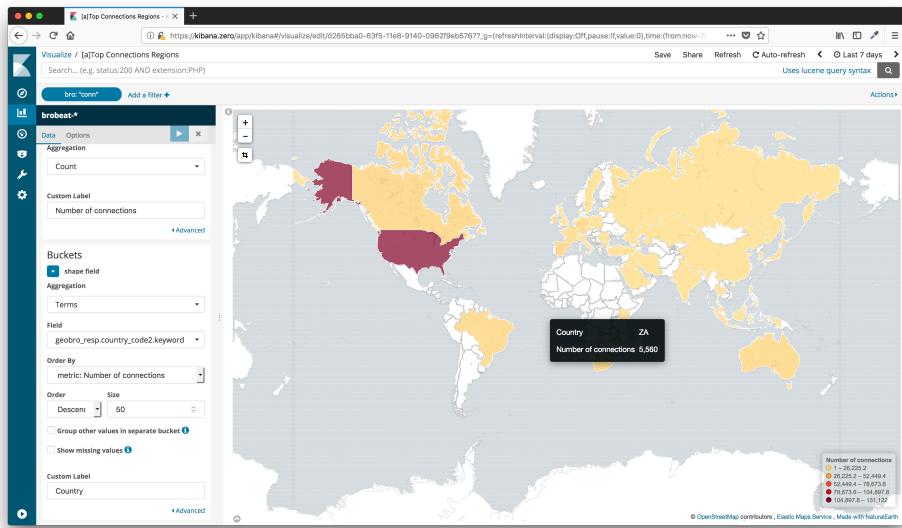


Figura 4.37: Geo posicionamiento por países de IPs en el log de conexiones

Script 4.43: Ejemplo de log conn en formato json

Al igual que en la sección 4.4.3, es posible definir mapas en donde apareza de manera agregada el número de conexiones por países o por posición GPS. Para ver un ejemplo real se visualiza el mapa de la figura 4.37 y se observa que aparece Sudáfrica como un país donde a priori no debiera de haber muchas conexiones. Aunque en el log de Critical Stack no apareciera como inseguro es un comportamiento, a priori, anómalo.

De manera fácil es posible localizar la dirección IP a la que hacen referencia estas conexiones añadiendo dos filtros dentro de menú *Discover* en Kibana y hacer una búsqueda como se muestra en la figura 4.38.

La dirección IP obtenida es *154.66.197.33*. Si se hace una búsqueda en cualquier servicio online de resolución inversa de nombres (figura 4.39) se obtiene que esta dirección IP está relacionada con el dominio *root-alice-joh-01.zerotier.com* por lo que a priori las conexiones a esta dirección IP son legítimas pero podría darse el caso de encontrar un alto número de conexiones a países como China o Rusia que pudieran ser indicios de que hay alguna máquina infectada de malware o que está habiendo accesos remotos que pueden no estar permitidos.

