Seguridad en el Internet de las Cosas (IoT)

Implementación de IoT Honeypot y análisis de resultados

Alumno: Fernando Javier Fernández Aparicio

Trabajo Final de Grado: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Área: Administración de redes y sistemas operativos

Consultor: Miguel Martín Mateo

Fecha: Enero de 2024

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mi familia por su apoyo incondicional, paciencia y comprensión a lo largo de este proyecto. Su amor y aliento constante fueron fundamentales en cada etapa de este camino académico.

Agradezco enormemente a mis compañeros de estudio y al consultor por sus valiosas contribuciones, discusiones enriquecedoras y por compartir sus conocimientos. Su colaboración fue esencial para enriquecer este proyecto con diferentes perspectivas y enfoques.

Además, agradezco a mis amigos Taru, Moi y Raúl por brindarme su apoyo en todo momento. Han sido fundamentales la información, recursos y opiniones que me han proporcionado para ayudarme a encaminar este proyecto. Su generosidad y disposición para compartir conocimientos fueron de gran ayuda para alcanzar mis objetivos.

Resumen

Es innegable el rápido crecimiento en la utilización de dispositivos IoT en multitud de áreas a nivel global, desde los hogares a grandes industrias de todo ámbito. Debido en gran medida a lo económico de su fabricación, los medidas de seguridad que implementan, a menudo, son insuficientes.

Esto los convierte en uno de los principales objetivos por parte de los ciberdelincuentes, los cuales encuentran en estos dispositivos, un fácil sistema que capturar y poner a trabajar bajo sus órdenes ya sea capturando información, minando criptomonedas o incluso realizando ataques coordinados.

El proyecto se enfoca en la implementación de un *Honeypot* que simula un dispositivo IoT para analizar y registrar patrones de ataques cibernéticos. Se estudian diferentes tipos de *Honeypots* y se elige la herramienta Cowrie para su despliegue, configuración y monitoreo. El objetivo principal es recolectar datos sobre amenazas potenciales que afectan a dispositivos IoT y estudiar el comportamiento de los atacantes para identificar vulnerabilidades. Se establece una arquitectura de seguridad proactiva para mitigar riesgos y se propone medidas para fortalecer la seguridad en dispositivos IoT.

La salida del sistema trampa se conecta a la herramienta Elastic Stack, capaz de buscar, analizar, y visualizar registros generados desde cualquier fuente y en cualquier formato, una práctica conocida como registro centralizado.

Finalmente, se estudia toda la información recogida para observar los principales métodos de ataque, el origen de estos y las fases en las que los atacantes los estructuran para lograr mayor efectividad.

Abstract

There is no denying the rapid growth in the use of IoT devices in a multitude of areas globally, from homes to large industries across the board. Largely due to the cheapness of their manufacture, the security measures they implement are often insufficient.

This makes them one of the one of the main targets for cybercriminals, who find these devices an easy target which find in these devices an easy system to capture and put to work under their command either by capturing information, mining cryptocurrencies or even carrying out coordinated attacks.

The project is focused on the implementation of a Honeypot that simulates an IoT device to analyze and record cyber attack patterns. Different types of Honeypots are studied and the Cowrie tool is chosen for its deployment, configuration and monitoring. The main objective is to collect data on potential threats affecting IoT devices and study the behavior of attackers to identify vulnerabilities. A proactive security architecture is established to mitigate risks and proposes measures to strengthen security in IoT devices.

The output of the trap system is connected to the Elastic Stack tool, capable of searching, analyzing, and visualizing logs generated from any source and in any format, a practice known as centralized logging.

Finally, all the information collected is studied to observe the main attack methods, their origin and the phases in which the attackers structure them to achieve greater effectiveness.

Contenido

Índice de figuras

Capitulo 1. Introducción

1.1. Descripción del trabajo

Este proyecto se ubica dentro del Área de Administración de Redes y Sistemas Operativos en la rama de ciberseguridad. Más concretamente, aborda la seguridad de los equipos de usuario IoT desde una perspectiva de investigación. La investigación se llevará a cabo mediante un experimento, con el posterior análisis de los datos obtenidos.

Dicho experimento consistirá en la configuración y despliegue de un *HoneyPot* o sistema trampa para exponerlo en Internet durante un tiempo. Se recabarán datos durante ese periodo para analizar los ataques recibidos y llegar a conclusiones.

1.2. Justificación del trabajo

¿Por qué analizar la seguridad de los dispositivos IoT?

En el Grado en Ingeniería de Telecomunicaciones nos hemos encontrado en varias materias con dispositivos de este tipo, ya que, sus posibilidades son muchísimas y amplían las posibilidades de este sector enormemente. En la última década, hemos sido testigos de un aumento vertiginoso en la adopción de dispositivos IoT en hogares, empresas e industrias. Esto ha creado un ecosistema de dispositivos altamente interconectados que juegan un papel crucial en nuestra vida cotidiana y en el mundo empresarial ya que, en esencia, es su objetivo. Se estima que su uso de cara al futuro esté aún más extendido y se adopte en aplicaciones críticas como sistemas de energía, transporte y salud.

Los dispositivos IoT desde el comienzo de su utilización son claro objetivo para los ciberdelincuentes. Esto es debido a que la conectividad a Internet de los dispositivos IoT es su principal fuerte, pero también su punto débil. En términos generales, estos dispositivos son más económicos y cuentan con *hardware* y *sofware* más simple del habitual, además que las actualizaciones son escasas o difíciles para el usuario. Por ello, en ocasiones se da el caso de que el dispositivo cuenta con configuraciones de seguridad deficientes o vulnerabilidades, así como contraseñas débiles o por defecto.

Por ello, en ocasiones los dispositivos IoT son utilizados por los ciberdelincuentes para formar una *botnet*, y así llevar a cabo otro tipo de ataques, como envío de *spam*, lanzamiento de ataques de denegación distribuida de servicio o DDoS, distribución de *malware*, etc. Además, los dispositivos IoT pueden poner en riesgo la privacidad de los usuarios ya que en ocasiones gestionan multitud de información de carácter confidencial.

¿Que herramienta puedo usar para obtener información sobre ataques?

Para recabar información sobre el comportamiento de los ciberdelincuentes y sus métodos es necesaria una potente herramienta con la que simular un sistema que, una

vez vulnerado, se encargará de monitorear o registrar la actividad de los intrusos. Este tipo de herramientas existen en muchas variantes y son conocidas como *Honeypot* o “sistema trampa”.

Como ya hemos adelantado, un *Honeypot*, se ubica en una red o sistema informático y su función principal es detectar y obtener información del ataque informático, incluso de su origen. En la actualidad, los *Honeypots* se han convertido en poderosas herramientas que nos brindan la capacidad de emular el comportamiento auténtico de sistemas, engañando a los ciberatacantes haciéndoles creer que han accedido a un sistema real y que es fácil tomar el control. No obstante, estos ciberdelincuentes operan en un entorno aislado, lo que nos permite supervisar con precisión sus actividades y las vulnerabilidades que intentan explotar.

Los *Honeypots*, incluyendo los que simulan dispositivos IoT, son esenciales en la detección temprana de amenazas cibernéticas y proporcionan información valiosa para el fortalecimiento de la seguridad en un entorno tecnológico en constante evolución. Este proyecto contribuirá a la comprensión de las amenazas que acechan a estos dispositivos y a la formulación de estrategias para protegerlos de manera efectiva.

Estas herramientas pueden estar diseñadas y programadas con diferentes y múltiples objetivos: detectar ataques, obtener información de los ataques o ralentizarlos. Por supuesto, estos objetivos pueden ser combinados en una misma herramienta adaptándolos al escenario y las necesidades.

1.3. Objetivo del trabajo

El principal objetivo de este proyecto es el de registrar patrones de ataque realizados por ciberdelincuentes, de cara a implementar mejoras en la configuración de seguridad de un dispositivo IoT que mitiguen o eliminen amenazas y vulnerabilidades existentes.

No obstante, deben de tenerse en cuenta y cumplirse los siguientes objetivos secundarios para conseguir dicho objetivo principal:

- Análisis y estudio de los diferentes *HoneyPots* existentes. Se valorarán las diferentes posibilidades aplicables a nuestro caso concreto.
- Análisis y estudio de los servicios a emular que pueden ser susceptibles de ataque.
- Configuración y despliegue de un *HoneyPot*. También de herramientas de análisis.
- Capturar y monitorizar los patrones de ataque registrados en el *HoneyPot*.
- Analizar la información y archivos obtenidos del *HoneyPot* con herramientas dedicadas.
- Proponer medidas de seguridad recomendables en la configuración de un dispositivo IoT para eliminar o mitigar riesgos y vulnerabilidades.

Estos objetivos puede variar dependiendo de las dificultades presentadas incluso pudiéndose ampliar.

1.4. Requisitos

Para llevar a cabo el proyecto, sobretodo la parte de implementación del *HoneyPot* y los componentes y/o herramientas de análisis será necesario contar con diferentes requisitos:

Software:

- *Sistema operativo:* En nuestro caso trabajaremos desde un ordenador con SO Ubuntu 22.04.
- *Herramientas SW:* Cowrie, VirusTotal, Termius (iOS app), Censys y Maxmind (GeoIP). Firewall Iptables y firmware propio del router.
- *Otras herramientas:* Gantt Project para el diagrama de Gantt. Elasticsearch, Logstash y Kibana para el tratamiento y visualización de datos.

Hardware:

- *Infraestructura*: Se usará un ordenador portátil Dell Inc Latitude E5420 con 10GB de RAM y 250 GB de disco duro.
- *Elementos de comunicación:* Tarjeta de red wifi Intel Corporation Centrino Advanced-N 6205 [Taylor Peak] y enrutador modelo ZTE ZXHN H298Q V7.0 instalado por la compañía Digi para clientes particulares.

Uso:

- *Afectación a los servicios:* La utilización simultánea de los diferentes elementos puede afectar notablemente al uso del equipo para otras tareas. Por ello, el equipo permanecerá ejecutando exclusivamente los programas indicados durante las pruebas y puesta en funcionamiento del sistema.

Normativa Legislativa, Seguridad y Privacidad:

- *Acceso autorizado:* Las herramientas utilizadas en el proyecto se encuentran en una red propia y con permiso para realizar pruebas.
- *Cumplimiento legal:* Compromiso de operar con el *Honeypot* de manera ética y dentro de los límites legales evitando cualquier tipo de actividad ilegal.
- *Políticas de seguridad:* Se emplearán mecanismos de firewall y DMZ para evitar cualquier acceso a la red personal.

1.5. Planificación

Son múltiples las tareas del proyecto para la consecución de los diferentes objetivos expuestos anteriormente:

• FASE 0

Elaboración de propuesta para TFG.

• FASE 1

Análisis y estudio de los diferentes *HoneyPots* existentes.

• FASE 2

Configuración y despliegue de un *HoneyPot*.

• FASE 3

Capturar, monitorizar de patrones de ataque recibidos.

• FASE 4

Conclusiones y propuesta de medidas de seguridad que fortalezcan los dispositivos IoT.

• FASE 5

Preparación de memoria y presentación.

A continuación se exponen en forma de diagrama de Gantt detallando la planificación completa.

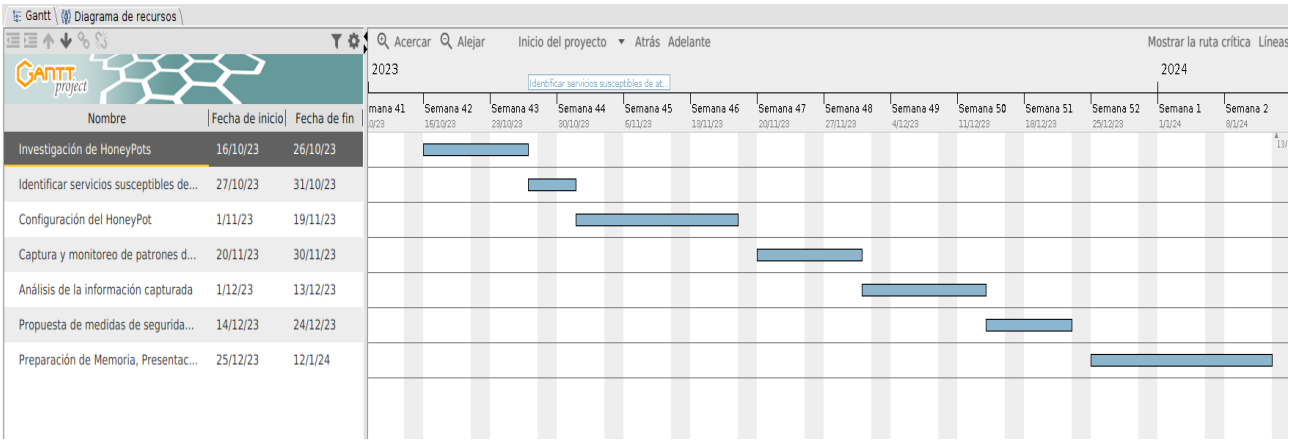


Figura 1. Diagrama de Gantt

1.6. Productos obtenidos

- Identificación del comportamiento de un atacante luego de obtener acceso al dispositivo y determinar si es un ataque robotizado o uno realizado por una mano humana.
- Identificación de los vectores utilizados por los delincuentes para tomar posesión de un dispositivo expuesto en internet.
- Identificación de las vulnerabilidades explotadas por un delincuente o robot para realizar un ataque.
- Informe de ataques realizados y vectores utilizados por los delincuentes para realizar las actividades maliciosas.

1.7. Breve descripción de otros capítulos de la memoria

➢ Capitulo 2. Estado del Arte

Se realiza una exposición del estado del arte actual de los dispositivos IoT y sus vulnerabilidades. Se profundiza en el concepto de Honeypot, los tipos que existen, para qué entornos sirven, sus características, dónde se ubican y cuáles son sus ventajas y desventajas.

➢ Capítulo 3. Herramientas

Se presentan herramientas que se utilizan para el desarrollo del proyecto.

➢ Capitulo 4. Infraestructura e implementación

Se presenta la infraestructura disponible y la implementación en detalle que se llevará a cabo para la ejecución del proyecto.

➢ Capitulo 5. Resultados obtenidos

Se realiza una exposición de los resultados obtenidos. Para aspectos con mucha información, se indicarán los datos más relevantes o destacados.

➢ Capitulo 6. Conclusiones

Se exponen las conclusiones a partir del desarrollo y resultados del proyecto y las medidas de seguridad que hay que llevar a cabo para proteger los dispositivos IoT.

➢ Glosario, fuentes y bibliografía

Términos usados en la memoria y las fuentes consultadas.

Capitulo 2. Estado del Arte

2.1. IoT (Internet de las Cosas)

El Internet de las Cosas (IoT) es un concepto revolucionario que ha transformado la forma en que interactuamos con el mundo digital y físico. En su esencia, el IoT se trata de la interconexión de objetos cotidianos y dispositivos, permitiéndoles comunicarse, recopilar datos y realizar acciones de manera inteligente. Esta interconexión se logra mediante sensores, software y tecnología de conectividad que permiten que estos objetos se conviertan en componentes activos de una red global y, lo más importante, no requieren de interacción directa por parte de los humanos una vez puestos en funcionamiento.

> *"El Internet de las cosas se ha considerado un término erróneo porque los dispositivos no necesitan estar conectados a la Internet pública. Sólo necesitan estar conectadas a una red y ser direccionables individualmente." fuente: Wikipedia.*

Este ecosistema se forma por los dispositivos IoT -que van desde *gadgets* personales y sensores industriales hasta vehículos conectados y edificios inteligentes- están diseñados para recopilar información en tiempo real y comunicarla a sistemas de control, aplicaciones y otros dispositivos IoT. Así, en combinación con otras tecnologías como el almacenamiento Cloud, el 5G, la Inteligencia Artifical o herramientas de análisis Big Data permiten aprovechar estos datos para mejorar la eficiencia, la productividad y la toma de decisiones, al tiempo que permite la automatización y la interconexión de dispositivos y sistemas para lograr un mayor control y gestión más eficiente.[1][2]

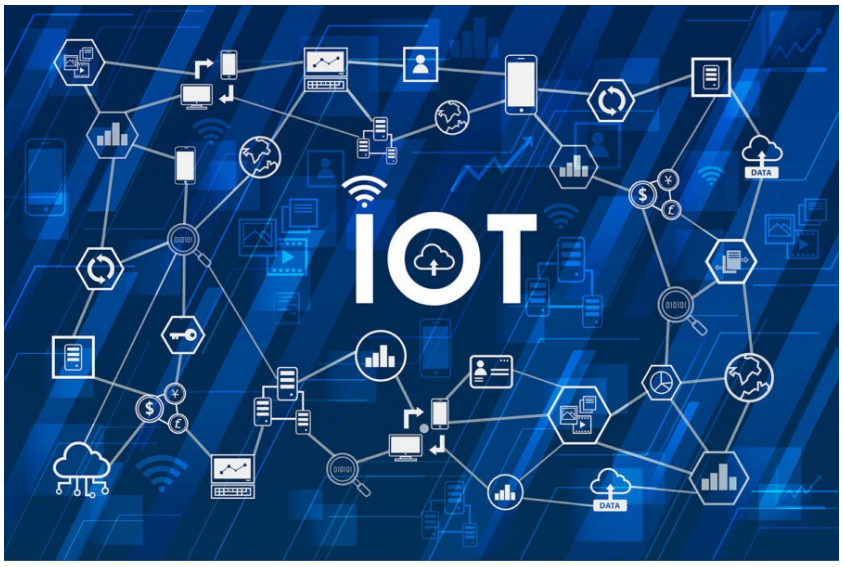


Figura 2. Ecosistema IoT

Los ingenieros en informática llevan agregando sensores y procesadores a los objetos cotidianos desde los años 90. Sin embargo, el progreso fue inicialmente lento porque los chips eran grandes y voluminosos [3]. Sin embargo, en los últimos años, este campo ha evolucionado gracias a la convergencia de múltiples tecnologías:

- Informática ubicua: Integración de la informática en el entorno de la persona.
- Los sensores: Detecta y mide una determinada magnitud del medio.
- Los sistemas integrados o embebidos cada vez más potentes: sistema de computación basado en un microprocesador o un microcontrolador diseñado para realizar una o algunas pocas funciones dedicadas
- Aprendizaje automático: Rama de la inteligencia artificial que desarrolla técnicas para que las computadoras “aprendan”.
- Avances en la ciencia de análisis de datos.

> ***Hemos de destacar este último punto, ya que no se puede hablar de soluciones IoT sin hablar del análisis de datos.*** *Para que la puesta en marcha de toda una estructura de IoT tenga sentido, se ha de contar con un sistema para gestionar y analizar los datos recopilados y así plantear mejoras. Por ello, la analítica avanzada con técnicas de Big Data, Inteligencia Artificial (IA), Machine Learning, etc. es un complemento inseparable de la tecnología IoT.*

El Internet de las Cosas (IoT) ofrece diversas ventajas, entre las que destacan la capacidad de conectarse a la red Internet, permitiendo el acceso a una amplia gama de servicios y recursos en línea (por ejemplo, el contenido al que acceden las *SmartTV*). Los dispositivos IoT utilizan diversas tecnologías inalámbricas como *WI-FI, Bluetooth, NFC, LTE, ZigBee,* etc. con la facilidad de despliegue que implica en lo relativo a infraestructuras. Además, la información se intercambia rápidamente y en tiempo real, lo que tiene aplicaciones beneficiosas, como notificaciones automáticas en casos de seguridad o emergencias.

Por otra parte, el IoT promueve la sostenibilidad al fomentar el ahorro energético al monitorear y automatizar procesos, mejorando la eficiencia y reduciendo el consumo de recursos. Aparte de todas estas ventajas, no hay que olvidar su impacto económico positivo, consecuencia directa de impulsar multitud de industrias en diferentes aspectos, e incluso oportunidades de negocio aun no descubiertas.

No obstante, el IoT presenta algunas desventajas. Para empezar, la información intercambiada a través de IoT en ocasiones carece de cifrado, aunque cabe decir que se

lleva tiempo implementando soluciones para proporcionar encriptación *end-to-end* en la mayoría de los dispositivos. Otro aspecto al que prestar atención es el relativo a la privacidad al abrir espacios privados al público y plantear problemas de seguridad, como la configuración incorrecta de sistemas de vigilancia.

También contribuye a la brecha tecnológica, ya que no todos tienen igual acceso a esta tecnología, especialmente entre diferentes regiones y áreas urbanas y rurales. Otra importante desventaja es la falta de estandarización puede dar lugar a problemas de compatibilidad entre dispositivos diseñados para funciones similares.

En resumen, el IoT ofrece ventajas significativas en términos de conectividad, eficiencia y automatización, pero presenta desafíos en seguridad, privacidad, costos iniciales y compatibilidad. Estos aspectos deben ser considerados cuidadosamente en su implementación y desarrollo continuo.[4]

2.2. Dispositivos IoT

Estos dispositivos están diseñados para objetivos muy concretos formando una red con otros dispositivos similares. De esta forma, estos elementos se suelen caracterizar porque disponen de baja capacidad de procesamiento, memoria o uso de energía, frente al resto de equipos informáticos que solemos utilizar para interactuar con la red.