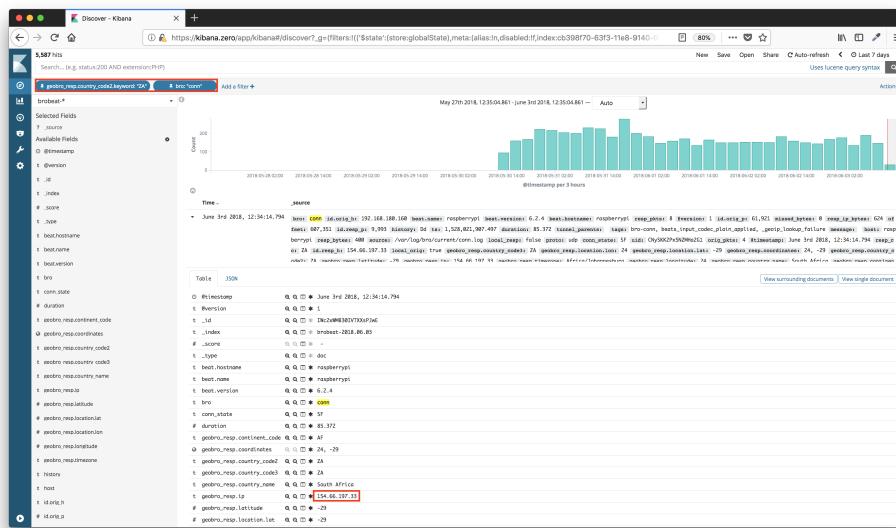


Figura 4.38: Búsqueda de IP por país de origen en el log de conexiones

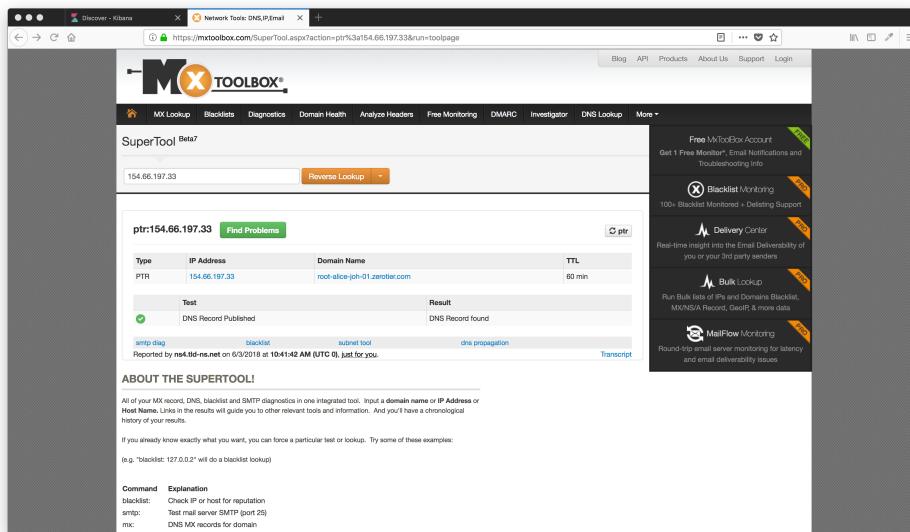


Figura 4.39: Resolución inversa de nombres a partir de dirección IP

Logs bro:files

Otro de los logs que puede ser interesante revisar y crear visualizaciones para facilitar su explotación es el archivo `/var/log/bro/current/files.log` en donde se guarda el registro de los archivos que han sido detectados por Bro y que indica que o se han descargado o se han subido a través de la conexión que se está monitorizando (Script 4.44).

Esto puede ser útil en el ámbito de la seguridad para controlar posibles fugas de información observando si el número de archivos subidos se sale de la media o también revisando si el número de archivos descargados no se corresponde con ningún tipo conocido pues es posible que haya una infección en la red con malware del tipo downloader y que se esté bajando el código malicioso. Una posible visualización para controlar estos escenarios se muestra en la figura 4.40.

```
{
  "_index": "brobeat-2018.06.03",
  "_type": "doc",
  "_id": "imdwxWMB30IVTXXs2Mti",
  "_score": 1,
  "_source": {
    "beat": {
      "name": "raspberrypi",
      "version": "6.2.4",
      "hostname": "raspberrypi"
    },
    "@version": "1",
    "rx_hosts": [
      "192.168.180.160"
    ],
    "seen_bytes": 1028,
    "fuid": "F2q00j20nAk0h2xwn8",
    "offset": 111669,
    "ts": 1528025894.16946,
    "duration": 0,
    "geobro_orig": {
      "country_code3": "US",
      "dma_code": 511,
      "latitude": 39.0481,
      "postal_code": "20149",
      "region_name": "Virginia",
      "timezone": "America/New_York",
      "longitude": -77.4728,
      "location": {
        "lon": -77.4728,
        "lat": 39.0481
      },
      "city_name": "Ashburn",
      "country_code2": "US",
      "ip": "54.209.151.147",
      "region_code": "VA",
      "country_name": "United States",
      "continent_code": "NA"
    },
    "md5": "91de0625abdaf32170cbb25172a8467",
  }
}
```

```

"tags": [
  "bro-files",
  "beats_input_codec_plain_applied",
  "_geoip_lookup_failure"
],
"depth": 0,
"message": "",
"mime_type": "application/pkix-cert",
"host": "raspberrypi",
"sha1": "2796bae63f1801e277261ba0d77770028f20eee4",
"conn_uids": [
  "CwVY2r25714iGg6hI4"
],
"source": "/var/log/bro/current/files.log",
"timedout": false,
"is_orig": false,
"tx_hosts": [
  "54.209.151.147"
],
"analyzers": [
  "MD5",
  "SHA1",
  "X509"
],
"@timestamp": "2018-06-03T11:38:15.946Z",
"local_orig": false,
"missing_bytes": 0,
"geobro_resp": {},
"prospector": {
  "type": "log"
},
"bro": "files",
"overflow_bytes": 0
},
"fields": {
  "@timestamp": [
    "2018-06-03T11:38:15.946Z"
  ]
}
}

```

Script 4.44: Ejemplo de log files en formato json

Logs bro:ssl y bro:http

En el log `/var/log/bro/current/ssl.log`, del que se muestra un ejemplo en 4.45, se recoge información de los certificados detectados por el sistema en el tráfico, el nombre del servidor, la entidad emisora del certificado, la versión del protocolo SSL/TLS usada o entrando en un mayor nivel de detalle la suite de cifrado utilizada.

```
{
  "_index": "brobeat-2018.06.03",
  "_type": "doc",
  "_id": "p2d5xWMB30IVTXXsQdHp",
  "_score": 1,
```

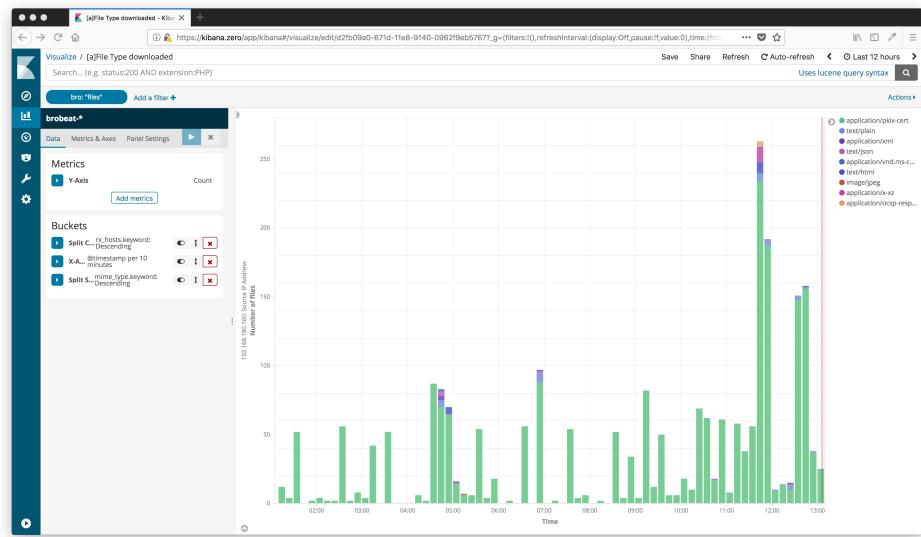


Figura 4.40: Número de archivos identificados agrupados por tipo

```

"_source": {
  "id.orig_h": "192.168.180.160",
  "cert_chain_fuids": [
    "FRMzYGG6MNT2mRyf",
    "Fo2NVp3gIwqMnL810h",
    "Fizoxn1zf9KZbGZikj",
    "FPIDH311yXsxyNQ3C8"
  ],
  "beat": {
    "name": "raspberrypi",
    "version": "6.2.4",
    "hostname": "raspberrypi"
  },
  "@version": "1",
  "id.orig_p": 44782,
  "offset": 79392,
  "id.resp_p": 443,
  "resumed": false,
  "ts": 1528026436.969453,
  "subject": "CN=api.gotinder.com,OU=Domain Control Validated",
  "geobro_orig": {},
  "cipher": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
  "tags": [
    "bro-ssl",
    "beats_input_codec_plain_applied",
    "_geoip_lookup_failure"
  ],
  "message": "",
  "issuer": "CN=Go Daddy Secure Certificate Authority - G2,OU=http://certs.godaddy.com/repository/,O=GoDaddy.com\\, Inc.,L=Scottsdale,ST=Arizona,C=US",
  "host": "raspberrypi",
  "source": "/var/log/bro/current/ssl.log",
  "version": "TLSv12",
  "client_cert_chain_fuids": []
}

```

```
"uid": "CaSpL8LW9YSTGAirf",
"@timestamp": "2018-06-03T11:47:27.992Z",
"id.resp_h": "54.209.151.147",
"established": true,
"geobro_resp": {
  "country_code3": "US",
  "dma_code": 511,
  "coordinates": [
    -77.4728,
    39.0481
  ],
  "latitude": 39.0481,
  "postal_code": "20149",
  "region_name": "Virginia",
  "timezone": "America/New_York",
  "longitude": -77.4728,
  "location": {
    "lon": -77.4728,
    "lat": 39.0481
  },
  "city_name": "Ashburn",
  "country_code2": "US",
  "ip": "54.209.151.147",
  "region_code": "VA",
  "country_name": "United States",
  "continent_code": "NA"
},
"server_name": "api.gotinder.com",
"prospector": {
  "type": "log"
},
"bro": "ssl",
"validation_status": "ok",
"curve": "secp256r1"
},
"fields": {
  "@timestamp": [
    "2018-06-03T11:47:27.992Z"
  ]
}
}
```

Script 4.45: Ejemplo de log ssl en formato json

Esta información es posible explotarla para obtener por ejemplo un diagrama de los principales certificados encontrados junto con su estado para que todos aquellos que por ejemplo no den positivo en la validación de estado se pueda proceder a investigarlos. Se muestra un ejemplo en la figura 4.41.

Otra de las posibles visualizaciones que se pueden hacer relacionadas con la seguridad es una tabla que relacione las direcciones IP origen, destino, el nombre del recurso accedido y el cifrado usado. De esta forma es fácilmente localizable accesos a servicios indebidamente securizados.

De forma similar a lo visto en el caso SSL es posible explotar los logs http (/var/log/bro/current/http.log) en donde siguiendo los ejemplos anteriores

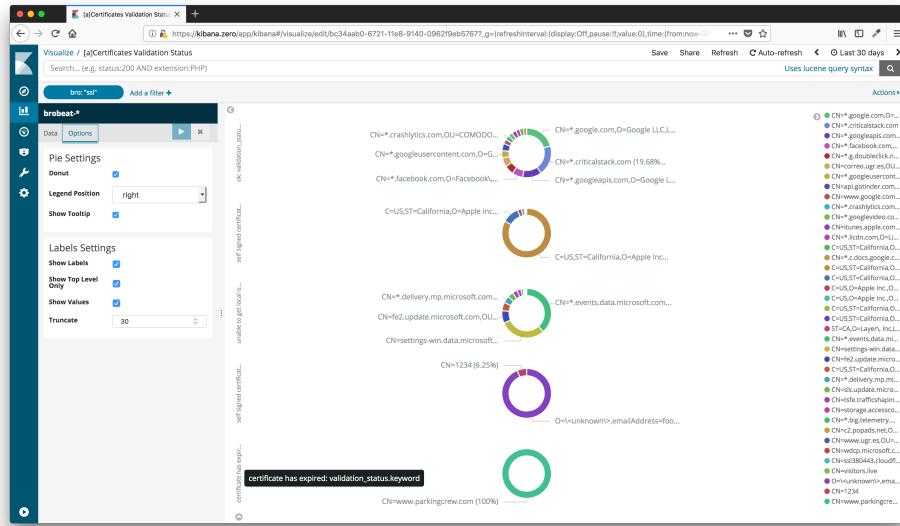


Figura 4.41: Principales certificados detectados agrupados por estado de validación