Por lo tanto, IoT implica extender la conectividad a Internet más allá de los dispositivos estándar, como PC's, portátiles, *smartphones* y tabletas, a cualquier gama de dispositivos físicos y objetos cotidianos habitualmente no conectados. Integrados con la tecnología, estos dispositivos pueden comunicarse e interactuar a través de Internet, y pueden ser monitoreados y controlados de forma remota.

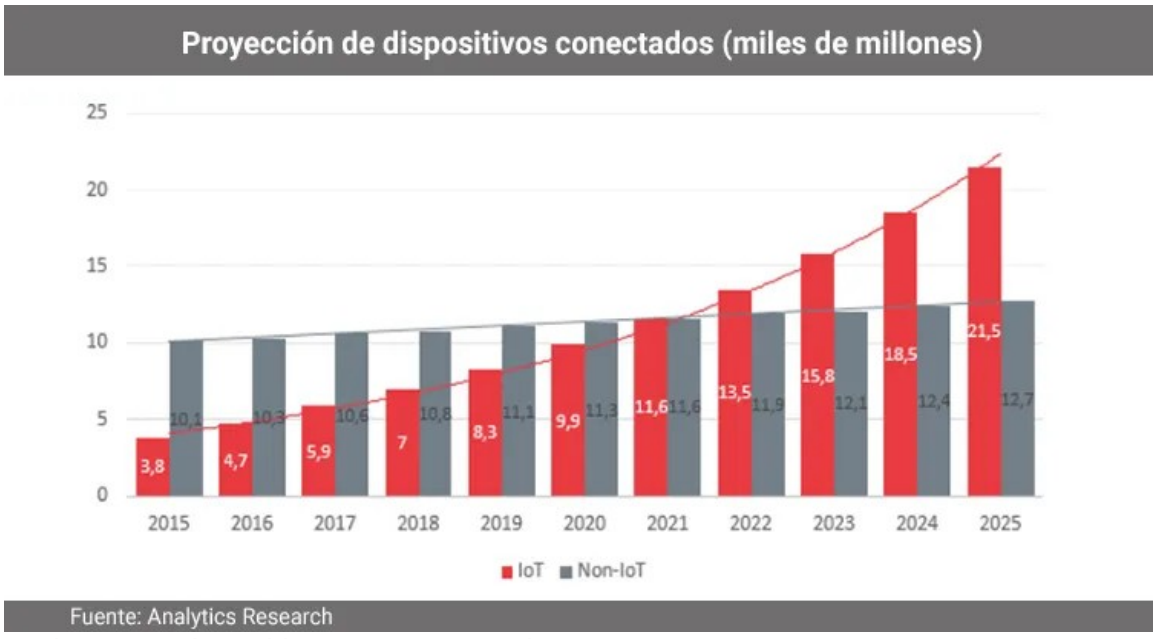


Figura 3. Dispositivos IoT conectados

2.2.1. Tipos de dispositivos IoT

Existen **diferentes tipos de dispositivos IoT** atendiendo a la complejidad de sus funcionalidades o el uso generalizado que se les da. [3]

Actualmente, si nos basamos en cómo de sofisticados pueden ser, podemos **clasificar los dispositivos IoT** en:

- **Sensores**, se encargan de la recopilación de información del ambiente, y después la envían a un gestor centralizado. Hay diferentes tipos de sensores: de proximidad, acelerómetros o giroscopios, de humedad, de temperatura, de presión, etc.
- **Sistemas embebidos o integrados**, se trata de sistemas de computación destinados a un proceso concreto. Suelen estar conectados a otros sensores y ejecutan acciones en base a determinadas condiciones. Sistemas de parada de emergencia o de alarma son ejemplos comunes en entornos industriales.
- **Wearables**, del inglés podría traducirse como “llevable” o “vestible”. Son dispositivos electrónicos que las personas llevamos en alguna parte del cuerpo regularmente y con los que interactuamos de manera constante. Hay gran variedad de dispositivos de este tipo: relojes inteligentes con GPS, monitores de frecuencia cardíaca o zapatillas con sensor de movimiento. Generalmente se gestionan mediante una App instalada en el *Smartphone* del usuario.[5]

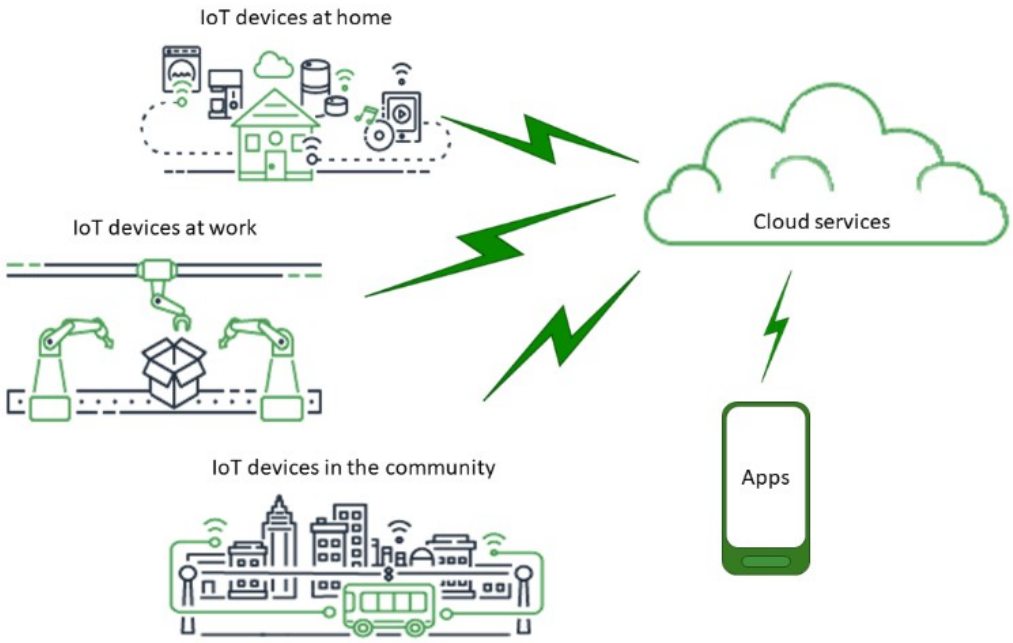


Figura 4. Tipos de dispositivos IoT

Si hablamos de los usos mayoritarios de los dispositivos IoT por sectores, entonces podemos hablar de cinco **tipos de dispositivos IoT** o aplicaciones IoT separados en:

- **IoT para consumo**: es decir, para uso cotidiano: domótica, asistentes de voz…
- **IoT comercial:** habitual en el sector salud o en la industria del transporte. Por ejemplo, un marcapasos cardíaco inteligente o un tracker de paquetería.
- **IoT militar (IoMT):** robots de vigilancia, wearables biométricos para monitorizar personas en campo, etc.
- **Industrial** IoT **(IioT):** usos típicos en el sector manufacturero y energético, por ejemplo, con sistemas digitales de control, smart agriculture y análisis de datos industriales.
- **IoT e infraestructuras:** por ejemplo, para crear Smart cities, smart buildings etc. Con todo tipo de sensores de medición y sistemas de control, incidencias, mantenimiento predictivo, etc.
- **IoT en medicina:** utilizado para una gran cantidad de funciones que te contamos en el artículo.[6]

2.2.2. Problemas de seguridad de los dispositivos IoT

Los dispositivos IoT pueden llegar a ser una presa fácil para los ciberdelincuentes que buscan este tipo de dispositivos como punto de entrada a las redes de las empresas o a otros puntos que se encuentran más protegidos.

En la mayoría de los casos, los problemas en cuanto a la seguridad de los sistemas IoT suelen ser consecuencia de la urgencia del usuario por poner en funcionamiento el dispositivo o su falta de conocimiento. Esto conduce a la utilización de configuraciones predeterminadas de fábrica, contraseñas que no se modifican, ejecución de servicios innecesarios y la exposición de información en la red Internet.

Adicionalmente, es importante considerar la carencia de actualizaciones de software en estos dispositivos. Resulta común que los usuarios no presten la debida atención a la actualización de sus *smartphones*, ordenadores personales y otros dispositivos para instalar las últimas versiones de sistemas operativos que contienen parches de seguridad diseñados para remediar o mitigar vulnerabilidades detectadas. Esta falta de atención es aún más notoria en el caso de los dispositivos IoT, con la particularidad de que los fabricantes tienden a no publicar actualizaciones o parches para los sistemas operativos de estos dispositivos, ya que se caracterizan por ser sistemas de bajo costo y con un ciclo de vida limitado. En consecuencia, suele recaer en el usuario la responsabilidad de buscar y aplicar dichas actualizaciones por su cuenta. Obviamente, esta responsabilidad resulta excesiva para usuarios sin conocimientos en sistemas informáticos y/o seguridad en redes. [6]

Es más, para ponernos en perspectiva de hasta que punto están expuestos los dispositivos, existen buscadores cuyo objetivo es indexar dispositivos y servicios accesibles desde Internet, como es el caso de **Shodan, Censys** u otros. Con estos

buscadores podemos encontrar todo tipo de dispositivos IoT, desde *webcams*, televisores inteligentes y dispositivos del hogar hasta semáforos, turbinas eólicas, y cualquier otro tipo de infraestructura que use la red para enviar los datos. No es un buscador de servidores vulnerables, sino que muestra todos los dispositivos, y puede que algunos sean vulnerables.

Los ataques sobre los dispositivos IoT buscan principalmente estos objetivos:

- Infectarlos mediante *malware* para formar parte de una red zombi (***botnet***) que los utilicen para realizar ciberataques (por ejemplo, de denegación de servicio distribuida o *DdoS*). *[Este apartado lo desarrollaremos más adelante]*.
- Utilizarlos como puente o punto de entrada para realizar movimientos laterales y atacar otros equipos de la misma red, robar información, comprometer servidores...
- Modificar su configuración para inhabilitarlos o cambiar sus condiciones de utilización.[7]

Vectores de ataque

Los vectores de ataque son los métodos que utilizan los ciberdelincuentes para hacerse con su objetivo, como por ejemplo, obtener información o tomar el control del dispositivo. [6]

Los principales vectores de ataque que afectan a los dispositivos IoT son:

- Errores en la implementación de los dispositivos IoT dentro de la red. El principal error que se comete es no segmentar adecuadamente la red (mediante *firewalls* u otras herramientas), evitando que ante un acceso no autorizado al dispositivo IoT, se tenga además acceso a otros recursos de la red.
- Si un atacante consigue acceso a la red local o LAN, donde se encuentra el dispositivo IoT, o al receptor de la información, podría modificar la información que se intercambian mediante ataques de *Man-in-the-Middle (*hombre en el medio).
- Acceso no autorizado a la plataforma de administración. Los dispositivos IoT suelen contar con una interfaz web o aplicación móvil para su administración. Si estas cuentan con vulnerabilidades o configuraciones de seguridad deficientes, se podría producir un acceso no autorizado y los ciberdelincuentes podrían controlar el dispositivo remotamente.
- El acceso físico a los dispositivos IoT por parte de los ciberdelincuentes puede ser el origen de un incidente de seguridad. En algunos casos, estos dispositivos se encuentran expuestos a posibles accesos no autorizados a la información que gestionan, robos o vandalismo.
- Como en todo sistema informático, los usuarios son un factor muy importante en cuanto a la seguridad. Los atacantes pueden llevar a cabo ataques de ingeniería

social, con los que provocan que los usuarios realicen acciones potencialmente maliciosas, como facilitar las credenciales de acceso al dispositivo o instalar actualizaciones fraudulentas que contienen algún tipo de *malware*.

> *"No obstante, las amenazas a los dispositivos IoT no se reducen a las derivadas de su conectividad a Internet. Muchos de estos aparatos cuentan también con capacidades de conexión inalámbrica como wifi, Bluetooth o Zigbee, lo que puede suponer otro vector de ataque para ciberdelincuentes si se encuentran dentro de su rango de acción." Fuente: Seguridad en la instalación y uso de dispositivos IoT de INCIBE*

Botnets

Es necesario dedicar un apartado a este concepto ya que los dispositivos IoT, por su amplia adopción y su conexión a internet, son comúnmente utilizados por ciberdelincuentes para formar *botnets*.

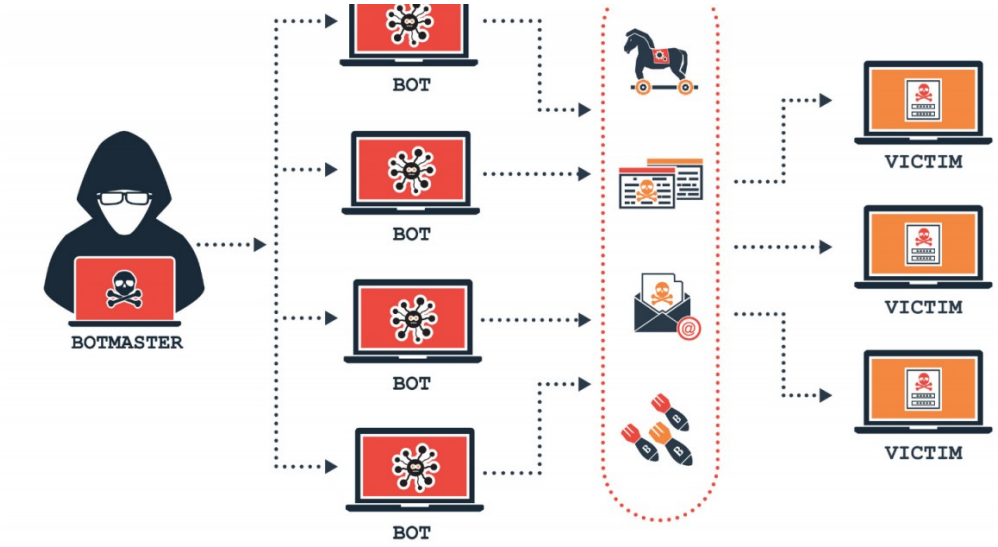


Figura 5. Bootnet

Una *botnet* es una red de dispositivos infectados, como computadoras, servidores, dispositivos IoT o incluso dispositivos móviles, que son controlados por un atacante de manera remota y coordinada. Estos dispositivos han sido infectados por *malware*, permitiendo al atacante controlarlos sin el conocimiento del propietario legítimo. Al realizar operaciones de manera concertada y masiva, pueden amplificar su impacto notablemente ya que puede tratarse de cientos o incluso miles de dispositivos involucrados en los ataques.[8]

*Las computadoras secuestradas en una botnet convencional son conocidas como zombies o bots. David Knight, de Proofpoint, acuñó la palabra **thingbot** para referirse a dispositivos que no sean computadoras que han sido infectados por una botnet. Fuente: "The Internet of Things – Thingbot" en YouTube.*

Las *botnets* se usan para distintos tipos de ataques. Algunos ataques se lanzan para agregar más dispositivos a la red zombi, pero hay otros con objetivos más concretos:

- **Leer y escribir datos del sistema:** se solicita a los dispositivos el envío de archivos a un servidor y así poder revisarlos en busca de datos delicados. Dichos archivos de sistema podrían contener credenciales para el acceso a una infraestructura, por ejemplo.
- **Monitorizar la actividad del usuario:** las *botnet* suelen incluir otros tipos de malware como, por ejemplo, *keyloggers* que registran y graban las pulsaciones de teclas del teclado del usuario y le envían la información robada a un servidor controlado por el atacante.
- **Escanear la red local en busca de vulnerabilidades adicionales:** los dispositivos infectados escanean los recursos de red locales una vez que se instalan en un dispositivo específico en busca de vulnerabilidades.
- **Lanzar un ataque distribuido de denegación de servicio o DDoS:** los DDoS son un tipo de ataque muy común con *botnets*. El atacante puede necesitar varios miles de equipos para lanzar un DDoS que puede provocar degradación del servicio muy grave de forma que se sobrecarga una red o un servidor, haciendo inaccesible el acceso a sus usuarios. Los ataques DDoS suelen dirigirse a organizaciones o administraciones con el fin de dañar su reputación. Uno de los más grandes fue ejecutado por la *botnet* **Mirai**, compuesta principalmente por dispositivos IoT.
- **Enviar spam por correo electrónico:** con acceso a cuentas de correo electrónico en dispositivos locales, el atacante enviar correos electrónicos a destinatarios específicos para expandir malware o realizar phishing con distintos objetivos.

El *malware* en los dispositivos infectados permanecen en modo hibernación hasta que reciben instrucciones del *hacker* que actúa como "*bootmaster*", a través de un servidor central llamado "C&C" (Comando y Control). El *malware* se comunica con el C&C mediante protocolos comunes que, generalmente, no son bloqueados por los cortafuegos, como el protocolo HTTP.

*Los ciberdelincuentes incluso se lucran ofreciendo **DDoS-as-a-service**, permitiendo a otros usuarios suscribirse a planes en el servidor C&C para enviar comandos a los dispositivos infectados.*

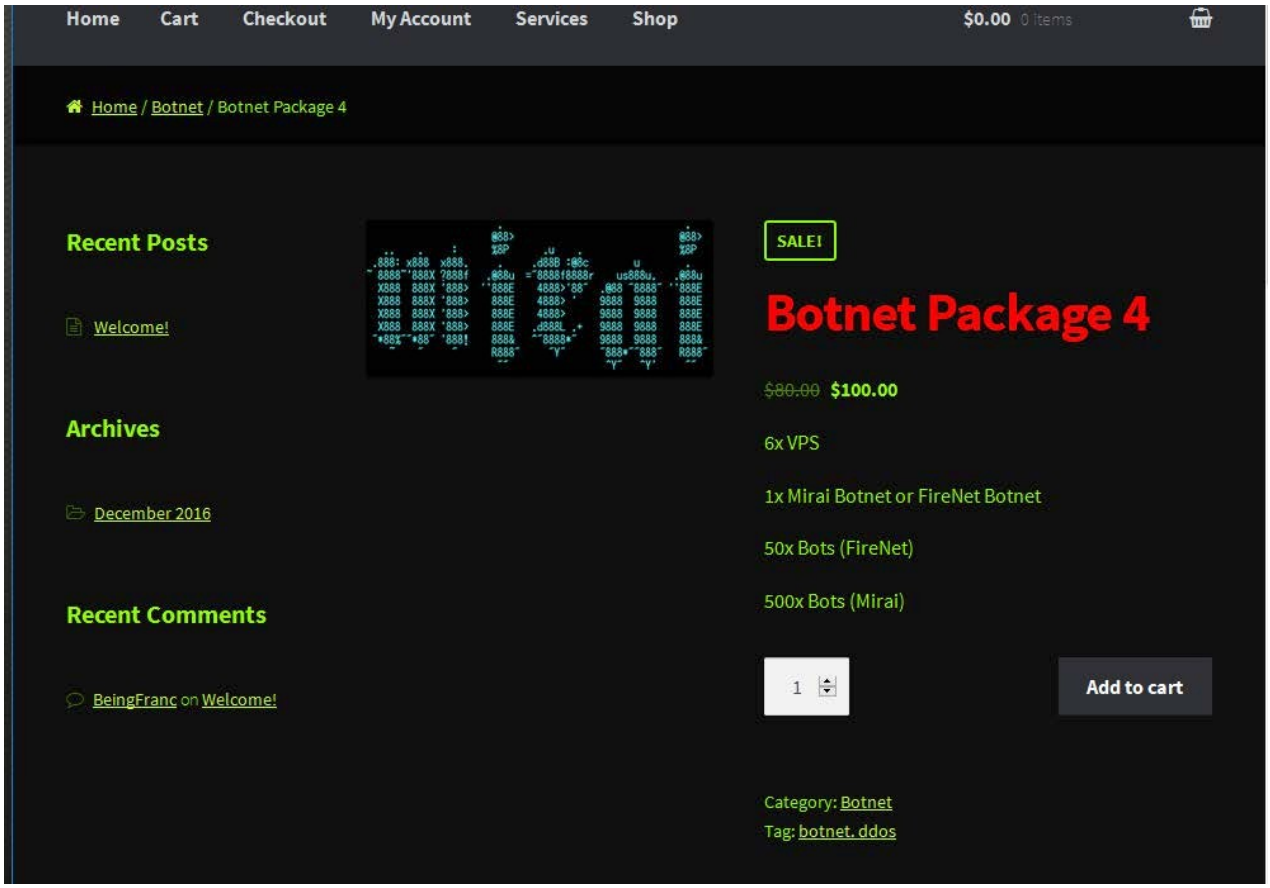


Figura 6. Servicio DDoS-as-a-Service Mirai

Para evitar la desactivación de la *botnet*, los creadores del *malware* incluyen múltiples opciones de C&C como el modelo peer-to-peer (P2P), donde los dispositivos infectados actúan como un sistema distribuido C&C, siendo más difíciles de desactivar y preferidos a los modelos centralizados por su mayor resistencia.