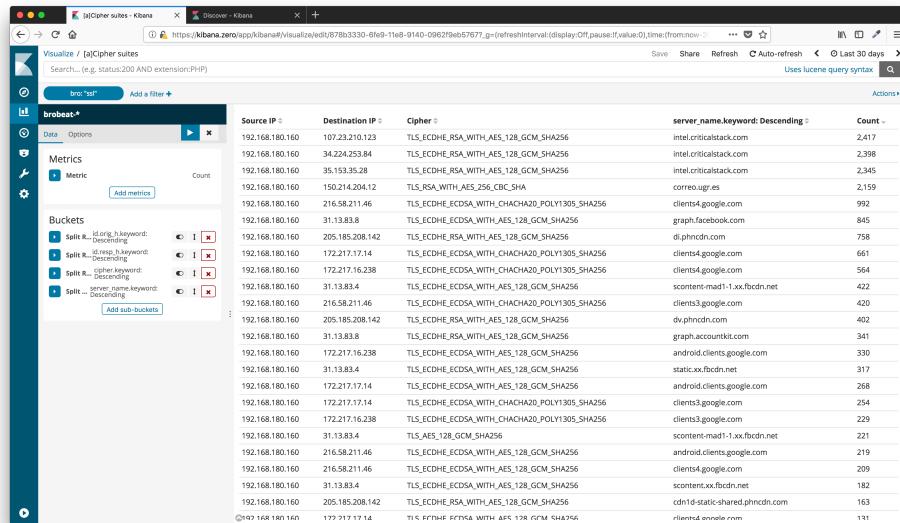


Figura 4.42: Tabla con los cifrados utilizados, recurso accedido y direcciones IP

se podría crear un mapa para el geo posicionamiento como el observado en la figura 4.43 de las direcciones IP con las que hay tráfico http a través del puerto 80.

```
{  
  "_index": "brobeat-2018.06.14",  
  "_type": "doc",  
  "_id": "9ZYI_2MB30IVTXXsDaNs",  
  "_score": 1,  
  "_source": {  
    "id.orig_h": "192.168.180.160",  
    "beat": {  
      "name": "raspberrypi",  
      "version": "6.2.4",  
      "hostname": "raspberrypi"  
    },  
    "status_msg": "No Content",  
    "trans_depth": 1,  
    "@version": "1",  
    "id.orig_p": 51219,  
    "response_body_len": 0,  
    "offset": 2756,  
    "id.resp_p": 80,  
    "ts": 1528992105.959467,  
    "geobro_orig": {},  
    "uri": "/generate_204",  
    "tags": [  
      "bro-http",  
      "beats_input_codec_plain_applied",  
      "_geoip_lookup_failure"  
    ],  
    "message": "",  
    "host": "clients1.google.com",  
    "status_code": 204,  
    "request_body_len": 0,  
    "method": "HEAD",  
    "source": "/var/log/bro/current/http.log",  
    "version": "1.1",  
    "uid": "CCTJxV14GjbWQUIBV8",  
    "@timestamp": "2018-06-14T16:01:46.749Z",  
    "id.resp_h": "172.217.17.14",  
    "geobro_resp": {  
      "country_code3": "US",  
      "dma_code": 807,  
      "coordinates": [  
        -122.0574,  
        37.419200000000004  
      ],  
      "latitude": 37.419200000000004,  
      "postal_code": "94043",  
      "region_name": "California",  
      "timezone": "America/Los_Angeles",  
      "longitude": -122.0574,  
      "location": {  
        "lon": -122.0574,  
        "lat": 37.419200000000004  
      },  
      "city_name": "Mountain View",  
      "country_code2": "US",  
      "ip": "172.217.17.14",  
      "region_code": "CA",  
    }  
  }  
}
```