Cuando se reciben comandos, los dispositivos infectados ejecutan ataques o realizan acciones según las instrucciones del controlador. Esto puede generar ralentizaciones en la red y degradaciones en el rendimiento de los dispositivos infectados durante el ataque pero, una vez completado, el rendimiento vuelve a la normalidad y el *malware* vuelve a su estado latente. La clave está en pasar desapercibido.[9]

A continuación, comentamos brevemente las *botnets* más conocidas

- **Mirai**, se trata de un malware de la familia de las botnets orientado a IoT, cuyo objetivo principal es infectar routers o cámaras IP. Se aprovecha de la falta de seguridad en configuraciones por defecto para acceder a los sistemas y así infectarlos. Esta botnet fue descubierta en 2016, ayudada por el crecimiento y la popularidad de las cámaras IP han permitido una rápida dispersión, además, de una simplificación de ataques DDoS. Mirai ha sido utilizada en los ataques más

conocidos del tipo DDoS, como por ejemplo el que recibió el proveedor de DNS Dyn en 2016. [10]

- **Hajime**, detectado por primera vez en 2016, este malware está construyendo una gran red botnet peer to peer, un grupo descentralizado de máquinas comprometidas que, de forma discreta, lanza ataques de *spam* o de DDoS. Principalmente realiza ataques directos contra contraseñas de los dispositivos, para infectarlos, dando a continuación un numero de pasos para ocultarse de la víctima. Este *malware* se dirige en concreto contra grabadores digitales de video, cámaras web y routers. [11]
- **Reaper**, detectada en 2017, está formada por dispositivos IoT multimedia que poseen algún tipo de vulnerabilidad. Sobre todo, decodificadores y reproductores de contenidos con sistemas operativos basados en Linux, aunque también se pueden encontrar cámaras IP y routers. Este *malware* se basa en la operación del rastreo Telnet en busca de servicios no protegidos de forma correcta y posee un listado con algunas de las vulnerabilidades más comunes que podrían no haberse resuelto. [12]

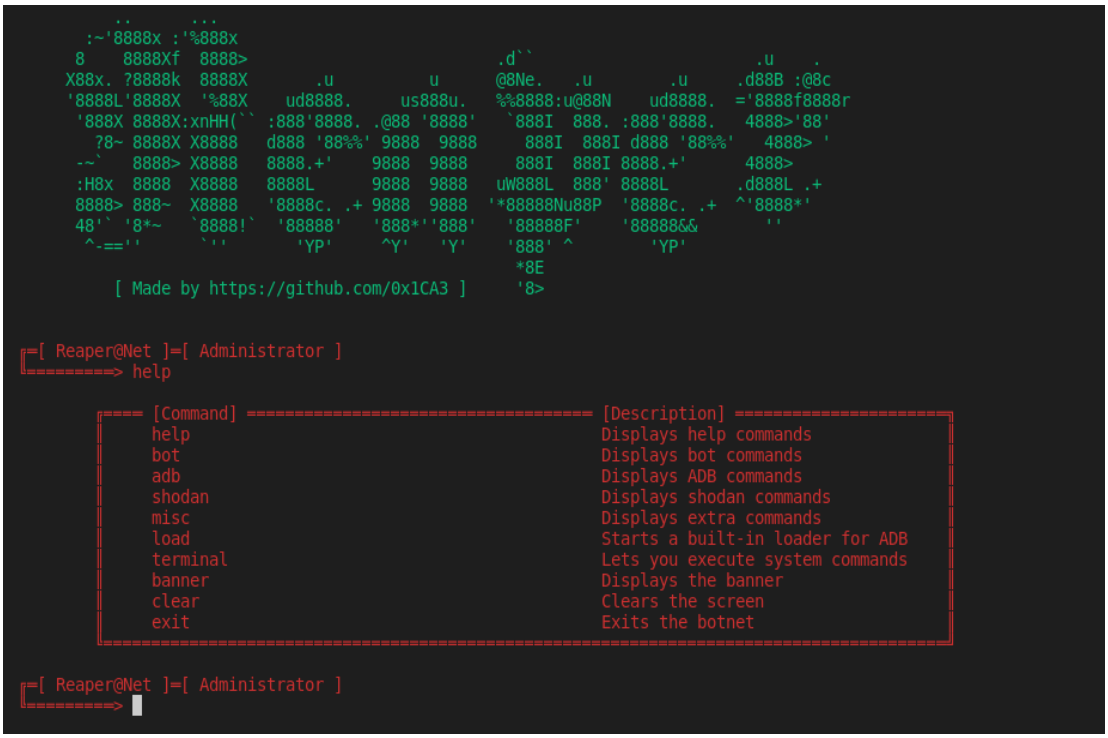


Figura 7. Interfaz de administrador Reaper

- **Rustock**, se propago como un troyano, infectando documentos descargados desde internet o como adjunto de *emails*. Las máquinas infectadas enviaban más de 20000 mensajes basuras por hora. Fue descubierta en el año 2006 y desarticulada por la Ley de EEUU en cooperación con Microsoft, FireEye y la universidad de Washington.

- **Storm**, realizaba múltiples ataques como incluir accesos secretos, retransmisiones SMTP, recolección de direcciones *email*, *spam* y DdoS. Fue la botnet más grande y con mayor propagación conocida hasta la fecha, destaca por su capacidad de realizar ataques DDoS. La ingeniería social y el *spam* ayudaron a su propagación, pero sus atacantes también lo lanzaron a través de las descargas en sitios web populares que se vieron comprometidos, haciendo de las descargas el principal factor de infección.
- **Cutwail**, se caracteriza por enviar *spam* y realizar ataques DDoS. En su apogeo fue la responsable de casi la mitad de todo el *spam* en internet, enviando 74 billones de mensajes de spam por día. Hoy en día sigue activa y disponible para alquilar.
- **Mariposa**, se caracteriza por realizar ciberestafas y ataques DdoS. Desarticulada gracias a los esfuerzos de las fuerzas de seguridad españolas, Defense Intelligence, Georgia Tech y Panda Security. Se trata de un *keylogger*, es decir, software que monitoriza y graba en un registro la actividad del teclado del usuario para capturar credenciales en sitios bancarios, credenciales con las que realizar envió masivos de spam y tomar el control del equipo para su utilización en ataques DDoS. La red Mariposa está disponible para alquilar. [13]

2.3. Honeypot

Desde los primeros incidentes de seguridad, la tarea principal de los sistemas de seguridad informáticos ha sido buscar e implementar defensas contra los ataques conocidos. La mera defensa limita la comprensión de los incidentes, con escaso conocimiento sobre las herramientas de ataque y su impacto en la seguridad de los sistemas. Para ampliar el conocimiento sobre los instrumentos y métodos empleados se desarrolló el *Honeypot*.

Un *Honeypot* busca recopilar datos sobre la actividad de los intrusos, permitiendo la detección temprana de vulnerabilidades y la comprensión de riesgos en los sistemas. Este enfoque busca aprender de amenazas y del comportamiento de los atacantes para implementar una seguridad proactiva que no solo defienda contra amenazas, sino también las identifique antes de que ocurran.

Figura 8. Honeypot

> *Definiremos Honeypot (tarro de miel textualmente) como "un recurso de red destinado a ser atacado o comprometido. De esta forma, un Honeypot será examinado, atacado y probablemente comprometido por cualquier atacante. Los Honeypot no tienen en ningún caso la finalidad de resolver o arreglar fallos de seguridad en nuestra red. Son los encargados de proporcionarnos información valiosa sobre los atacantes en potencia a nuestra red antes de que comprometan sistemas reales." Fuente: Gabriel Verdejo Alvarez – "CAPÍTULO 4: SEGURIDAD EN REDES IP: Honeypots y Honeynets" Departamento de Ciencias de la Computación (UPC)*

Generalmente un *honeypot* puede ser una computadora o un sitio de red (real o virtual) que parecen formar parte de una red pero que en realidad están aislados, protegidos y monitorizados, y que aparentan contener información o recursos potencialmente valiosos.

Gracias a esta herramienta, se pueden descubrir nuevas formas de ataque desconocidas hasta el momento. Además, se pueden descubrir vulnerabilidades en nuestra red o de los sistemas informáticos, lo que ayudará en el diseño e implantación de soluciones y estrategias de protección más efectivas y eficientes.

Este concepto se puede ampliar de tal forma que se pueden tener varios Honeypots implementados formando una red (*Honeynet*) existiendo comunicación entre ellos y proporcionando más credibilidad en su función de señuelo. [14]

Objetivos

Los Honeypots pueden estar diseñados y programados para diferentes objetivos, como puede ser:

- Alertar, un Honeypot puede estar diseñado y programado para detectar un ataque.
- Obtener información, un Honeypot puede estar diseñado y programado con el objetivo de obtener información sobre el ataque que está detectando.
- Ralentizar, un Honeypot puede estar diseñado y programado con el objetivo de ralentizar el ataque que está detectando.
- Combinación, un Honeypot puede estar diseñado y programado con el objetivo de alertar, obtener información y ralentizar el ataque.

Dependiendo de los objetivos a alcanzar se implementará un tipo u otro, aunque hay que tener en cuenta que, a mayor capacidad o complejidad, serán necesarios mayores requisitos para su correcto funcionamiento.[15]

Clasificación

Existen multitud de clasificaciones atendiendo a distintos aspectos y características.

Una de ellas, se basa en el medio de implementación del sistema. Hay *honeypots* físicos y virtuales:

- **Físico**: se trata de una máquina real con su propia dirección IP, esta máquina simula comportamientos modelados por el sistema. No es el más utilizado debido sobretodo al mayor coste en *hardware* al requerir de equipos dedicados, su mantenimiento y configuración de algunos dispositivos de HW especializados.

- **Virtual**: es la modalidad más frecuente ya que es sencillo, se implementa con facilidad en diferentes plataformas, y es más económico ya que se aprovecha la capacidad del *hardware* existente.

También, se pueden distinguir los *honeypot* atendiendo a su finalidad:

- **De producción:** Ayudan a minimizar los riesgos de una organización, los mismos que añaden nuevas características a las medidas de seguridad. Así mismo se encargan de detectar ataques y disuadir a los atacantes. Utilizados generalmente por grandes empresas y organizaciones.
- **De investigación:** Están diseñados para obtener información de los atacantes, no añaden ninguna característica de seguridad a una organización, solamente son utilizados en la investigación de nuevos mecanismos de intrusión en los sistemas. Son utilizados mayormente por organizadores sin ánimos de lucro e instituciones educativas con fines educativos y de investigación.

Además, según los criterios de diseño, los *honeypots* se pueden clasificar en:

- **Honeypots puros:** sistemas de producción completos. Las actividades del atacante se controlan mediante el uso de un *bug tap* (o capturador de paquetes) en el enlace del *honeypot* a la red. No es necesario instalar ningún otro *software*. Aunque un *honeypot* puro es útil, el sigilo de los mecanismos de defensa puede garantizarse mediante un mecanismo más controlado.
- **Honeypots de alta interacción:** estos *honeypots* normalmente son equipos con sistemas reales que trabajan en una red real, como puede ser cualquier servidor físico (por este motivo deben estar cuidadosamente aislados). Además, se defienden de los atacantes para darle un mayor realismo.
  En general, los *honeypots* de alta interacción brindan más seguridad al ser difíciles de detectar, pero su mantenimiento es costoso.
- **Honeypots de baja interacción**: tienen una interacción casi nula, y su funcionalidad se limita a imitar a aplicaciones u otros sistema o equipos de la red. Este tipo de *honeypot* no realiza ningún tipo de interacción con el atacante, actúa de forma completamente pasiva y no «se defiende» de los atacantes, simplemente registra todo lo ocurrido.

Dado que consumen relativamente pocos recursos, varias máquinas virtuales se pueden alojar fácilmente en un sistema físico.

Se podría decir que una diferencia fundamental entre los Honeypot de baja interacción y los Honeypot de alta interacción es que los de baja interacción están diseñados para identificar y analizar los ataques que recibe de forma automática. Los Honeypot de alta interacción están para recibir ataques que le llegan de forma manual. [16]

Ejemplos

En la actualidad, los *honeypots* son utilidades muy potentes, que permiten simular el comportamiento real de un sistema, haciendo creer a los intrusos que han entrado en un sistema real y que es fácil hacerse con el control.

Hay multitud de servicios a emular como por ejemplo SSH, FTP, HTTP, RDP, MySQL, SMTP, VNC, entre otros.

En el caso del SSH, se trata de uno de los que más incidencias registran de forma diaria. A estos llegan gran cantidad de ataques, *malwares* o direcciones IP. Por lo general, la mayor parte de estos no es detectado por los antivirus, y la mayoría de las IP analizadas, no están reflejadas en listas de reputación.

Entre los diferentes tipos dependiendo de su funcionalidad, podemos encontrar:

- Honeypots SSH
- Honeypots HTTP
- Honeypots de WordPress
- Honeypots de bases de datos (BBDD)
- Honeypots de correo electrónico
- Honeypots de IOT (Internet of Things)

Cabe destacar que existe una solución magnífica llamada T-POT que nos permite tener varios *Honeypots* (incluidos los diseñados para simular *ICS*) dentro de una sola máquina virtual (o física, según sea necesario) y también nos permite visualizar la información de forma espectacular utilizando ELK.

La gran ventaja de esta distribución *T-POT* es que los integra todo es una misma instalación (en el mismo servidor) y todos virtualizados con *Docker*. Esto permite tener en ejecución varios demonios actuando sobre la misma tarjeta de red sin problemas. Además, al tener cada *Honeypot* su entorno *dockerizado*, es muy sencillo su mantenimiento (actualizaciones, por ejemplo), gestión y personalización. Es ampliable, de forma que se pueden añadir diferentes recursos que la comunidad va creando: [https://github.com/paralax/awesome-honeypots].

Ubicación de los Honeypots

La ubicación de los Honeypots es muy importante ya que de esta manera se puede maximizar la seguridad, es decir, debido a su carácter pasivo una ubicación de difícil acceso eliminará gran parte de su atractivo para posibles atacantes.

Por otro lado, si su ubicación es demasiado obvia cualquier potencial atacante la descubrirá y evitará todo contacto con ella.

Existen tres ubicaciones para ubicar los *honeypots* dependiendo de sus necesidades:

- **Antes del *firewall:*** Esta localización permite evitar el aumento de los riesgos inherentes a las instalaciones de los *honeypots*. Como se encuentra fuera de la zona protegida por el *firewall*, puede ser atacado sin ningún tipo de peligro o problema para el resto de la red.

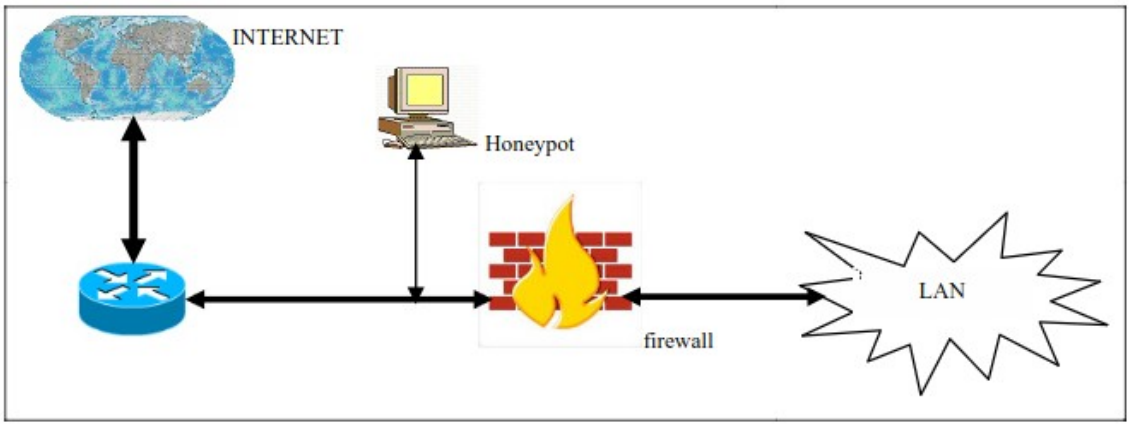


Figura 9. HP antes del firewall

En la figura se aprecia que el *honeypot* se encuentra desprotegido del *firewall* por lo que es propenso a ser atacado con más facilidad.

Con esta configuración evitaremos las alarmas de otros sistemas de seguridad de nuestra red (IDS) al recibir ataques en el *honeypot*. Sin embargo, existe la posibilidad de generar mucho tráfico debido precisamente a la exposición del *honeypot*.

Cualquier atacante externo será lo primero que encuentra y esto generará un gran consumo de ancho de banda y espacio en los ficheros de log. Por otro lado, esta ubicación nos evita la detección de atacantes internos.

- **Detrás del firewall:** En esta posición el *honeypot* queda afectado por las reglas de filtrado del *firewall*. Por un lado, se tiene que modificar las reglas para permitir algún tipo de acceso al *Honeypot* por posibles atacantes externos. Por otro lado, al introducir un elemento potencialmente peligroso dentro de la red, puede permitir a un atacante que gane acceso al *Honeypot* tener acceso a todos los sistemas que están dentro de la red.

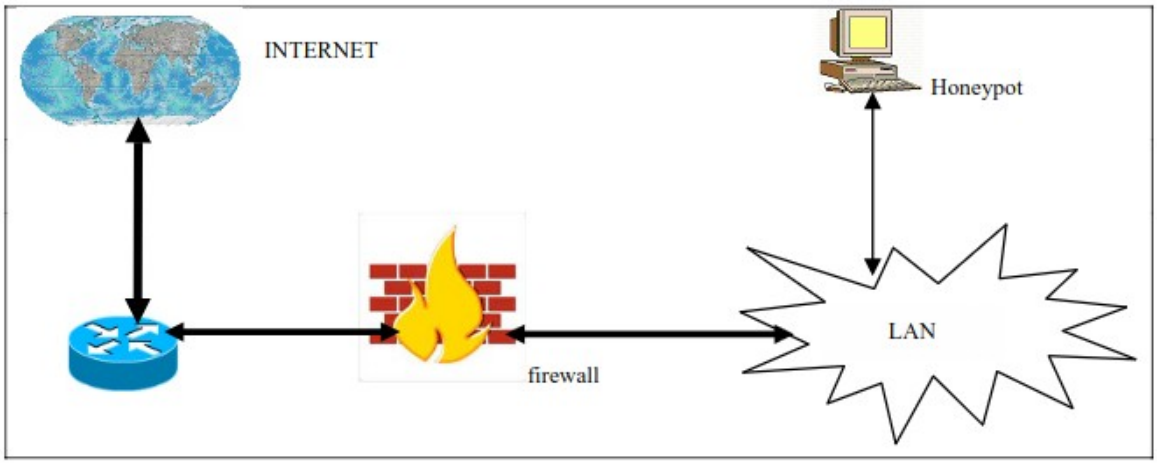


Figura 10. HP detrás del firewall

- Esta ubicación permite la detección de ataques internos, así como Firewall mal configurados o maquinas infectadas por gusanos y virus.

  Estas configuración se suele realizar en casos como la detección de atacantes internos o la imposibilidad de utilizar una dirección IP externa para el *Honeypot.*

- **En la zona desmilitarizada o DMZ:** la ubicación en la DMZ permite por un lado juntar en el mismo segmento a servidores de producción con el *Honeypot*. Por otro lado, permite controlar el peligro que añade su uso, puesto que se tiene un Firewall que lo aísla del resto de la red. Esta ubicación permite tener la posibilidad de detectar ataques externos e internos con una simple re-configuración del Firewall, ya que se encuentra en la zona de acceso público. [17]

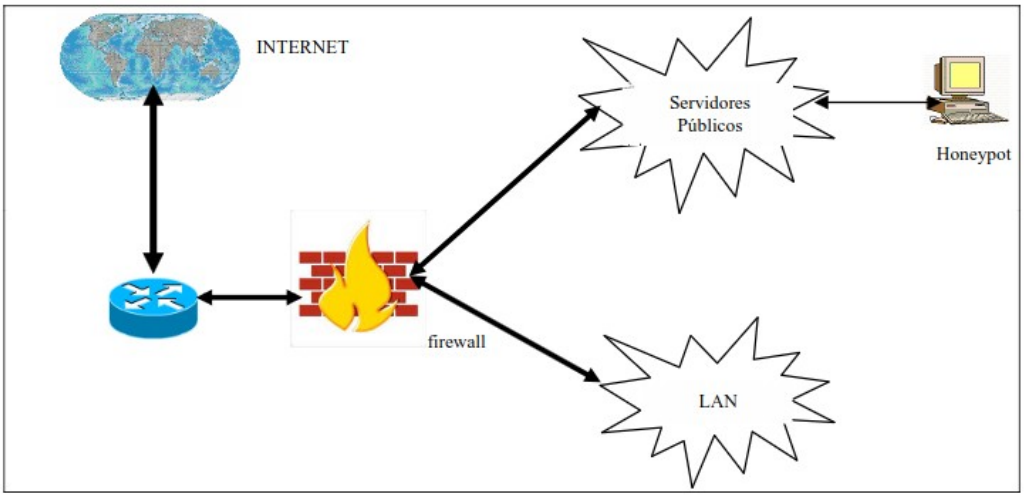


Figura 11. En la DMZ

Esta arquitectura nos permite tener la posibilidad de detectar ataques externos e internos con una simple re-configuración de nuestro sistema de *firewall* puesto que se encuentra en la zona de acceso público.

Además, eliminamos las alarmas de nuestros sistemas internos de seguridad y el peligro que supone para nuestra red al no estar en contacto directo con esta.

La detección de atacantes internos se va algo debilitada, puesto que al no compartir el mismo segmento de red que nuestra LAN un atacante local no accederá al *Honeypot*. Sin embargo, desde la red local si que es posible acceder al *Honeypot*, por lo que un atacante interno que intente atacar nuestros servidores públicos u otros sistemas externos (un gusano por ejemplo) muy probablemente acabe siendo detectado.