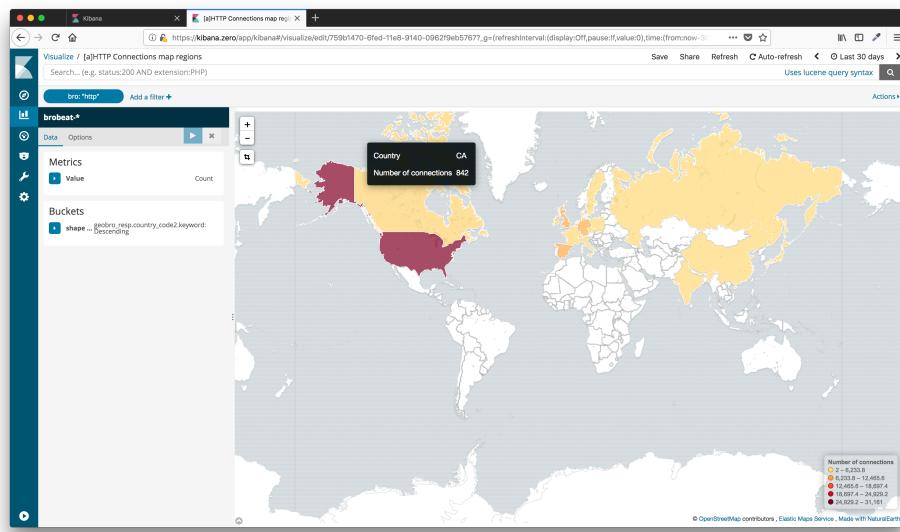


Figura 4.43: Geo posicionamiento por países del tráfico HTTP

```

    "country_name": "United States",
    "continent_code": "NA"
},
"prospector": {
    "type": "log"
},
"bro": "http"
},
"fields": [
    "@timestamp": [
        "2018-06-14T16:01:46.749Z"
    ]
}
}

```

Script 4.46: Ejemplo de log http en formato json

Logs bro:dns

Bro almacena en el archivo de logs `/var/log/bro/current/dns.log` un registro de todas las peticiones DNS que se realizan a servidores que estén fuera de nuestra red. Junto con la información de la query realizada se guarda la dirección IP del servidor al que se le hace la petición, la dirección IP origen y el tipo de registro DNS solicitado. Un ejemplo de estos logs se puede observar en 4.47.

```
{
    "_index": "brobeat-2018.06.14",
```

```
  "_type": "doc",
  "_id": "HZZD_2MB30IVTXXsHu0F",
  "_score": 1,
  "_source": {
    "id.orig_h": "192.168.180.160",
    "AA": false,
    "beat": {
      "name": "raspberrypi",
      "version": "6.2.4",
      "hostname": "raspberrypi"
    },
    "@version": "1",
    "id.orig_p": 54577,
    "qtype": 1,
    "RA": true,
    "offset": 86384,
    "id.resp_p": 53,
    "ts": 1528995976.379448,
    "RD": true,
    "geobro_orig": {},
    "qclass_name": "C_INTERNET",
    "qclass": 1,
    "query": "www.youtube.com",
    "tags": [
      "bro-dns",
      "beats_input_codec_plain_applied",
      "_geoip_lookup_failure"
    ],
    "message": "",
    "TTLs": [
      21565,
      265,
      265,
      265,
      265,
      265,
      265,
      265
    ],
    "host": "raspberrypi",
    "rejected": false,
    "rcode_name": "NOERROR",
    "trans_id": 38274,
    "source": "/var/log/bro/current/dns.log",
    "answers": [
      "youtube-ui.l.google.com",
      "216.58.201.142",
      "172.217.17.14",
      "172.217.168.174",
      "216.58.211.206",
      "172.217.16.238",
      "216.58.211.46",
      "216.58.210.142"
    ],
    "proto": "udp",
    "rtt": 0.059968,
    "TC": false,
    "uid": "CGH1jK1vGtktCN7Mg",
    "@timestamp": "2018-06-14T17:06:17.601Z",
    "id.resp_h": "8.8.8.8",
    "geobro_resp": {
      "country_code3": "US",
      "lat": 37.7749,
      "lon": -122.4194
    }
  }
}
```

```

    "location": {
      "lon": -97.822,
      "lat": 37.751
    },
    "coordinates": [
      -97.822,
      37.751
    ],
    "country_code2": "US",
    "latitude": 37.751,
    "ip": "8.8.8.8",
    "longitude": -97.822,
    "country_name": "United States",
    "continent_code": "NA"
  },
  "prospector": {
    "type": "log"
  },
  "rcode": 0,
  "qtype_name": "A",
  "Z": 0,
  "bro": "dns"
},
"fields": {
  "@timestamp": [
    "2018-06-14T17:06:17.601Z"
  ]
}
}

```

Script 4.47: Ejemplo de log dns en formato json

Dos de las posibles visualizaciones que pueden configurarse se muestran en las figuras 4.44 y 4.45. En ellas se podría ver si una determinada dirección IP origen está haciendo conexiones a dominios sospechosos.

Logs bro:stats

El último de los logs a analizar para poder extraer información relativa a la seguridad es el de estadísticas. Se encuentra en `/var/log/bro/current/stats.log` y contiene información del número de conexiones activas, el número de paquetes transmitidos, recibidos, el número de bytes, etc. Se muestra su estructura y un ejemplo en 4.48.

```

{
  "_index": "brobeat-2018.06.14",
  "_type": "doc",
  "_id": "0Zdr_2MB30IVTXXs2RuP",
  "_score": 1,
  "_source": {
    "reassem_unknown_size": 0,
    "beat": {
      "name": "raspberrypi",
      "version": "6.2.4",
      "hostname": "raspberrypi"
    },