Ventajas y desventajas

Por un lado, la utilización de Honeypots presenta las siguientes ventajas:

- Los Honeypots son sistemas a los que ningún usuario puede acceder, permitiendo de esta forma, revelar cualquier tipo de acceso del atacante o una configuración errónea del sistema, sin llegar a tener prácticamente falsos positivos.
- Se necesitan recursos mínimos, puesto que, a diferencia de otros sistemas de seguridad, sus requisitos son mínimos, prácticamente no consumen ancho de banda ni memoria. No se necesitan arquitecturas complejas, ni un gran número de sistemas centralizados, cualquier ordenador conectado a la red puede realizar el trabajo de un Honeypot.
- Es un tipo de sistema que sirve para atacantes internos como externos. Su objetivo es el de pasar de manera desapercibido en una red con más equipos.
- Generan un pequeño volumen de datos cuando están desplegados en la ubicación correcta y al contrario que los demás sistemas de seguridad, como Firewall o el IDS, que generan un gran volumen de datos conteniendo incluso información que no es necesaria, mientras que los Honeypots generan información con muy pocos datos, pero de muy alto valor.

Por otro lado, se pueden enumerar algunas desventajas:

- Son elementos totalmente pasivos, es decir, si no reciben ningún ataque no sirven para nada, por ello siempre tienen que estar ubicados correctamente.
- Son fuentes potenciales de riesgo para una red, debido a la atracción que ejercen sobre los ciberdelincuentes. De tal manera que, si no se mide su alcance y no se convierte en un entorno controlado, puede ser utilizado como fuente para ataques a otras redes o incluso la propia red.
- Tienen una visión limitada, puesto que solo pueden rastrear y capturar determinado tipo de actividad. En este sentido, cabe tener en cuenta que los Honeypots no

sustituyen a ningún elemento de seguridad (Firewall, IDS), si no que trabajan paralelamente con estos, permitiendo mejorar el perímetro de seguridad de una red. [14][18]

Capítulo 3. Herramientas

En el siguiente capítulo, se presentan las herramientas que se han usado para el desarrollo del proyecto y algunas que se han estudiado a fin de utilizarlas. Como ya se explicó anteriormente, la solución propuesta pasa por crear una máquina virtual en el equipo en el que trabajamos habitualmente, para así, contar con un sistema "independiente" donde vamos a implentar la herramienta HoneyPot. Hecho esto, se recurrirá a otras aplicaciones adicionales con el organizar y visonar los datos recogidos y analizar archivos.

A continuación, desarrollamos los diferentes componentes software del proyecto:

3.1. Honeypots

3.1.1. Kippo

Es uno de los *Honeypots* de baja interacción más utilizados, debido a su alto nivel de análisis y detección. Se trata de un Honeypot que emula un servicio SSH de interacción media diseñado para registrar ataques de fuerza bruta y, lo que es más importante, registrar toda la interacción shell realizada por el atacante.

Hay que tener en cuenta que la gran mayoría de ataques producidos en una red son realizados una vez el atacante ha tomado el control, por lo que es interesante implantar un Honeypot que detecte este tipo de ataques, teniendo la posibilidad de mitigar el ataque desde la raíz. [19]

3.1.2. IoTPot

Se trata de un *Honeypot* que atrae y analiza ataques basados en Telnet contra dispositivos IoT, emulando arquitecturas de CPU como MIPS, ARM, PPC, ofreciendo diferentes tácticas de login. Consiste en dos partes, una interfaz que responde con poca interacción y un backend que es un entorno virtual de alta interacción llamado IoTBOX que admite entre otras las arquitecturas mencionadas. [20]

3.1.3. Telnet IoT Honeypot

Este Honeypot simula un dispositivo IoT que expone el puerto Telnet (TCP 23) para recibir ataques. De esta forma, recoge malware y lo analiza, mostrándolo todo en un frontend.

Esta aplicación se trata de una arquitectura cliente/servidor, con un cliente fron-end que acepta las conexiones telnet y un servidor que recibe información sobre las conexiones, para posteriormente analizarlas. El servidor backend ofrece una interfaz HTTP que sirve para acceder al frontend, así como para enviar nueva información de conexión al backend por parte de los clientes. [21]

3.1.4. Cowrie

Este es el *Honeypot* elegido para nuestro proyecto.

Esta herramienta Cowrie tiene características muy interesantes, como por ejemplo simular un sistema de archivos completo con la posibilidad de crear y borrar archivos, de esta manera, un posible atacante podrá creerse que está en el sistema operativo real, cuando en realidad está dentro del Honeypot.

Otra característica interesante es que podremos añadir ficheros “falsos” (*honeytoken*) que proporcionan más interacción con el atacante y, por tanto, más realismo a la trampa.

Todos los logs son almacenados en un formato UML compatible, de esta manera, podremos estudiar detalladamente todos los pasos que ha realizado un posible atacante. Cowrie también es capaz de guardar archivos descargados de Internet a través de wget o cURL, o también de archivos subidos a través del protocolo SFTP o SCP para posteriormente estudiar a fondo qué son esos archivos que el ciberdelincuente ha intentado colarnos en el sistema.

> *Cowrie está basado en el honeypot Kippo, pero tiene características adicionales que lo hacen mucho más interesante. Por ejemplo, soporta comandos ejecutados a través de SSH (SSH exec), de esta manera, el atacante podrá enviar comandos.*

Por otra parte, este Honeypot cuenta con la capacidad de registrar en un log todos los intentos de conexiones . También es capaz de hacer un reenvío de las conexiones SMTP a un honeypot SMTP diseñado para tal fin. El único requisito para hacer funcionar esta herramienta Cowrie Honeypot es tener instalado Python y también necesitamos Python-virtualenv. [22]

3.1.5. T-POT

Se trata de una plataforma de Honeypots que tiene como base una distribución Linux Ubuntu Server. Esta plataforma incluye una gran variedad de Honeypots ya preparados, configurados y listos para entrar en funcionamiento.

Figura 12. T-POT

Es una herramienta bastante potente y completa, y por ello, en un primer planteamiento del proyecto, se estudió utilizarlo pero sus requerimientos resultan excesivos para los medios hardware con los que contamos. Su rendimiento no fue bueno en algunas pruebas iniciales y comprendimos que realmente no era necesario tal despliegue teniendo en cuenta los objetivos reales del proyecto.

```
### Removing NGINX default website.
### Waiting a few seconds to avoid interference with service messages.
### Please choose your install type and notice HW recommendation.

    [T] - T-Pot Standard Installation
          - Cowrie, Dionaea, Elasticpot, Glastopf, Honeytrap, Suricata & ELK
          - 4 GB RAM (6-8 GB recommended)
          - 64GB disk (128 GB SSD recommended)

    [H] - Honeypots Only Installation
          - Cowrie, Dionaea, ElasticPot, Glastopf & Honeytrap
          - 3 GB RAM (4-6 GB recommended)
          - 64 GB disk (64 GB SSD recommended)

    [I] - Industrial
          - ConPot, eMobility, ELK & Suricata
          - 4 GB RAM (8 GB recommended)
          - 64 GB disk (128 GB SSD recommended)

    [E] - Everything
          - All of the above
          - 8 GB RAM
          - 128 GB disk or larger (128 GB SSD or larger recommended)

Install Type:          Starting Cleanup of Temporary Directories...
[  OK  ] Started Cleanup of Temporary Directories.
```

Figura 13. Modos de instalación T-POT

Algunos de los Honeypots y herramientas que contiene son las siguientes:

- **Conpot**, es un Honeypot para ICS, el cual permite simular un entorno industrial completo, capaz de hacer ver al atacante de que esta accediendo a un entorno industrial.
- **Cowrie**, se trata de un Honeypot que simula un servidor SHH y Telnet diseñado para monitorizar los ataques de acceso, así como la interacción con la Shell.
- **Dionaea**, otro honeypot de caracter general diseñado para simular vulnerabilidades de red y servicios como SMB, http, FTP, MSSQL e incluso VoIP.
- **Elasticpot**, es un Honeypot basado en una versión simplificada de ElascticSearch.
- **Emobility**, Honeypot de infraestructuras ICS que simula un centro de carga eléctrica de vehículos, simulando incluso que los vehículos están en proceso de carga.
- **Glastopf**, es un Honeypot orientado a aplicaciones web, como, por ejemplo: web mail, wikis... cualquier aplicación en la que el cliente se ejecute desde el navegador web.
- **Honeytrap**, este Honeypot se centra especialmente en observar ataques contra servicios TCP y UDP.
- **Suricata**, es un monitor de seguridad de red para detectar intrusiones en tiempo real inspeccionando el tráfico de red.
- **ELK**, son tres herramientas en una, la primera Elasticsearch (servidor de búsquedas). La segunda, Logstash (administración de logs). La tercera, Kibana (visualización y gestión de los datos almacenados). Estas herramientas vamos a utilizarlas para nuestro proyecto y nos ayudarán a interpretar los datos obtenidos fácilmente.

La gran ventaja de esta distribución es que integra todo en una misma instalación y todos los servicios están virtualizados con Docker. Esto permite tener en ejecución varios demonios actuando sobre la misma tarjeta de red sin problemas. Además, al tener cada Honeypot su entorno dockerizado, su mantenimiento, gestión y personalización es muy sencilla.[23][24]

3.2. Ubuntu

Figura 14. Ubuntu

Ubuntu es un popular sistema operativo gratuito y de código abierto basado en Linux utilizable tanto en un ordenador como en un servidor privado virtual. [25]

Las principales características de Ubuntu le han hecho ser una distribución Linux muy popular:

- Interfaz intuitiva y fácil de usar.
- Seguridad sólida, actualizaciones regulares y soporte a largo plazo.
- Gran comunidad de soporte y recursos en línea.
- Amplia compatibilidad con hardware y software.
- Varias ediciones para diferentes necesidades (escritorio, servidor, etc.).
- Rendimiento ligero.
- Gratuidad

En nuestro caso hemos utilizado la versión 22.04 de esta distribución.

3.3. Pila ELK o Elastic Stack

Figura 15. ELK

La pila **ELK** es un acrónimo que se usa para describir una pila que se compone de tres proyectos de código abierto: Elasticsearch, Logstash y Kibana. Estos proyectos se usan juntos para administrar, analizar y visualizar grandes volúmenes de datos estructurados y no estructurados.

- **Elasticsearch** es un motor de búsqueda y análisis distribuido basado en Apache Lucene. Permite agregar registros de todos los sistemas y aplicaciones, analizar estos registros y crear visualizaciones para la supervisión de las aplicaciones y la infraestructura, una solución de problemas más rápida, análisis de seguridad y mucho más.
- **Logstash** es una herramienta de ingesta de datos que permite recopilar datos de diversos orígenes, transformarlos y enviarlos al destino deseado. Gracias a los filtros preconfigurados y a la compatibilidad con más de 200 complementos, Logstash permite incorporar los datos con facilidad, sin importar el origen o el tipo de datos.
- **Kibana** es una herramienta de visualización que permite explorar, analizar y presentar los datos almacenados en Elasticsearch. Ofrece una interfaz gráfica intuitiva que facilita la creación de paneles interactivos con gráficos, mapas, tablas y otros elementos visuales.

La pila ELK es una solución popular para el análisis avanzado del big data, ya que ofrece una alta velocidad, escalabilidad, flexibilidad y rendimiento. Además, tiene una gran comunidad de desarrolladores que contribuyen al mantenimiento y mejora del software. [26][27]

3.4. Otros recursos

3.4.1. Sandbox-as-a-Service

Un sandbox de ciberseguridad proporciona un entorno seguro para abrir archivos sospechosos, ejecutar programas poco fiables o descargar direcciones URL sin afectar a los dispositivos en los que se encuentran. Se puede utilizar en cualquier momento y en cualquier situación para examinar de forma segura un archivo o código que podría ser malicioso antes de que se muestre en los dispositivos, manteniéndolo aislado en todo momento de un PC y de la propia red.

> *Además del sandboxing en servidores localizados, han proliferado recientemente servicios de este tipo en la nube que permiten que las URL, los archivos o el código se prueben bajo demanda en un sandbox virtual completamente independiente del ordenador o de cualquier dispositivo de red. [28]*

Existen multitud de servicios de este tipo, y hemos usado algunos de ellos para nuestro proposito que no es más que revisar los ficheros subidos a nuestro Honeypot con el fin de determinar si son peligrosos o potencialmente dañinos:

- **Hybrid Analysis** - Servicio gratuito de análisis de malware impulsado por Payload Security que detecta y analiza amenazas desconocidas utilizando una tecnología única de Análisis Híbrido.

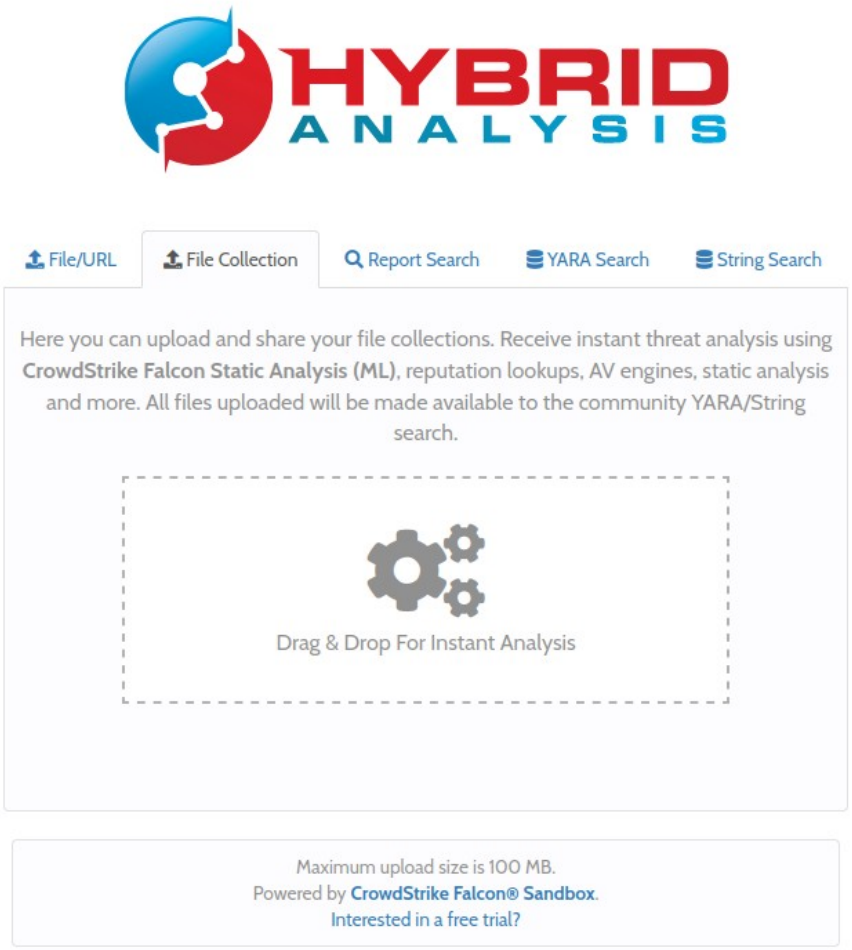


Figura 16. Hybrid Analysis

- **Joebox Cloud** - Analiza el comportamiento de archivos maliciosos incluyendo PEs, PDFs, DOCs, PPTs, XLSs, APKs, URLs y MachOs en Windows, Android y Mac OS X en busca de actividades sospechosas. [29]
- **VirusTotal** - Analiza archivos y URL sospechosos para detectar tipos de malware y compartirlos automáticamente con la comunidad de seguridad.
- **malwr.com** - Servicio gratuito de análisis de malware y comunidad.

3.4.2. Motores de búsqueda (Shodan, Censys, Zoomeye...)

Existen motores de búsqueda especializados en la localización de sistemas y servicios conectados a internet, es decir, lo que buscan son máquinas conectadas a la red. [30]

> *Este tipo de buscadores son bastante útiles para auditores de seguridad informática, ya que les permite buscar en Internet los servidores que estén auditando u otros servicios, y obtener de ellos mucha información que puede resultarles útil. Pueden encontrar la IP, la geolocalización, el país al que pertenece un elemento encontrado, o incluso los puertos y servicios que tengan abiertos.*

Por todo esto, este tipo de herramientas son precisamente muy útiles en el ámbito de la seguridad, ya que se puede conocer hasta qué punto está expuesto un servidor u otras "cosas" conectadas a internet. [31]

En nuestro proyecto comprobaremos si nuestro *Honeypot* aparece en este buscador una vez que lo expondramos en la red y analizaremos qué datos nos arroja al respecto.

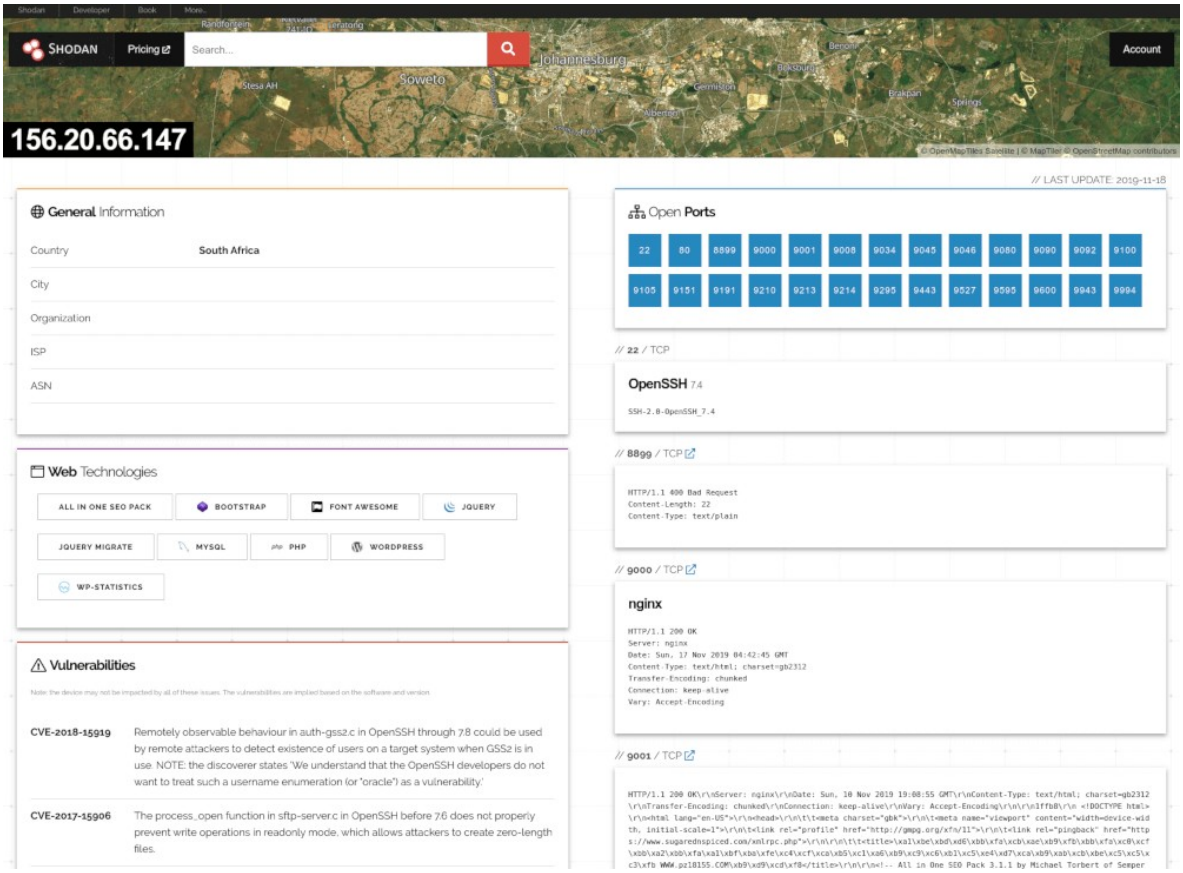


Figura 17. Interfaz Shodan

3.4.3. Maxmind (Geo IP)

GeoIP son una serie de productos, de la empresa MaxMind, que te permiten descubrir información relativa a la posición geográfica de una dirección IP específica. GeoLite y GeoLite2 son una serie de bases de datos gratuitas que nos ayudarán a conocer estos datos. [32]

En nuestro proyecto, mediante Logstash, con la configuración pertinente, consultará la base de datos de GeoIP2 para añadir campos de geolocalización a los logs que reciba de Cowrie.

3.4.4. Termius para iOS

Se trata de un terminal para poder acceder a hosts remotos via SSH u otros. Su fácil manejo y capacidad de organizar las conexiones la convierten en una útil herramienta para administrar servidores de forma remota desde cualquier dispositivo móvil.

Capitulo 4. Diseño e implementación

4.1. Planificación y Diseño

4.1.1. Identificación de objetivos

Para conseguir los objetivos del proyecto, es fundamental definir el tipo de *Honeypot* a implementar, y para ello hemos de filtrar las herramientas de este tipo que permitan emular un dispositivo IoT dentro de todas las opciones existentes.

En el proceso de llevar a cabo este proyecto, es esencial considerar una serie de requisitos fundamentales que permitirán alcanzar los objetivos planteados, desde el engaño a los posibles atacantes, pasando por el correcto funcionamiento y rendimiento del señuelo, hasta la eficiente recopilación de información.

Algunos de los requisitos más significativos a tener en cuenta son:

- Realismo: La credibilidad del *Honeypot* es crucial para minimizar su detección. Por consiguiente, se busca recrear un dispositivo que simule con precisión a los dispositivos presentes en las redes IoT, imitando sus características y comportamientos.

- Rendimiento. Se ha desplegar una estructura básica pero suficiente para alcanzar nuestros objetivos. Para ello tendremos en cuenta el entorno en el cual trabajamos y los requisitos de ejecución ya que nuestro *hardware* tiene unas limitaciones. Cabe decir que es suficiente si se realiza una elección acorde al alcance.

- Recopilación de información relevante: Definir y establecer los datos cruciales a recolectar es esencial, ya que esta información servirá como base para analizar y reconocer nuevas amenazas emergentes en el entorno de las redes IoT.

Desarrollar un *Honeypot* que simule con exactitud un dispositivo IoT no solo implica emular sus características, sino también garantizar la recopilación de datos pertinentes que faciliten un análisis exhaustivo del *malware* que pueda afectar estos dispositivos. Este enfoque asegura no solo la efectividad del engaño a los atacantes, sino también la generación de información valiosa para identificar y mitigar futuras amenazas.

4.1.2. Selección de tecnologías

Primeramente, hemos estudiado diferentes Honeypot que nos dan la posibilidad de simular un dispositivo IoT.

Como ya vimos en el capitulo 3 “Herramientas”, existen diversas herramientas disponibles. En un primer momento se optó por utilizar **T-POT**, el conocido recopilatorio de honeypots, pero debido a sus requisitos HW se descartó. Se pretendía desactivar los honeypots (o dockers) que no necesitáramos pero quedó patente en algunas pruebas iniciales que su rendimiento no era bueno en nuestro equipamiento y se decidió ajustarnos más a los objetivos concretos del proyecto eligiendo uno concreto.

Por tanto, concretando herramientas dedicadas, hemos realizado una comparativa de algunos Honeypot ya presentados en el capítulo anterior, con las ventajas y desventajas (respecto a nuestro objetivo) para valorar la más idónea:

Honeypot	Ventajas	Desventajas
Kippo	- Emula servicio SSH, - Detecta intrusiones y atraques de fuerza bruta - No depende del hardware	- Kippo es una versión más antigua y ya no se actualiza activamente
Cowrie	- Cowrie es un *fork* de Kippo y ha continuado su desarrollo, ofreciendo mejoras, actualizaciones y soporte continuo. - Eficiente recogida de información y de ficheros. - No depende del Hardware.	- No es específica para IoT
IoTPot	- No depende del Hardware.	- No existe repositorio
Telnet IoT Honeypot	- Recoge malware y lo analiza. - No depende del Hardware.	- Solo dispone de servicio Telnet

En base al análisis realizado se descarta **IoTPot**, puesto que no tiene repositorio, resultando imposible su implementación. Tampoco se ha elegido **Telnet IoT HoneyPot**, ya que solo permite Telnet, y, aunque resulta un sistema muy adecuado para la ejecución del proyecto, no se quiere renunciar a otros servicios como SSH, por ejemplo, muy presente en dispositivos IoT.

Si descartamos **Kippo** por ser una versión más antigua que **Cowrie**, se decide utilizar este último, ya que se cuenta con amplia documentación, permite emular servicio SSH, un servicio de login flexible, e incluso, un sistema de ficheros simulado.

Se ha decidido implentar dicho Honeypot en una distribución Ubuntu ya que es totalmente compatible.

Por último, para gestionar la información que dará salida el honeypot se elige utilizar la pila **ELK** (Elasticsearch, Logstash, Kibana) para simplificar y facilitar el análisis de la información obtenido en la solución configurada.

Es una elección basada en multitud de ejemplos encontrados en internet que ponen en muy buen lugar el ELK como herramienta de tratamiento de datos. Sin ir más lejos, está incluido en el propio **T-POT** y eso nos garantizó que era una opción ideal para nuestro proyecto.

4.1.3. Arquitectura del Honeypot

En nuestro proyecto, hemos implementado una arquitectura de Honeypot Cowrie desplegado en una máquina en un entorno doméstico. A continuación, se detallan algunos aspectos técnicos:

- La máquina anfitrión forma parte de la red local. Junto con el resto de dispositivos de la red, están protegidos de ser identificados desde el exterior ya que el router gateway utiliza NAT para asignar direcciones IP privadas a los dispositivos en la red local.
- El router actúa como un punto de entrada y salida de tráfico entre la red local y el Internet. Además, el router cuenta con un firewall que controla y filtra el tráfico, lo que ayuda a proteger los dispositivos internos.
- La máquina con Cowrie se desplegará en la red interna, de forma que el router ZTE actuará como el puerta de enlace y salida a Internet.
- A la máquina Ubuntu se le asignará la IP estática 192.168.1.222. Esto lo haremos modificando el archivo /etc/network/interfaces.d.
- La opción de DMZ (Zona Desmilitarizada) del router se configura para alojar la máquina virtual, proporcionando un nivel adicional de seguridad al aislar la red doméstica principal de posibles ataques dirigidos al Honeypot.

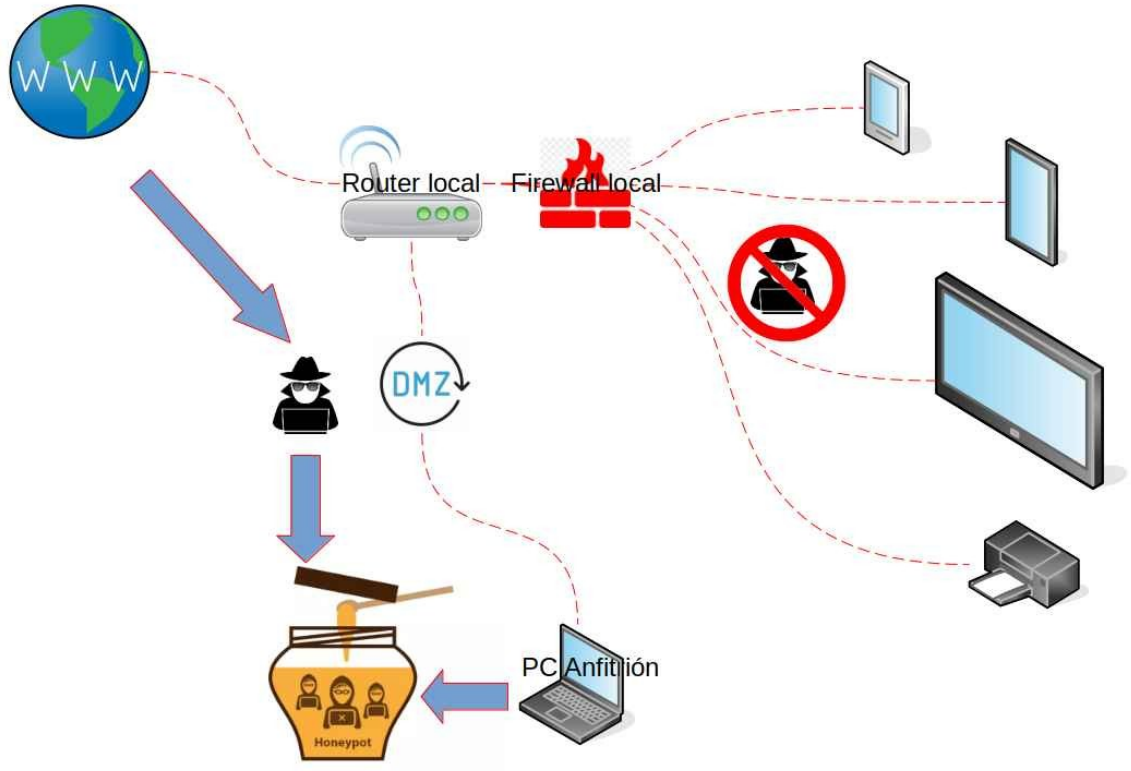


Figura 18. Arquitectura de red

4.2. Despliegue del sistema:

En este apartado se explica el proceso de despliegue de las herramientas utilizadas en la ejecución del proyecto y que consta de varios hitos:

a) Honeypot Cowrie:

Cowrie se instala y se configura en la máquina Ubuntu Linux. Se configura Cowrie para simular servicios y puertos comúnmente utilizados por los dispositivos IoT, proporcionando así una trampa atractiva para posibles intrusos.

El Honeypot está diseñado para emular un sistema SSH (Secure Shell) falso y registra las interacciones con los posibles atacantes. En el anexo se explica la instalación completa pero hay que señalar algunas configuraciones específicas tanto en sistema operativo como en la propia herramienta que hemos realizado para nuestro caso:

- ✔ La configuración de Cowrie se almacena en cowrie.cfg.dist y cowrie.cfg (ubicados en cowrie/etc). Ambos archivos se leen en el arranque, donde las entradas de cowrie.cfg tienen prioridad. El archivo .dist puede ser sobrescrito por actualizaciones, pero cowrie.cfg no será tocado. Para ejecutar con una configuración estándar, no hay necesidad de cambiar nada. Como nosotros queríamos habilitar telnet hemos modificado cowrie.cfg con este contenido:

```
[telnet]
enabled = true
```

- ✔ Es importante recalcar, que antes de poner en marcha Cowrie dentro de Ubuntu hay que introducir una serie de reglas en el firewall del sistema operativo (IPTables) que redirijan el tráfico de los puertos tcp 22/23 a los puertos 2222/2223 respectivamente, que son con los que trabaja Cowrie. De esta manera, cuando dichos servicio SSH o Telnet del sistema reciban paquetes, estos serán redirigidos a Cowrie.

```
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
```

```
root@fercho-Latitude-E5420:/home/fercho# sudo iptables -t nat -L PREROUTING
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DOCKER     all  --  anywhere             anywhere             ADDRTYPE match dst-type LOCAL
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:ssh redir ports 2222
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:telnet redir ports 2223
```

Figura 19. Consulta de reglas iptables

b) Configuración del router ZTE:

La opción de DMZ se habilita para la dirección IP de la máquina que aloja el *honeypot*, permitiendo que cualquier tráfico no solicitado se dirija a dicho sistema.

En nuestro router ZTE ZXHN H298Q hemos añadido la IP 192.168.1.222 a la DMZ.

Figura 20. Interfaz router: Añadiendo host a la DMZ

La ubicación en la zona desmilitarizada (DMZ) permite por un lado juntar en el mismo segmento a nuestros servidores de producción con el Honeypot y por el otro controlar el peligro que añade su uso, ya que ***tiene un firewall que lo aísla de resto de nuestra red local***

Además, se abren los puertos tcp 22 y 23 (ssh y telnet) para el host que aloja el Honeypot.

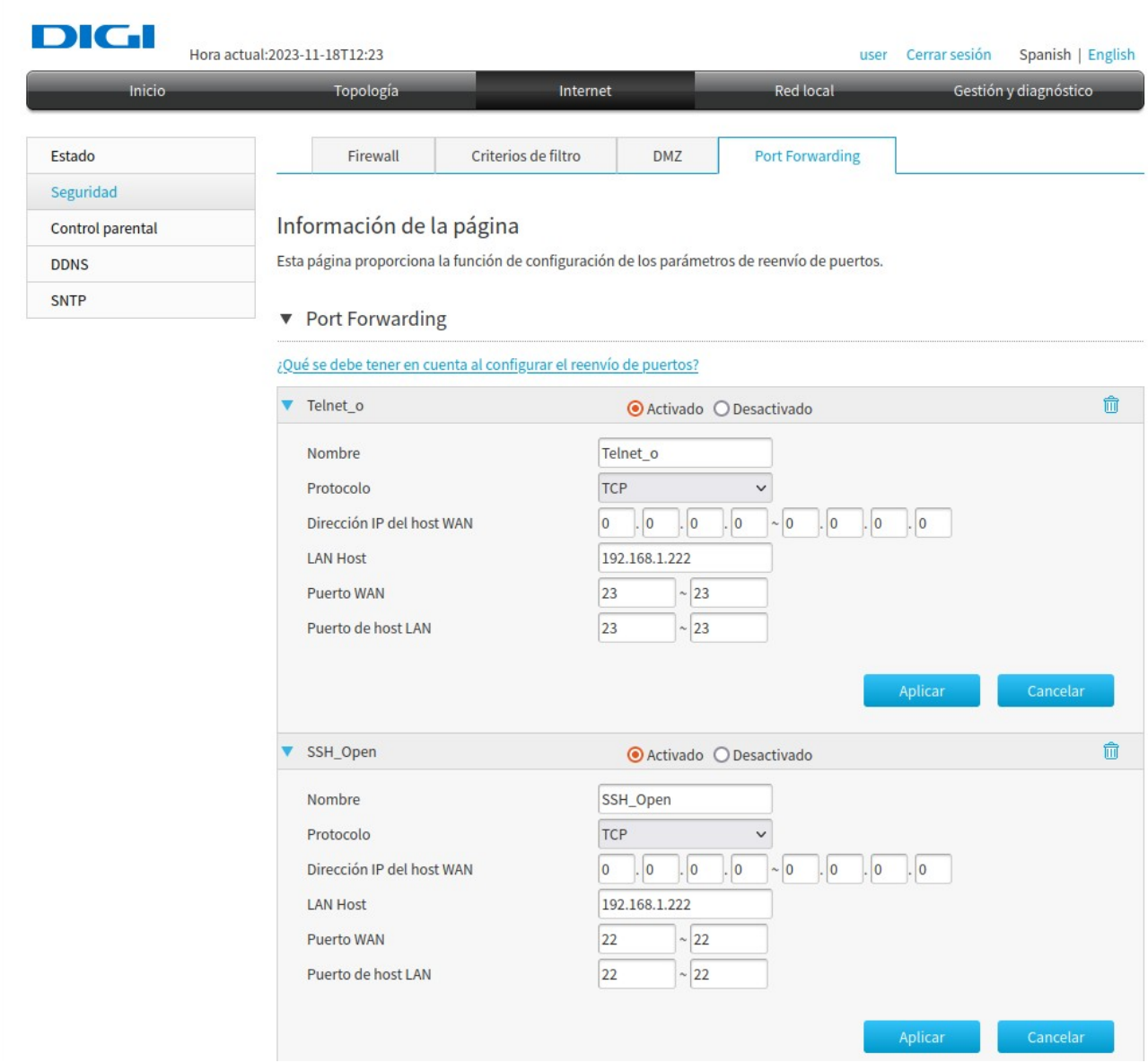


Figura 21. Interfaz Router: redirigir puertos 22 y 23

c) Pila ELK o Elastic Stack

Una vez instaladas las tres aplicaciones ElasticSearch, Logstash y Kibana (ver Anexo) se procede a ajustar los parámetros para poder recoger los datos del Honeypot y también para personalizar dichas herramientas acorde a nuestras necesidades y preferencias.

Tras la instalación de Elasticsearch, obtenemos información sobre seguridad donde se nos facilita el password para el superusuario entre otros datos importantes:

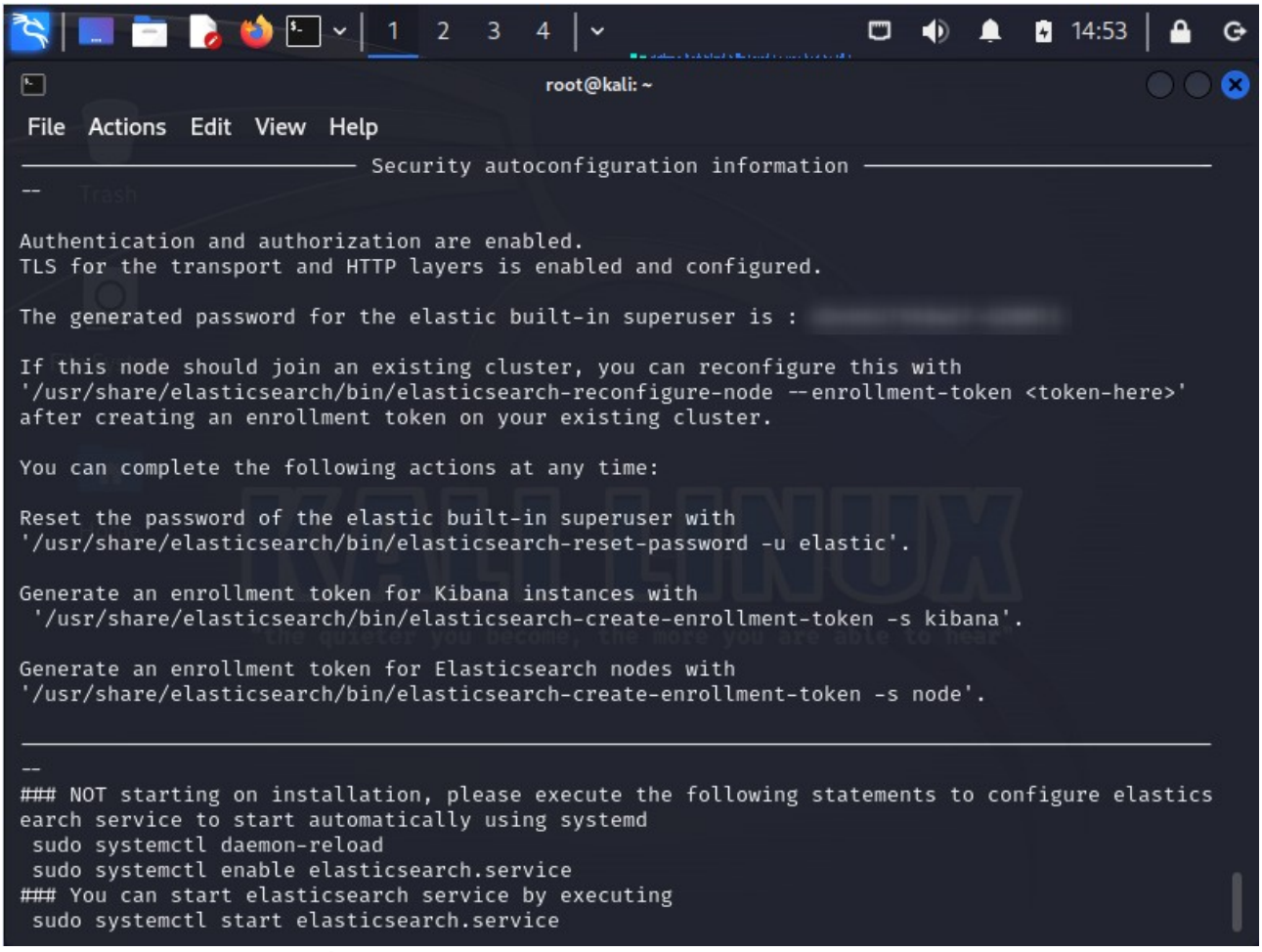


Figura 22. Información de seguridad Elasticsearch

✔ Para Logstash crearemos un fichero de configuración que definirá la entrada y salida de los datos además de aplicar filtros y añadir información sobre geolocalización. Vista del fichero */etc/logstash/conf.d/logstash-cowrie.conf*:

```
  #Obtener logs recopilados por cowrie en los ficheros con extension json
  input {
         file{
           path => "/home/cowrie/cowrie/var/log/cowrie/*json"
  #               path => "/tmp/cowrie.json"
                 start_position => beginning
                 codec => json
                 type => "cowrie"
          }
  }

  #Filtrado de los datos recibidos
  filter{
         if [type] == "cowrie" {
           date {
           #Aplicar formato ISO8601 a la marca temporal
                        match => [ "timestamp", "ISO8601" ]
                 }

           mutate {
           #Renombrar nombre de campos
                        rename => {"dst_port" => "dest_port" "dst_ip" =>"dest_ip" }
           }
         }
  #Añadir geolocalización y el ASN de la IP de Origen con GeoIP
          if [src_ip] {
                 geoip {
                        cache_size => 10000
                        source => "src_ip"
                  target => "geoip"
                        database =>
"/usr/share/logstash/vendor/bundle/jruby/3.1.0/gems/logstash-filter-geoip-7.2.13-java/
vendor/GeoLite2-City.mmdb"
                 }
           }
           geoip {
                 cache_size => 10000
                 source => "src_ip"
             target => "geoip"
                 database =>
"/usr/share/logstash/vendor/bundle/jruby/3.1.0/gems/logstash-filter-geoip-7.2.13-java/
vendor/GeoLite2-ASN.mmdb"
           }
           translate {
                 refresh_interval => 86400
                 field => "src_ip"
                 destination => "ip_rep"
           }
  }

  #Enviar a elasticsearch logs capturados en formato json ya filtrados
  output{
         elasticsearch{

  }

  #Mostrar por pantalla los logs enviados a Elasticsearch aplicando codec rubydebug
         stdout{
                codec =>rubydebug
       }
```

```
file {
            path => "/tmp/cowrie-logstash.log"
            codec => json
        }
}
```

4.3. Puesta en marcha y pruebas

En este apartado vamos a explicar como se han ido poniendo en marcha las diferentes herramientas desplegadas y las pruebas de funcionamiento pertinentes además de algunas de seguridad adicionales.