```

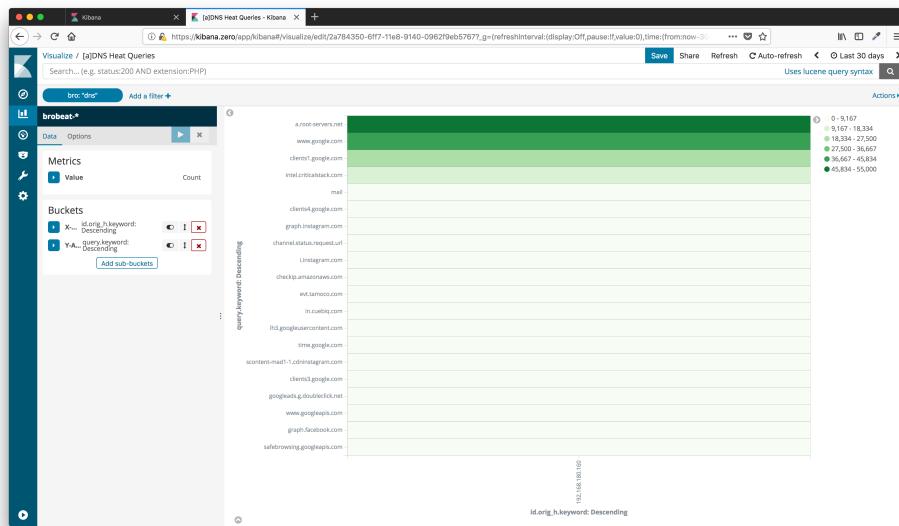


Figura 4.44: Visualización Heat Zones para el log DNS

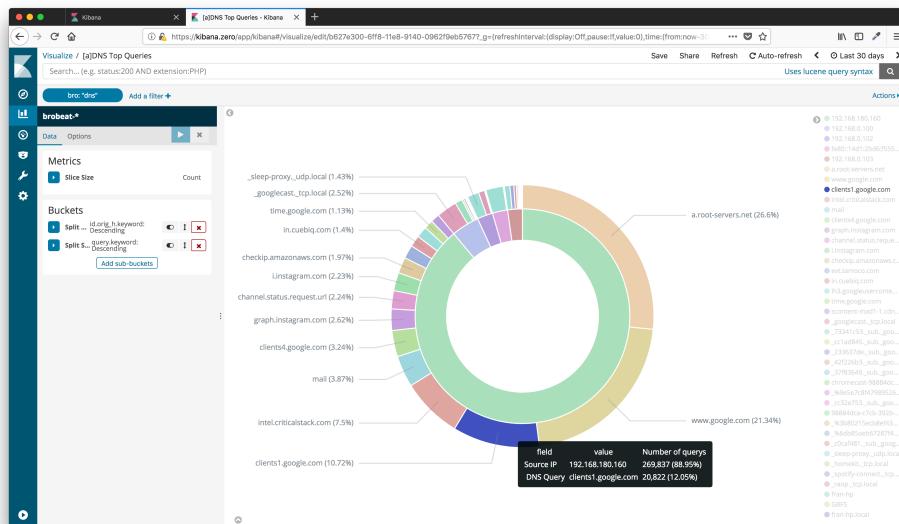


Figura 4.45: Visualización de mayores consultas DNS

```

"active_dns_requests": 0,
"@version": "1",
"mem": 235,
"offset": 2442,
"events_queued": 30312,
"active_files": 0,
"ts": 1528998638.381531,
"dns_requests": 0,
"tags": [
    "bro-stats",
    "beats_input_codec_plain_applied"
],
"message": "",
"pkt_lag": 0.000782,
"files": 73,
"host": "raspberrypi",
"reassem_frag_size": 0,
"timers": 5824,
"pkts_proc": 36704,
"source": "/var/log/bro/current/stats.log",
"active_udp_conns": 126,
"pkts_dropped": 0,
"active_timers": 335,
"@timestamp": "2018-06-14T17:50:47.043Z",
"peer": "bro",
"udp_conns": 387,
"tcp_conns": 112,
"bytes_recv": 28462051,
"icmp_conns": 10,
"prospector": {
    "type": "log"
},
"active_tcp_conns": 58,
"reassem_tcp_size": 2528,
"pkts_link": 36711,
"active_icmp_conns": 4,
"reassem_file_size": 0,
"bro": "stats",
"events_proc": 30309
},
"fields": {
    "@timestamp": [
        "2018-06-14T17:50:47.043Z"
    ]
}
}

```

Script 4.48: Ejemplo de log stats en formato json

Un posible uso de este log relativo al campo de la seguridad es la visualización del número de conexiones en función del tiempo, como la mostrada en la figura 4.46 permitiendo crear una línea base para que si se observa alguna anomalía se pueda proceder a su análisis.

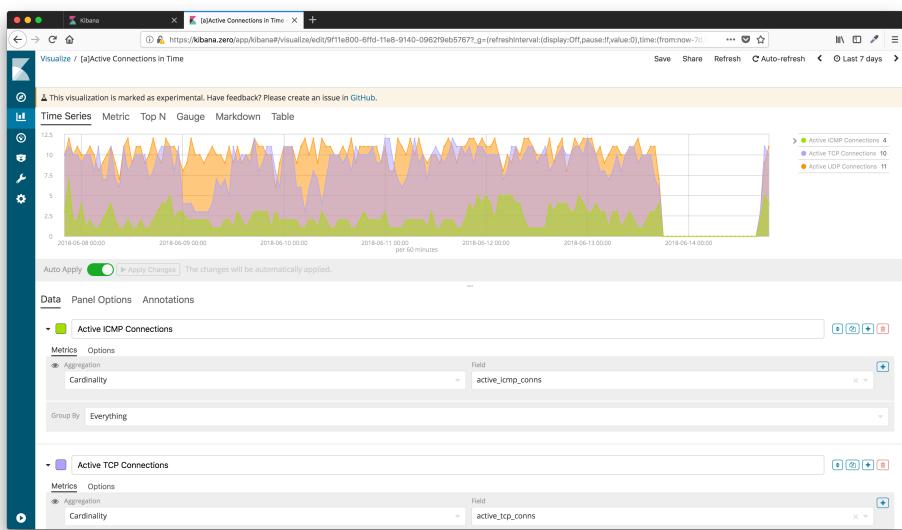


Figura 4.46: Línea base de conexiones activas

Capítulo 5

Conclusiones y Trabajo Futuro

En este capítulo se detallan los logros y resultados obtenidos tras la finalización del proyecto, analizando las aportaciones de éste y el cumplimiento de los objetivos y requisitos establecidos en los capítulos anteriores. Además se hace una exposición de las posibles líneas de trabajo futuras surgidas para la mejora del sistema desarrollado. Para finalizar se ha incluido un apartado en el que se incluyen algunas reflexiones personales relativas a la realización del proyecto.

5.1. Conclusiones

En el presente trabajo fin de máster se ha hecho un repaso acerca de la importancia de los sistemas de monitorización dado el incremento de los ataques que se producen cada día. También se ha hecho una revisión de las principales herramientas para este fin basadas en código abierto y de los frameworks de desarrollo.

Tras la fase de planificación se ha realizado el diseño del sistema con todas las funcionalidades que deberá de incluir el sistema entre las que se encuentran la recogida de capturas de los paquetes de red, análisis de malware basado en reglas YARA y comprobación del tráfico observado con las reglas y feeds que se seleccionen en CriticalStack para detección de conexiones maliciosas.

De manera extra se ha implementado un servidor con Logstash, Elasticsearch y Kibana para la recogida de todos los logs, tanto del sistema como los generados por Bro, como motor IDS y unos ejemplos acerca de la posible explotación de estos logs desde el punto de vista de la seguridad.

Como resumen de este trabajo, se han completado las fases planificadas desarrollando un sistema portable de detección de intrusiones, fácilmente replicable y con un motor de análisis de logs muy potente.

5.2. Líneas de trabajo futuro

Durante la realización de este trabajo fin de máster se ha desarrollado un sistema autónomo de detección de intrusiones sobre una Raspberry Pi, dada el constante aumento de las amenazas ciberneticas es posible completar y añadir funcionalidades extras al sistema que complementen las funcionalidades incluidas.

Una de las posibles herramientas adicionales a instalar podría ser Snort para dotar aún de mayor inteligencia al sistema en la detección de intrusiones a través de la correlación de sus reglas con los indicadores de compromiso que se le configuren y el tráfico analizado.

Para dotar de mayor fluidez a la interfaz gráfica sería posible migrarla del escenario actual híbrido desarrollado entre Blade de Laravel y AngularJS a un front-end 100 % desarrollado con AngularJS y además dotar de mecanismos de reconfiguración del sistema desde la interfaz web para los usuarios de administración (conexión wifi, iniciar y parar servicios, cambiar el espacio reservado para almacenar ficheros, modificación del hostname, ...).

Dada la relativa complejidad que puede tener para usuarios domésticos reconfigurar la red para crear un único punto por el que pase todo el tráfico para crear el port mirroring se estima difícil poder desarrollar un sistema comercial para estos entornos. La solución desarrollada tiene un mayor potencial para los entornos de pequeñas y medianas empresas y en donde además se podría proporcionar un servicio integral de seguridad con la instalación de la sonda, la reconfiguración de la electrónica de red y en donde se produzca el envío de los logs a un sistema centralizado. El equipo del SOC se encargará del envío de alertas al cliente y de la posible resolución de las mismas.

En la parte software del sistema, tanto el instalado como al desarrollado específicamente, una gran línea de trabajo consiste tanto en el hardening de todo el entorno a través del ajuste de las configuraciones que se han dejado por defecto como una revisión de seguridad del software desarrollado como del middleware y framework utilizado para tal fin. Además, revisando la interfaz web sería posible añadir funcionalidades extras como un sistema de gestión de usuarios completo, integración con LDAP, etc.

Sería también posible añadir funcionalidad extra al servidor de recolección y análisis de logs añadiendo el paquete X-Pack que permite definir reglas para el envío de alertas y la adición de herramientas de reporting y

machine learning.

5.3. Valoración personal

Dentro de la valoración personal y autocrítica que realicé tras la culminación de este proyecto debo de reconocer que el grado de satisfacción es muy alto debido a que se ha logrado completar con éxito todos los objetivos marcados al inicio del mismo y a los que además se ha añadido toda una capa extra de gestión de logs basada en ELK como punto adicional.

Después de un largo periodo de trabajo, de recopilación de información, de investigación sobre herramientas, de pruebas y ajustes de configuración y de desarrollo, este trabajo fin de master me ha ayudado a adquirir una gran cantidad de conocimientos y me ha permitido poner en práctica una gran cantidad de áreas y conocimientos adquiridos durante la realización de este máster.