Ponemos en marcha Cowrie con los puertos redireccionados 22→ 2222 y 23→ 2223.

```
fercho@fercho-Latitude-E5420:~$ sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:ssh redir ports 2222
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:telnet redir ports 2223
```

Figura 23. Puertos redirigidos con iptables

```
cowrie@fercho-Latitude-E5420:~/cowrie$ bin/cowrie start
Using default Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd  --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:106: CryptographyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning: CAST5 has been deprecated
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:115: CryptographyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/home/cowrie/cowrie/cowrie-env/lib/python3.10/site-packages/twisted/conch/ssh/transport.py:116: CryptographyDeprecationWarning: CAST5 has been deprecated
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
cowrie@fercho-Latitude-E5420:~/cowrie$
```

Figura 24. Iniciando Cowrie

Lo que ocurre es que cuando conectamos desde la máquina anfitrión por ssh o telnet, en realidad, estamos accediendo al entorno simulado del honeypot:

via SSH

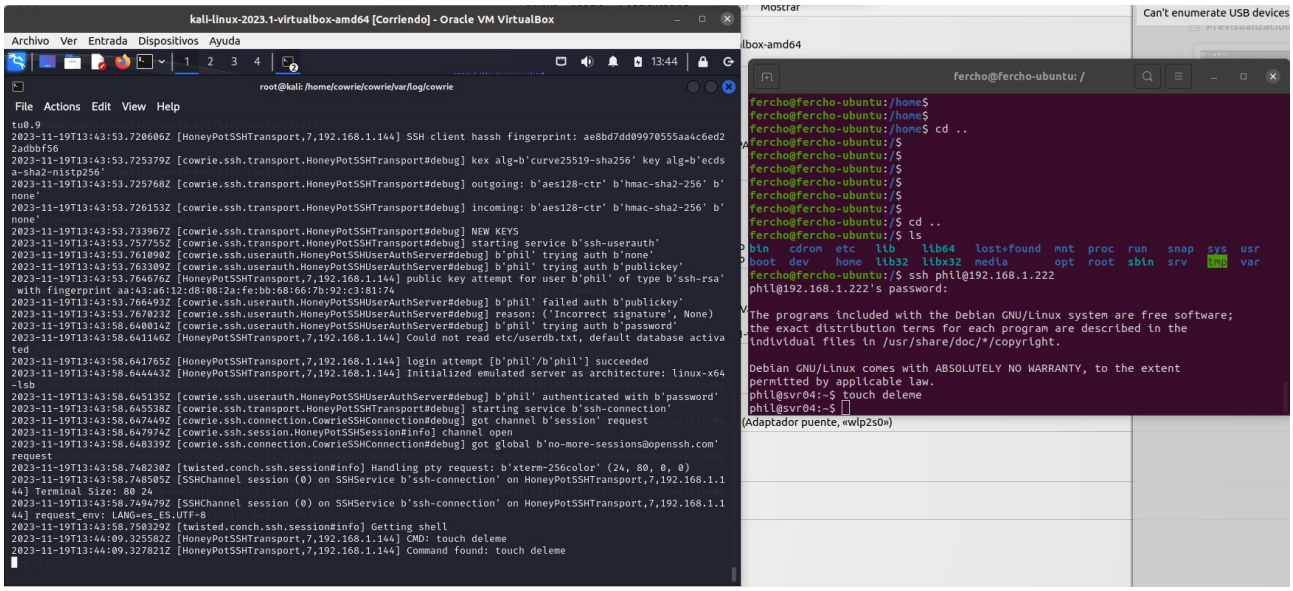


Figura 25. Conexión al honeypot por ssh

via TELNET:

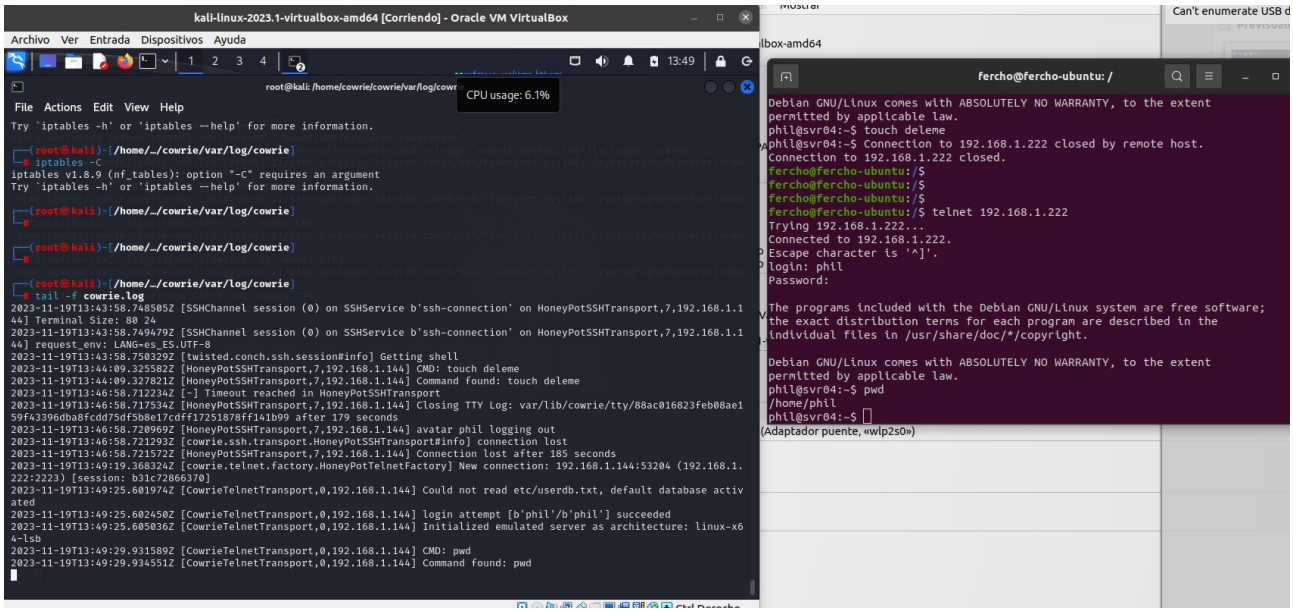


Figura 26. conexión al honeyport por telnet

Iniciamos el servicio Elasticsearch y comprobamos que está operativo:

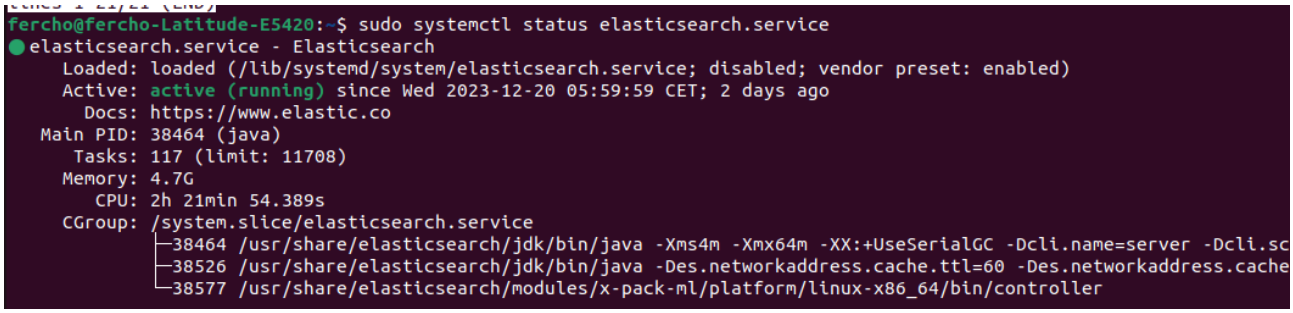


Figura 27. Iniciando elasticsearch.service

Con curl podremos comprobar que Elasticsearch genera ficheros JSON, y por tanto, está operativo:

```
fercho@fercho-Latitude-E5420:~$ curl localhost:9200
{
  "name" : "fercho-Latitude-E5420",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "cv8Q975YQZaUDAOmw6-7og",
  "version" : {
    "number" : "8.11.3",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "64cf052f3b56b1fd4449f5454cb88aca7e739d9a",
    "build_date" : "2023-12-08T11:33:53.634979452Z",
    "build_snapshot" : false,
    "lucene_version" : "9.8.0",
    "minimum_wire_compatibility_version" : "7.17.0",
    "minimum_index_compatibility_version" : "7.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

Figura 28. Comprobacion de que elasticsearch está iniciado

Después arrancamos el servicio Logstash y Kibana mediante

```
systemctl start logstash kibana
```

Kibana inicializado, se puede verificar en localhost:5601

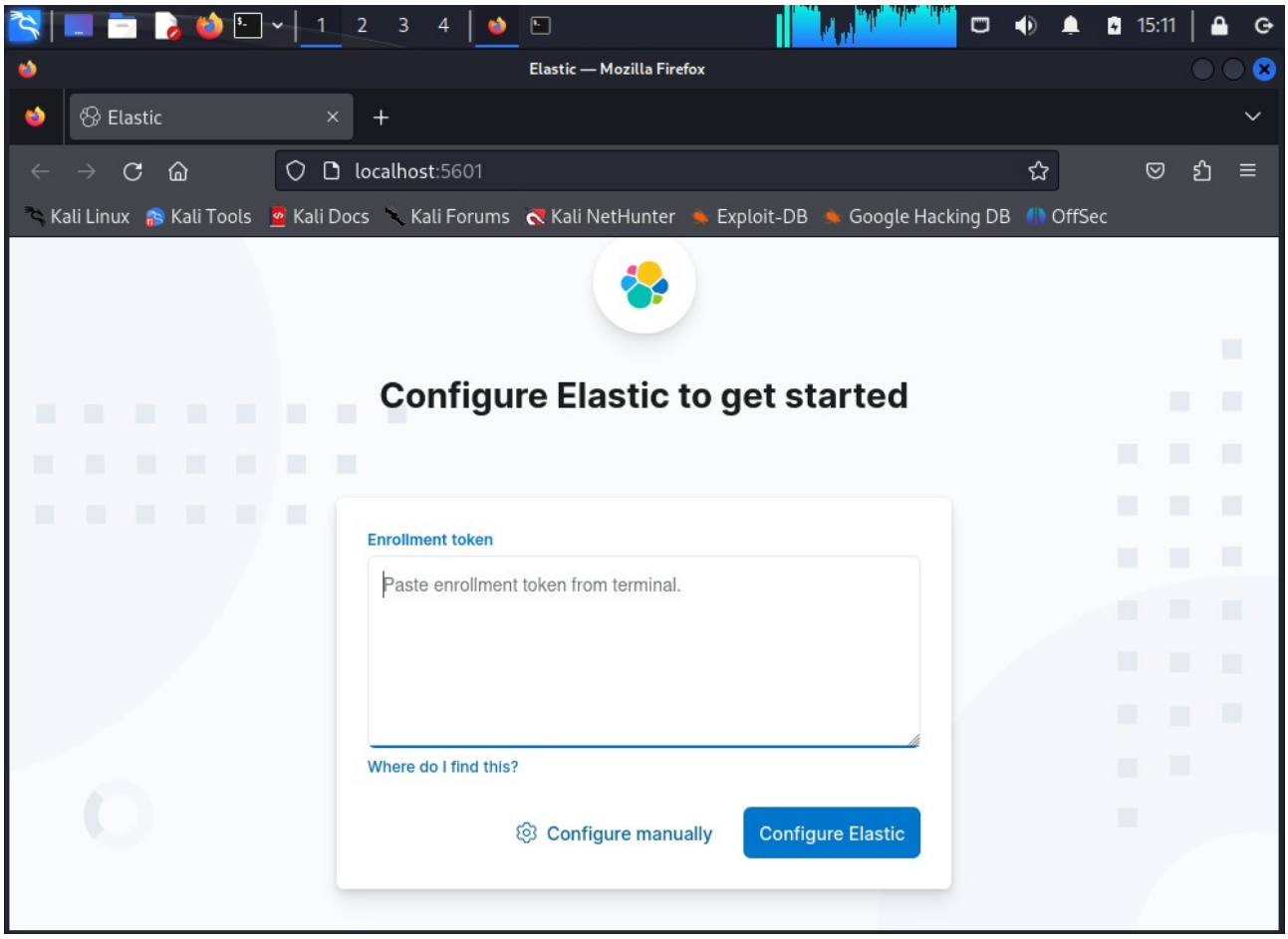


Figura 29. Comprobación de que Kibana está iniciado

Podemos detectar los diferentes servicios en la lista de puertos abiertos del equipo. Siendo 9200 elasticsearch, 5601 kibana y 2222-2223 cowrie:

```
fercho@fercho-Latitude-E5420:~$ sudo netstat -tupl
Conexiones activas de Internet (solo servidores)
Proto  Recib Enviad Dirección local         Dirección remota       Estado       PID/Program name
tcp        0      0 localhost:5601          0.0.0.0:*              ESCUCHAR     29951/node
tcp        0      0 localhost:domain        0.0.0.0:*              ESCUCHAR     555/systemd-resolve
tcp        0      0 localhost:ipp           0.0.0.0:*              ESCUCHAR     118405/cupsd
tcp        0      0 0.0.0.0:2222            0.0.0.0:*              ESCUCHAR     107853/python
tcp        0      0 0.0.0.0:2223            0.0.0.0:*              ESCUCHAR     107853/python
tcp6       0      0 ip6-localhost:9300      [::]:*                 ESCUCHAR     38526/java
tcp6       0      0 ip6-localhost:ipp       [::]:*                 ESCUCHAR     118405/cupsd
tcp6       0      0 [::]:9200               [::]:*                 ESCUCHAR     38526/java
tcp6       0      0 localhost:9300          [::]:*                 ESCUCHAR     38526/java
tcp6       0      0 localhost:9600          [::]:*                 ESCUCHAR     108211/java
```

Figura 30. Comprobación de puertos en escucha

Una vez estando todo el sistema *honeypot* en marcha en la máquina 192.168.1.222, vamos a intentar acceder desde diferentes ubicaciones al propio equipo via telnet y ssh tal y como si fuéramos atacantes.

Primeramente, probamos desde otra máquina en la misma red local.

Entramos con el usuario que hay configurado y podemos observar en el cowrie.log que se comienzan a registrar la interacción con el honeypot:

```
fercho@fercho-ubuntu:~$ ssh phil@192.168.1.222
phil@192.168.1.222's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
phil@svr04:~$ pwd
/home/phil
phil@svr04:~$
```

Figura 31. Conexión al honeypot cowrie