Bibliografía

- [1] Portaltic/EP. (mayo de 2017). Día de Internet.- Los cambios sociales que Internet ha provocado y la importancia de la seguridad, dirección: <http://www.europapress.es/portaltic/internet/noticia-dia-internet-cambios-sociales-internet-provocado-importancia-seguridad-20110517093046.html>.
- [2] L. González. (). La importancia de la ciberseguridad, dirección: <http://www.metropoliscom.com/la-importancia-de-la-ciberseguridad/>.
- [3] J. Benítez. (mayo de 2017). España, tercer país del mundo con más ciberataques, dirección: <http://www.elmundo.es/españa/2017/05/15/5918ae9222601d51718b46d7.html>.
- [4] M. Puente García. (dic. de 2017). Riesgos y retos de ciberseguridad y privacidad en IoT, dirección: <https://www.certsi.es/blog/riesgos-y-retos-ciberseguridad-y-privacidad-iot>.
- [5] C. C. Nacional. (2016). CCN-CERT BP-01/16 Principios y recomendaciones básicas en ciberseguridad, dirección: <https://www.ccn-cert.cni.es/informes/informes-de-buenas-practicas-bp/2473-ccn-cert-bp-01-16-principios-y-recomendaciones-basicas-en-ciberseguridad/file.html>.
- [6] E. J. Mira Alfaro. (). Proyecto Final de Carrera: Implementación de un Sistema de Detección de Intrusos en la Universidad de Valencia, dirección: <https://www.rediris.es/cert/doc/unixsec/unixsec.pdf>.
- [7] A. Santana. (jun. de 2015). Violación a sistemas informáticos. Hackers y Crackers, dirección: <http://andressantana.260mb.net/violacion-a-sistemas-informaticos-hackers-y-crackers/?i=1>.
- [8] A. Villalón Huerta. (jul. de 2002). Seguridad en UNIX y Redes, dirección: <https://www.rediris.es/cert/doc/unixsec/unixsec.pdf>.
- [9] Snort. (), dirección: <https://www.snort.org>.
- [10] Wikipedia. (abr. de 2016). MartinRoesch, dirección: https://en.wikipedia.org/wiki/Martin_Roesch.

- [11] V. Truica. (mar. de 2014). Understanding the Snort architecture, dirección: <https://truica-victor.com/snort-architecture/>.
- [12] Suricata. (), dirección: <https://suricata-ids.org/>.
- [13] O. I. S. Foundation. (), dirección: <https://oisf.net>.
- [14] Bro. (), dirección: <https://www.bro.org>.
- [15] Wikipedia. (jun. de 2017). Vern Paxson, dirección: https://en.wikipedia.org/wiki/Vern_Paxson.
- [16] CodeIgniter. (), dirección: <https://codeigniter.com/>.
- [17] Laravel. (), dirección: <https://laravel.com/>.
- [18] AngularJS. (), dirección: <https://angularjs.org/>.
- [19] Wikipedia. (dic. de 2017). AngularJS, dirección: <https://es.wikipedia.org/wiki/AngularJS>.
- [20] ReactJS. (), dirección: <https://reactjs.org/>.
- [21] Wikipedia. (feb. de 2018). ReactJS, dirección: <https://es.wikipedia.org/wiki/React>.
- [22] Á. Luis. (mar. de 2014). Características arquitectura 64 bits ARMv8-A para tablets y smartphones, dirección: <http://alsitecno.com/2014/03/28/caracteristicas-arquitectura-64-bits-armv8-a-para-tablets-y-smartphones/>.
- [23] Wikipedia. (ene. de 2018). Raspbian, dirección: <https://en.wikipedia.org/wiki/Raspbian>.
- [24] DevExpress. (). HTML5 JavaScript Component Suite for Responsive Web Development, dirección: <https://js.devexpress.com/>.
- [25] R. P. Foundation. (abr. de 2018). Raspbian Stretch Lite, dirección: <https://www.raspberrypi.org/downloads/raspbian>.
- [26] S. Association. (). SD Memory Card Formatter, dirección: https://www.sdcard.org/downloads/formatter_4/.
- [27] t. gruemaster tuxinator2009. (). Win32 Disk Imager, dirección: <https://sourceforge.net/projects/win32diskimager/>.
- [28] ZeroTier. (). ZeroTier, dirección: <https://www.zerotier.com>.
- [29] g. sneakymonk3y gebhard73. (). foxhound-nsm, dirección: <https://github.com/sneakymonk3y/foxhound-nsm>.
- [30] Intel. (). Critical Stack, dirección: <https://intel.criticalstack.com/apis>.
- [31] elastic. (), dirección: <https://discuss.elastic.co/t/how-to-install-filebeat-on-a-arm-based-sbc-eg-raspberry-pi-3/103670/3>.

- [32] dam90. (). pibeats, dirección: https://github.com/dam90/pibeats/blob/master/build_script.sh.
- [33] N. Congleton. (). Install ELK On Ubuntu 18.04 Bionic Beaver Linux, dirección: <https://linuxconfig.org/install-elk-on-ubuntu-18-04-bionic-beaver-linux>.