```
┌──(cowrie㉿kali)-[~/cowrie/var/log/cowrie]
└─$ tail -vf cowrie.log
==> cowrie.log <==
2023-12-08T14:26:12.754762Z [twisted.conch.ssh.session#info] Getting shell
2023-12-08T14:26:14.189101Z [HoneyPotSSHTransport,5,192.168.1.144] CMD:
2023-12-08T14:26:14.329766Z [HoneyPotSSHTransport,5,192.168.1.144] CMD:
2023-12-08T14:26:53.105228Z [HoneyPotSSHTransport,5,192.168.1.144] CMD: pwd
2023-12-08T14:26:53.106659Z [HoneyPotSSHTransport,5,192.168.1.144] Command found: pwd
2023-12-08T14:29:12.138275Z [-] Timeout reached in HoneyPotSSHTransport
2023-12-08T14:29:12.139035Z [HoneyPotSSHTransport,5,192.168.1.144] Closing TTY Log: var/lib/cowrie/tty/ccca8a10f1e1afcd926c1130979841529c5d0ecb97a4c0f60d735314d7703f77 after 179 seconds
2023-12-08T14:29:12.139961Z [HoneyPotSSHTransport,5,192.168.1.144] avatar phil logging out
2023-12-08T14:29:12.140070Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2023-12-08T14:29:12.140161Z [HoneyPotSSHTransport,5,192.168.1.144] Connection lost after 185 seconds
2023-12-08T14:31:51.233425Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.1.144:39000 (192.168.1.222:2222) [session: 7e2a700dd1b5]
2023-12-08T14:31:51.234154Z [HoneyPotSSHTransport,6,192.168.1.144] Remote SSH version: SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.9
2023-12-08T14:31:51.241578Z [HoneyPotSSHTransport,6,192.168.1.144] SSH client hassh fingerprint: ae8bd7dd09970555aa4c6ed22adbbf56
2023-12-08T14:31:51.242900Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'curve25519-sha256' key alg=b'ecdsa-sha2-nistp256'
2023-12-08T14:31:51.243433Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes128-ctr' b'hmac-sha2-256' b'none'
2023-12-08T14:31:51.243662Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha2-256' b'none'
2023-12-08T14:31:51.254976Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2023-12-08T14:31:51.262186Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2023-12-08T14:31:51.268020Z [cowrie.ssh.useradth.HoneyPotSSHUserAuthServer#debug] b'phil' trying auth b'none'
2023-12-08T14:31:51.271960Z [cowrie.ssh.useradth.HoneyPotSSHUserAuthServer#debug] b'phil' trying auth b'publickey'
2023-12-08T14:31:51.272467Z [HoneyPotSSHTransport,6,192.168.1.144] public key attempt for user b'phil' of type b'ssh-rsa' with fingerprint aa:43:a6:12:d8:08:2a:fe:bb:68:66:7b:92:c3:81:74
2023-12-08T14:31:51.273191Z [cowrie.ssh.useradth.HoneyPotSSHUserAuthServer#debug] b'phil' failed auth b'publickey'
2023-12-08T14:31:51.273361Z [cowrie.ssh.useradth.HoneyPotSSHUserAuthServer#debug] reason: ('Incorrect signature', None)
2023-12-08T14:31:56.517616Z [cowrie.ssh.useradth.HoneyPotSSHUserAuthServer#debug] b'phil' trying auth b'password'
2023-12-08T14:31:56.518057Z [HoneyPotSSHTransport,6,192.168.1.144] Could not read etc/userdb.txt, default database activated
2023-12-08T14:31:56.518268Z [HoneyPotSSHTransport,6,192.168.1.144] login attempt [b'phil'/b'Phil California'] succeeded
2023-12-08T14:31:56.518829Z [HoneyPotSSHTransport,6,192.168.1.144] Initialized emulated server as architecture: linux-x64-lsb
2023-12-08T14:31:56.570621Z [cowrie.ssh.useradth.HoneyPotSSHUserAuthServer#debug] b'phil' authenticated with b'password'
2023-12-08T14:31:56.570984Z [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2023-12-08T14:31:56.575426Z [cowrie.ssh.connection.CowrieSSHConnection#debug] got channel b'session' request
2023-12-08T14:31:56.575820Z [cowrie.ssh.session.HoneyPotSSHSession#info] channel open
2023-12-08T14:31:56.576129Z [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-more-sessions@openssh.com' request
2023-12-08T14:31:57.078408Z [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256color' (23, 79, 0, 0)
2023-12-08T14:31:57.078655Z [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,6,192.168.1.144] Terminal Size: 79 23
2023-12-08T14:31:57.079319Z [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,6,192.168.1.144] request_env: LANG=es_ES.UTF-8
2023-12-08T14:31:57.079840Z [twisted.conch.ssh.session#info] Getting shell
2023-12-08T14:32:02.562555Z [HoneyPotSSHTransport,6,192.168.1.144] CMD: pwd
2023-12-08T14:32:02.563186Z [HoneyPotSSHTransport,6,192.168.1.144] Command found: pwd
2023-12-08T14:34:56.636832Z [-] Timeout reached in HoneyPotSSHTransport
2023-12-08T14:34:57.106723Z [HoneyPotSSHTransport,6,192.168.1.144] Closing TTY Log: var/lib/cowrie/tty/d451b4ca972da8b8815888326f839f9942f4b7a6ac633d3531a927809bd7e595 after 180 seconds
2023-12-08T14:34:57.238858Z [HoneyPotSSHTransport,6,192.168.1.144] avatar phil logging out
2023-12-08T14:34:57.238984Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2023-12-08T14:34:57.239063Z [HoneyPotSSHTransport,6,192.168.1.144] Connection lost after 186 seconds
```

Figura 32. Fichero cowrie.log registra interacciones

Creamos un fichero y lo subimos al servidor.

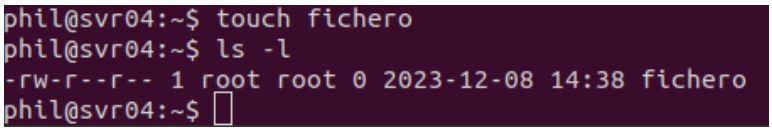


Figura 33. Creacion de fichero en remoto

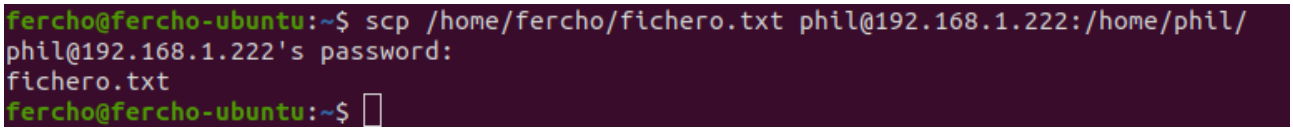


Figura 34. Subida mediante scp

En el sistema trampa ya podemos observar el fichero alojado en *var/lib/cowrie/downloads/*

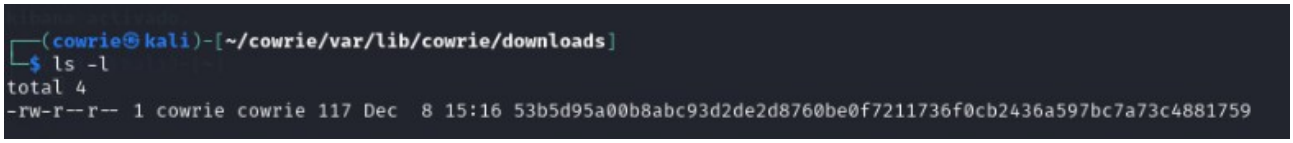


Figura 35. Directorio donde se almacenan ficheros volcados a Cowrie

Exponemos el *honeypot* en internet incluyéndolo en la DMZ y abriendo los puertos como ya se explicó anteriormente.

Además, hemos tenido que contactar con el operador DIGI y activar el servicio Conexion Plus por 1€ al mes. Esto ha sido necesario para salir de la CG-NAT y así tener una IP propia a la que intentar conectar, ya que con CG-NAT estabamos compartiendo dirección IP con otros clientes y no podíamos publicar servicios como deseamos.

Para probarlo, configuramos el host cliente para conectarnos a la red movil y así probar a conectarnos desde el exterior al servidor. Para ello hemos de conocer la ip que tenemos, esto lo averiguamos con el consultando mediante *curl ifconfig.me* o cualquier web tipo *howismyip*.

Probamos a conectarnos, esta vez desde el exterior, y comprobamos que igualmente se registran las interacciones en el *cowrie.log.*

Para hacer pruebas y comprobar periódicamente que el honeypot sigue activo, he instalado en mi dispositivo móvil la aplicación Termius y he configurado dos conexiones. Una para usarla desde mi red local (conectado por wifi) y otra para conectar desde el exterior apuntando a mi ip pública:

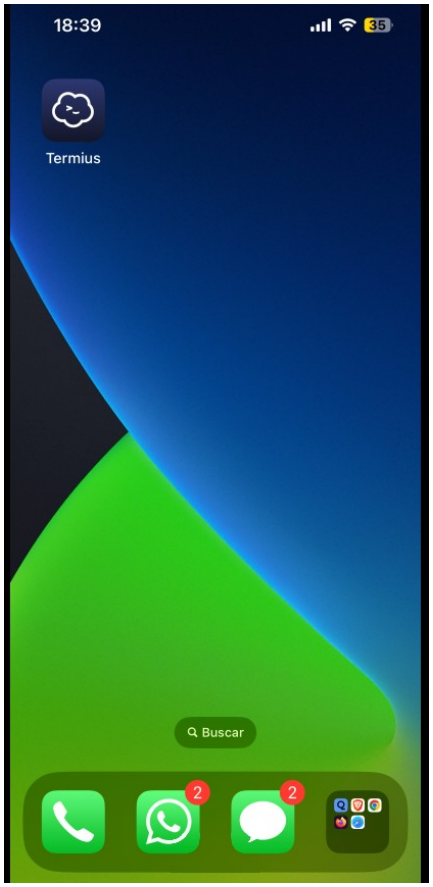


Figura 37. App para iOS

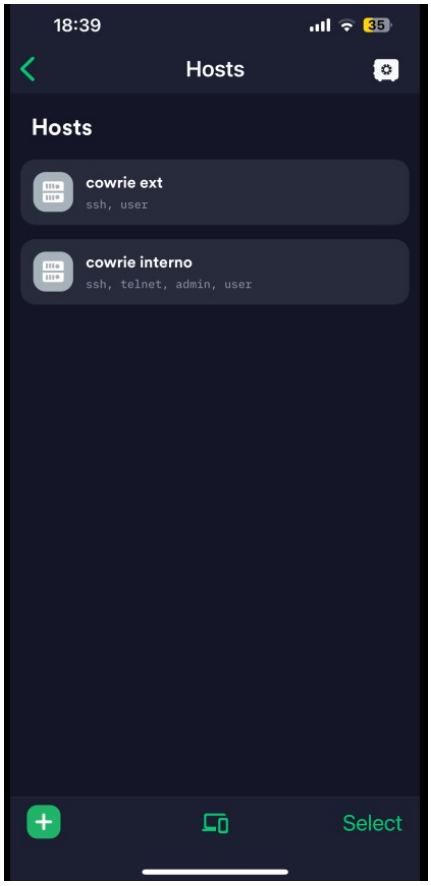


Figura 38. Conexiones configuradas Termius

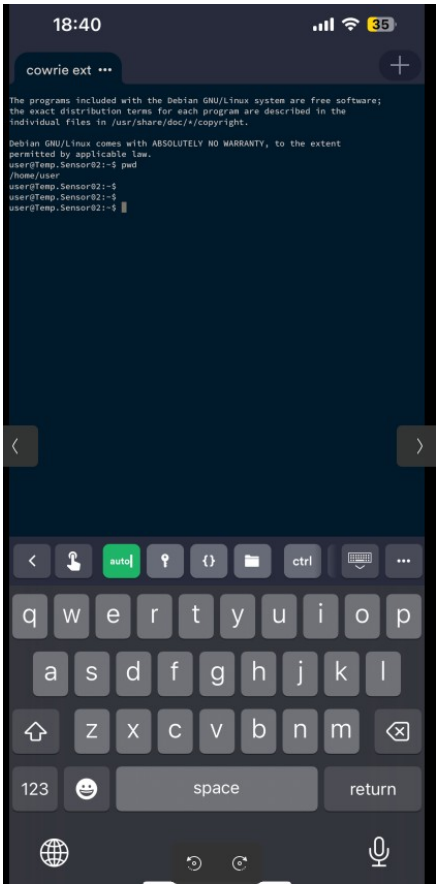


Figura 36. Termius conectando al honeypot

Usamos la web *Censys* para chequear nuestra exposición pública y observamos información sobre nuestro host con puertos abiertos 22 y 23:

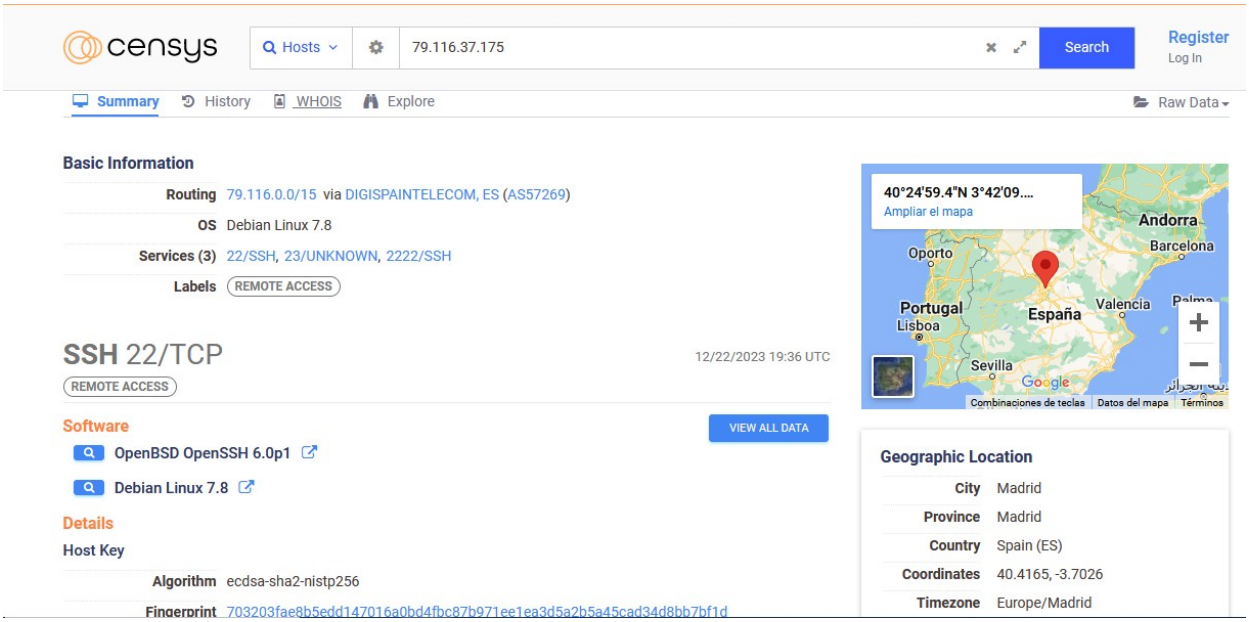


Figura 39. Consulta en web censys

Adicionalmente, hemos alterado la configuracion por defecto del sistema trampa para evitar que sea detectado fácilmente:

```
cowrie@fercho-Latitude-E5420:~/cowrie/honeyfs/etc$ ls -l
total 40
-rw-rw-r-- 1 cowrie logstash  540 dic 22 21:26 group
-rw-rw-r-- 1 cowrie logstash    9 dic 20 04:15 host.conf
-rw-rw-r-- 1 cowrie logstash    6 dic 20 04:15 hostname
-rw-rw-r-- 1 cowrie logstash  184 dic 20 04:15 hosts
-rw-rw-r-- 1 cowrie logstash 2013 dic 20 04:15 inittab
-rw-rw-r-- 1 cowrie logstash   22 dic 22 21:05 issue
-rw-rw-r-- 1 cowrie logstash  286 dic 20 04:15 motd
-rw-rw-r-- 1 cowrie logstash  872 dic 20 21:22 passwd
-rw-rw-r-- 1 cowrie logstash   38 dic 20 04:15 resolv.conf
-rw-rw-r-- 1 cowrie logstash  752 dic 22 20:34 shadow
cowrie@fercho-Latitude-E5420:~/cowrie/honeyfs/etc$ cat hostname
svr04
cowrie@fercho-Latitude-E5420:~/cowrie/honeyfs/etc$ cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
sshd:x:101:65534::/var/run/sshd:/usr/sbin/nologin
robert:x:1000:1000:robert,,,:/home/robert:/bin/bash
julia:x:1000:1000:julia,,,:/home/julia:/bin/bash
```

Figura 40. Cambios en ficheros Cowrie

Capitulo 5. Resultados obtenidos

5.1. Resultados generales

- Durante los 4 días de exposición se han registrado un total de 18613 ataques.
- Se han registrado ataques desde todos los continentes.
- Se ha recogido 1 fichero que podría resultar malware subido por los atacantes. Los ficheros se han alojado en */cowrie/var/lib/cowrie/downloads.*
- Los intentos de conexión fueron destinados a los puertos 22 y 23 (SSH/Telnet), aunque también se han detectado ataques a otros puertos conocidos.
- En los intentos de conexión se han usado algunas combinaciones de usuario-contraseña bastante habituales por defecto en los dispositivos y algunas muy sencillas usadas como mala práctica por algunos usuarios.

Mediante Kibana, hemos podido crear filtros y vistas personalizadas de los datos para así interpretarlos fácilmente:

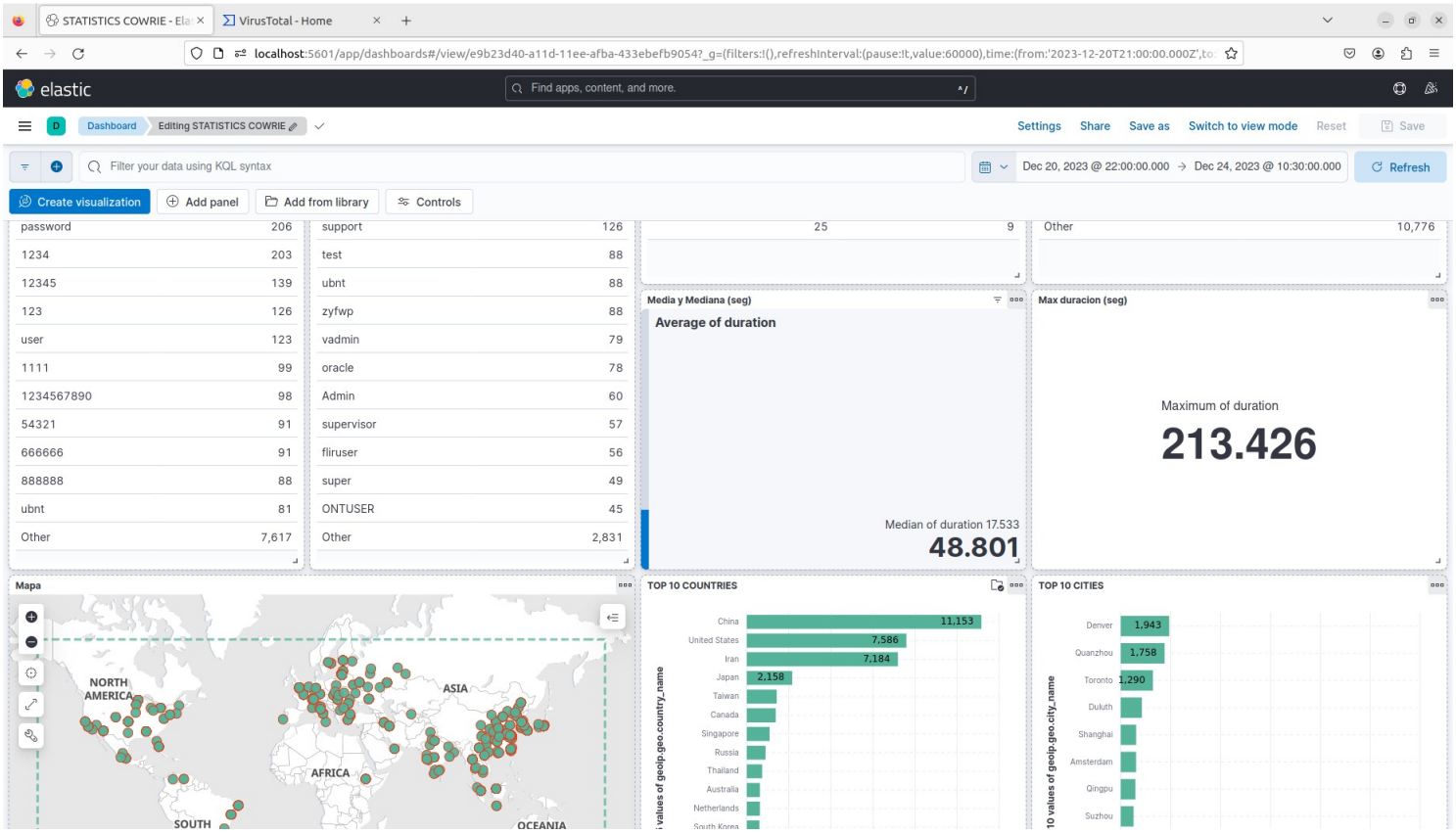


Figura 41. Vista personalizada de datos en Kibana

En la gráfica temporal por horas se puede apreciar un volumen alto de ataques con algunos picos puntuales.

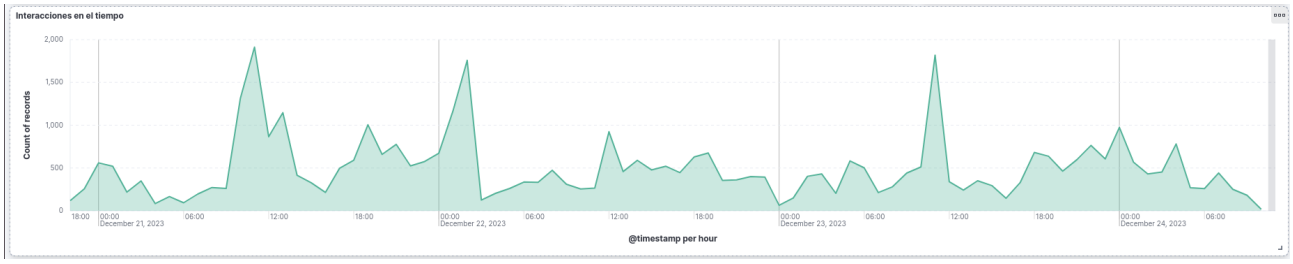


Figura 42. Gráfica temporal de ataques

Estos son los usuarios y contraseñas más utilizadas por los *hackers* para intentar acceder al dispositivo. Como ya comentamos, se suelen emplear las más típicas y sencillas.

TOP 10 USERS

Top 15 values of usernam	Count of records
root	4,484
admin	1,876
guest	190
user	188
support	126
test	88
ubnt	88
zyfwp	88
vadmin	79
oracle	78
Admin	60
supervisor	57
fliruser	56
super	49
ONTUSER	45
Other	2,831

Figura 44. Top 15 usuarios

TOP 10 PASSW

Top 15 values of passv	Count of records
admin	474
123456	439
root	235
(empty)	217
password	206
1234	203
12345	139
123	126
user	123
1111	99
1234567890	98
54321	91
666666	91
888888	88
ubnt	81
Other	7,617

Figura 43. Top 15 contraseñas

Ya que nuestra configuración no ha permitido un acceso “libre” con cualquier usuario y contraseña, algunos intentos de conexión han fracaso. Aun así, las credenciales que hemos establecidos eran muy débiles y predecibles por lo que casi la mitad de los intentos han tenido éxito. Aquí podemos apreciar dichos datos:

Eventos

Top 4 values of eventid	Count of records
cowrie.login.failed	9,708
cowrie.session.closed	8,905
cowrie.session.connect	8,905
cowrie.client.version	3,701
Other	10,776

Figura 45. Recuento de eventos

5.3 Direcciones IP y puertos.

En las siguientes ilustraciones se puede ver el Top 10 de las direcciones IP desde las que se han recibido los ataques. No describen ningún patrón claro.

top 10 ips de origen

Top 10 values of src_ip	Count of records
151.235.196.1	3,500
151.247.205.211	2,436
170.64.172.70	2,081
164.92.89.60	1,925
222.77.96.62	1,747
85.133.237.72	1,195
138.197.132.230	1,164
43.153.42.200	1,005
128.199.79.88	968
178.62.234.187	608
Other	25,366

Figura 46. Top 10 direcciones de origen

Top 10 values of src_port	Count of src_port
61,000	33
0	13
65,105	12
64,001	8
10,000	5
33,216	4
35,194	4
46,262	4
49,244	4
49,912	4
Other	9,008

Figura 47. Top 10 puertos de origen

5.4 Países, ciudades y ASN

En las siguientes ilustraciones se muestran el Top 10 de los países, ciudades y ASN (Número de Sistema Autónomo: grupo de redes de direcciones IP gestionadas por uno o más operadores de red) de los que se han recibido los ciberataques.

En términos globales, la gran mayoría de las interacciones principalmente provienen de China, Estados Unidos e Irán, si bien es cierto que se observan multitud de ellas segregadas por miles de lugares del mundo.

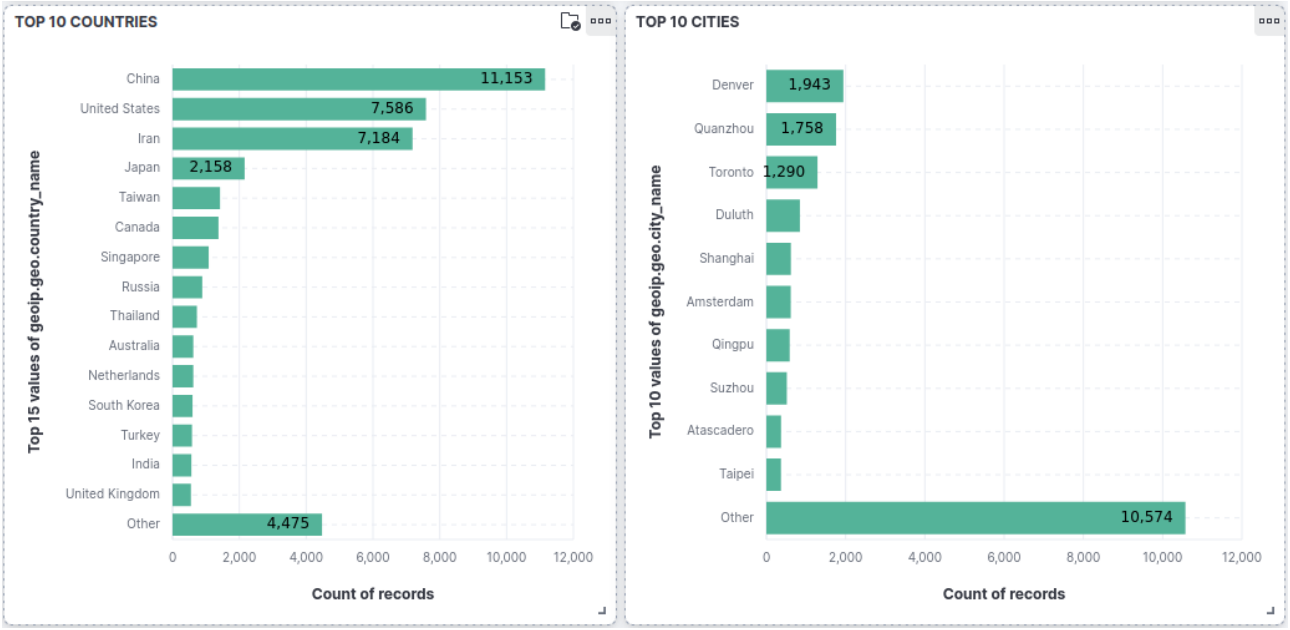


Figura 48. Top 10 paises y ciudades de origen

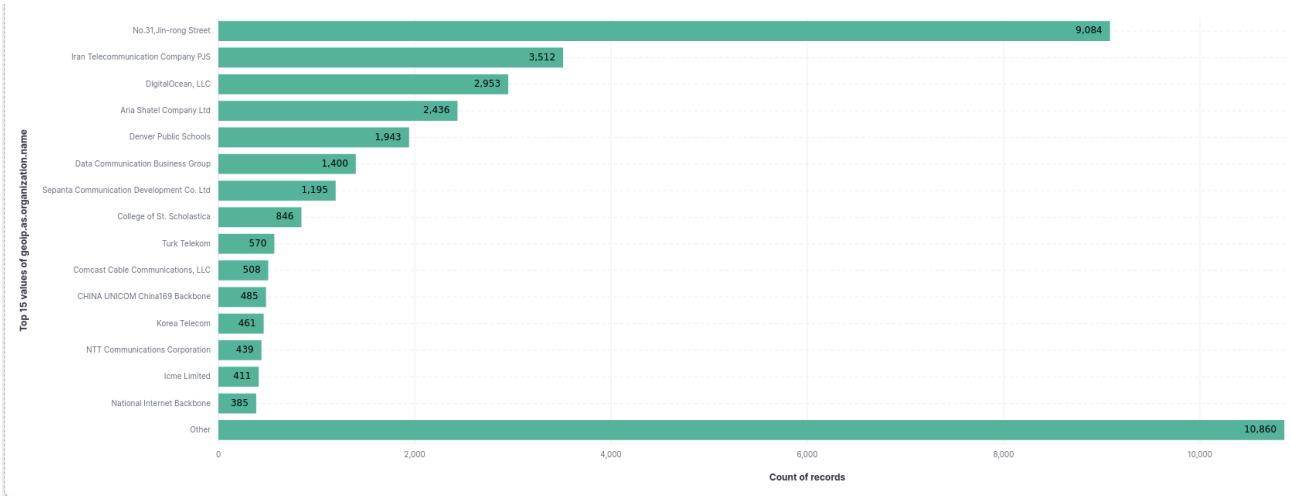


Figura 49. Top 10 ASN

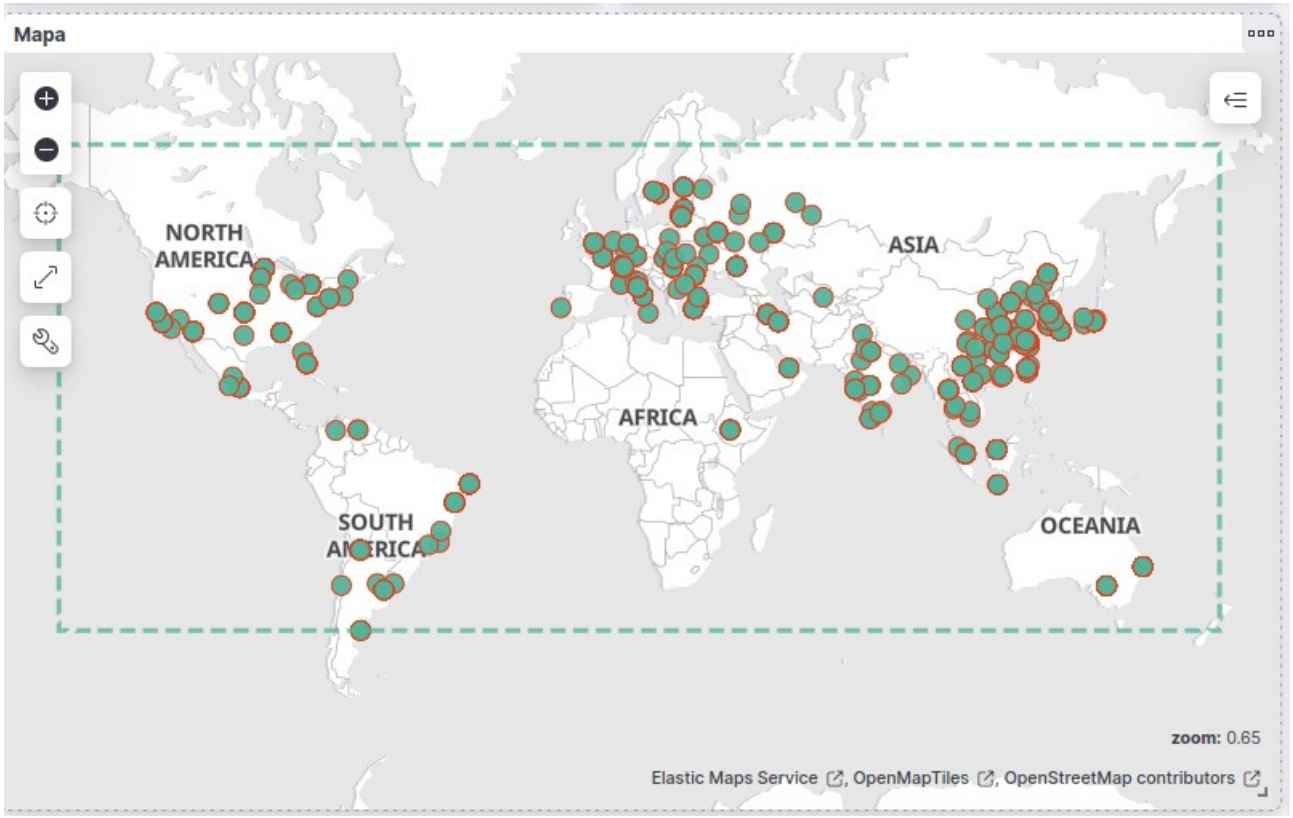


Figura 50. Mapa de origen de ataques

5.5 Protocolo y puertos de destino

En las siguientes ilustraciones se puede ver cuál es el protocolo que ha recibido más ataques, juntos con los puertos de destino que han sido atacados. También, se ha registrado intentos en los puertos 80 (HTTP), 443 (HTTPS) y 25 (SMTP). En la segunda gráfica se comparan los dos protocolos más empleados.

Puertos

Top 5 values of dest_port	Count of records
2,223	4,826
2,222	4,079
80	294
443	74
25	9

Figura 51. Puertos pretendidos

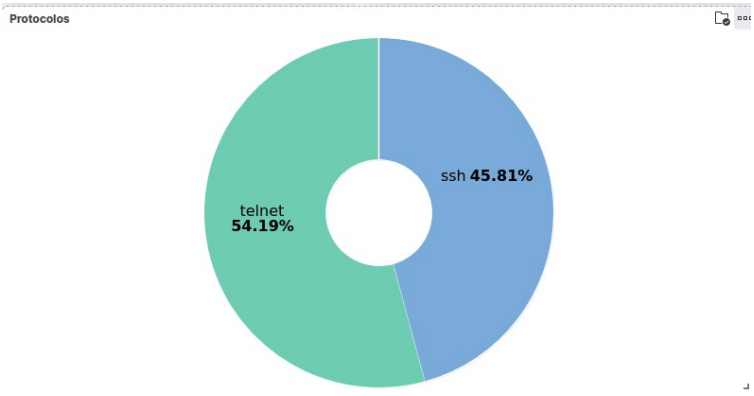


Figura 52. Reparto conexiones Telnet-SSH

5.6 Comandos utilizados

En la siguiente ilustración se pueden observar los comandos introducidos por los atacantes una vez conectados a nuestro sistema. Esta información es especialmente útil para conocer los pasos y patrones que siguen los ciberdelincuentes.

A partir de esta información, se puede observar que ficheros consultan habitualmente, los ficheros que crean y descargan, y hasta conocer las direcciones IP donde tienen alojados el *malware.*

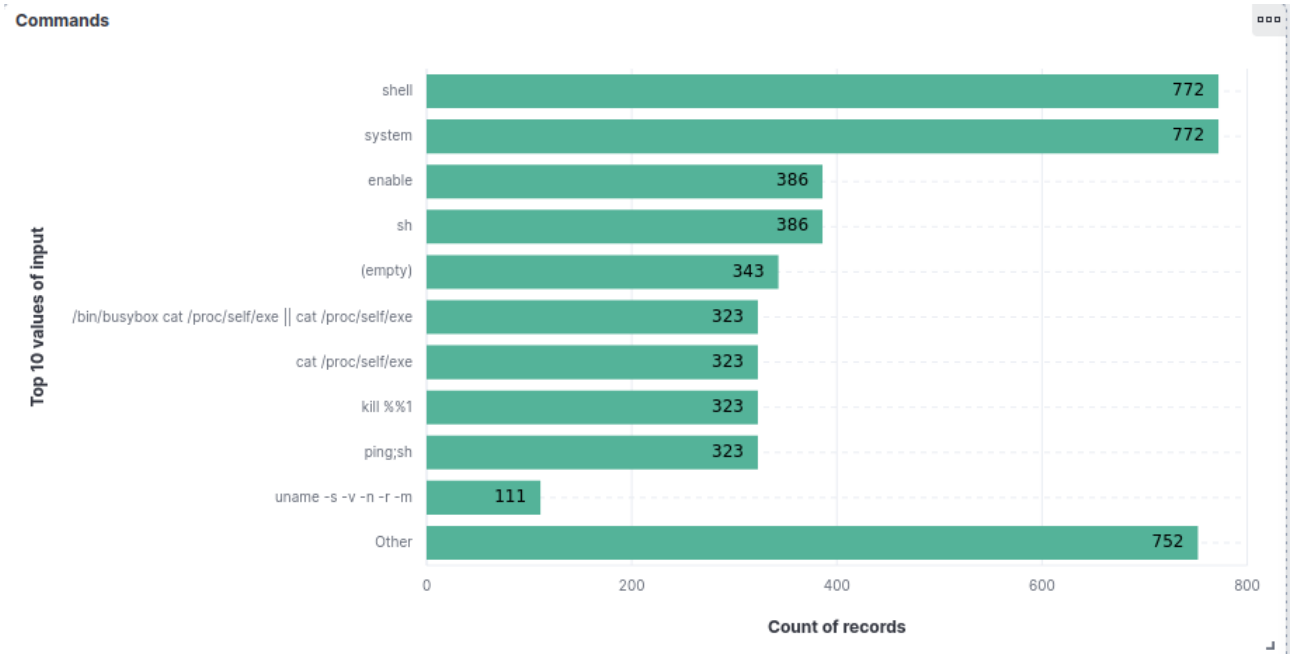


Figura 53. Top 10 comandos más utilizados

5.7 *Malware*

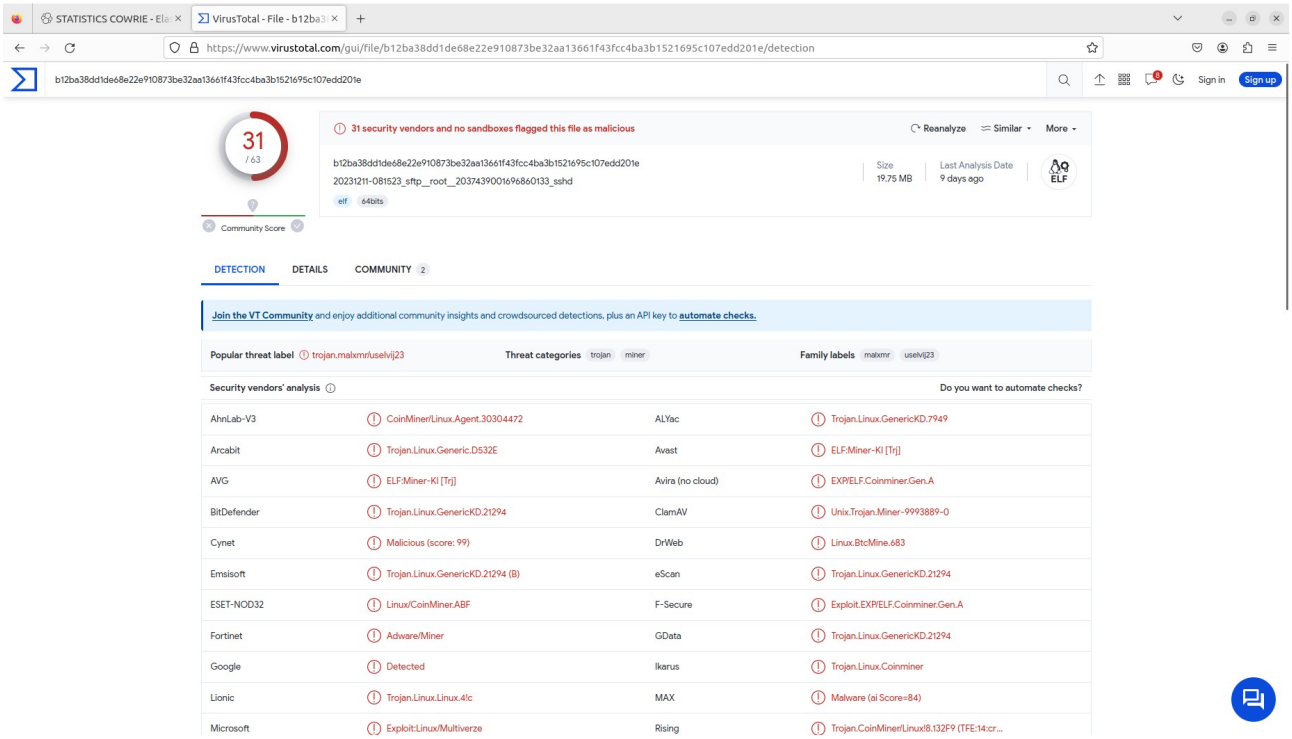
Durante el tiempo de exposición de *Cowrie*, se ha obtenido 1 muestras de *malware* que alguien ha descargado en el sistema de ficheros que implementa este *HoneyPot*. Estas muestras han quedado registradas como una copia del binario con su hash en SHA-256.

```
cowrie@fercho-Latitude-E5420:~/cowrie/var/lib/cowrie/downloads$ ls -l
total 20224
-rw------- 1 cowrie logstash 20709376 dic 23 20:50 b12ba38dd1de68e22e910873be32aa13661f43fcc4ba3b1521695c107edd201e
cowrie@fercho-Latitude-E5420:~/cowrie/var/lib/cowrie/downloads$
```

Figura 54. Fichero descargado en nuestro honeypot

Si las analizamos con la herramienta de **VirusTotal** nos indica que es un virus troyano minero. Este tipo de *malware* utiliza el poder computacional de los equipos infectados para minar criptomonedas de manera secreta e ilícita.

Figura 55. Análisis en VirusTotal



6. Capítulo 6. Conclusiones y propuestas

En este último capítulo se exponen las conclusiones obtenidas después de haber ejecutado el proyecto, junto con una serie de propuestas para mejorar la seguridad en los dispositivos IoT.

6.1 Conclusiones

La ejecución de este proyecto se ha realizado con la idea de ofrecer información objetiva y contrastada del peligro al que está expuesto constantemente una red IP en la que estén integrados dispositivos IoT.

Ha sido una sorpresa descubrir la velocidad con que un dispositivo comienza a recibir ataques a los poco minutos de exponerse en Internet. Por ello, ya se trate de una red doméstica o empresarial, se debe tomar consciencia de la cantidad de ataques automatizados que se reciben y que pueden ser evitados con una buena configuración de seguridad en dichos dispositivos IoT y otros elementos de la red.

Tal y como se puede observar en los resultados obtenidos durante el tiempo que se han estado tomando muestras, los HoneyPots son sistemas de gran ayuda para detectar patrones de ataque y obtener bastante información de los atacantes, como puede ser su IP, localización geográfica, ISP del que procede la IP, dirección de servidor del cual descargan malware y así tomar medidas acorde a los riesgos detectados.

Otra conclusión es que el procesamiento de la ingente cantidad de datos es una parte esencial de este tipo de proyectos. En poco tiempo hemos recibido tanta información que es importante contar con herramientas que faciliten la gestión, tratamiento y visualización de datos. Aun teniendo desplegado un solo *honeypot*, hubiera sido imposible llegar a las conclusiones extraídas si no hubiéramos implementado la pila Elastic (ELK). En los casos de varios despliegues o una *honeynet* es aun más imprescindible.

Generalmente, en la gran mayoría de los hogares y algunas empresas no se toman las medidas necesarias y oportunas de seguridad, ya sea por desconocimiento o por tener asociado algún tipo de coste. En realidad no debe enfocarse como un gasto, si no como una inversión que puede evitar intrusiones que conlleven consecuencias aun más costosas derivadas de una obtención de datos bancarios, claves de seguridad, robo de información confidencial, consumo energético excesivo, merma del rendimiento de los equipamientos, etc.

6.2. Medidas de Seguridad recomendadas

Tras observar los riesgos a los que están expuestos los dispositivos IoT que forman parte del ecosistema doméstico, se plantean las siguientes medidas de seguridad que hay que implementar para mitigar o eliminar las posibles amenazas existentes:

- Modificar las credenciales de acceso por defecto. Cambie siempre las contraseñas preinstaladas. Utilice contraseñas complicadas que incluyan letras, números y símbolos de mayúsculas y minúsculas si es posible.
- Evitar utilizar nombres de usuarios genéricos como admin, root, etc.
- Utilizar contraseñas robustas, que incluyan mayúsculas, minúsculas, números y símbolos, con una longitud de ocho caracteres como mínimo.
- Comunicaciones seguras que utilicen técnicas criptográficas que cifren la información para la interfaz de acceso al dispositivo. Cuando se acceda al dispositivo por medio de un navegador se debe comprobar que al comienzo de la dirección se utiliza el protocolo HTTPS. En caso de que el acceso al dispositivo no cuente con el protocolo de comunicación HTTPS se recomienda no administrarlo desde Internet.
- En el caso de comunicaciones seguras no cifradas y siempre que sean necesaria la comunicación con el dispositivo vía Internet, existe la opción de utilizar redes privadas virtuales. La implementación de una VPN ofrece comunicaciones seguras con el dispositivo IoT desde cualquier tipo de conexión, incluidas redes Wifi-públicas.

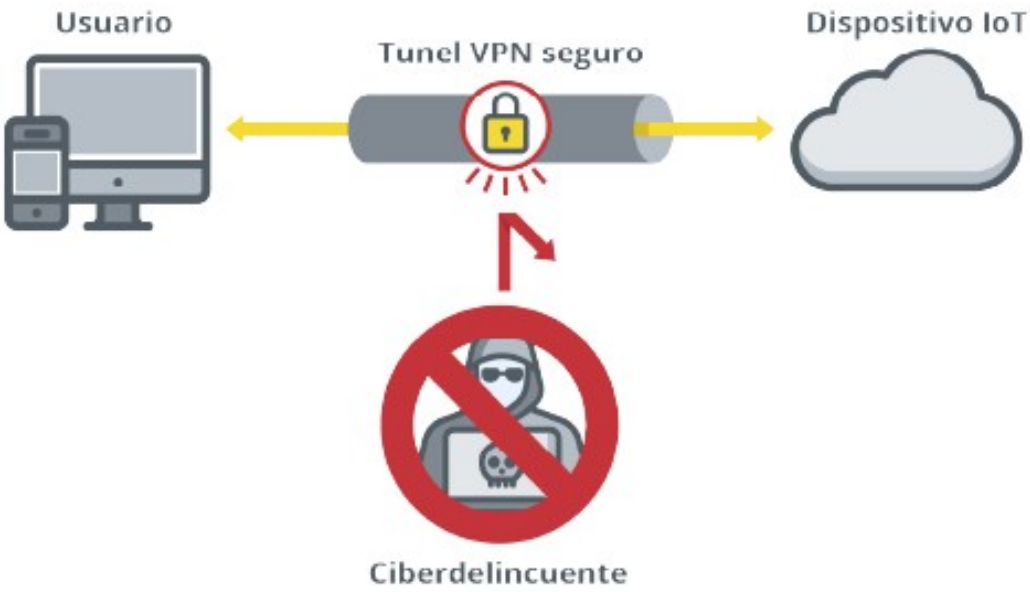


Figura 56. Utilización de una VPN

- Cuando el acceso se realiza por medio de una aplicación móvil se aconseja comprobar en las especificaciones de la propia aplicación si se utilizan mecanismos seguros de comunicación. En caso de no indicarlo es necesario contactar con el fabricante para que informe si los datos en tránsito están protegidos o no. Si la aplicación no cifra las comunicaciones, de la misma forma que sucede con el acceso por medio de una interfaz web, se debe utilizar una VPN.
- Aplicar las ultimas actualizaciones y parches de seguridad del dispositivo IoT, con esto se corregirán las ultimas vulnerabilidades descubiertas y se contara con las ultimas funcionalidades implementadas por el fabricante.
- Deshabilitar las características y funcionalidades que no vayan a ser utilizadas.
- Apagar la conectividad de red del dispositivo, en caso de que no vaya a ser utilizada. Por el contrario, si se va a utilizar o es necesaria para el funcionamiento del dispositivo, se debe comprobar que el panel de administración no es accesible desde internet.
- Aplicar medidas seguridad en otros dispositivos y capas de la red doméstica. Es recomendable configurar en el *firewall* del router local el filtrado de conexiones que se establecen con los dispositivos IoT para que solo sean permitidas aquellas conexiones desde determinados dispositivos y servicios. Es recomendable crear una o varias redes específicas para los dispositivos IoT y configurarlas como una DMZ.
- Si el dispositivo lo permite, se recomienda habilitar un registro todos los eventos que se produce en el dispositivo, como por ejemplo accesos, cambios de contraseña, actualizaciones, etc. Es recomendable que dichos logs sean compartidos con un servidor, *cloud* o similar para garantizar su disponibilidad en caso de averiarse o perder la conexión el dispositivo.
- Han de existir, en la medida de los posible, mecanismos de notificación cuando se produce un evento que pueda afectar a la seguridad del dispositivo.
- Aplicar las últimas actualizaciones y parches de seguridad será una prioridad, así se corregirán las últimas vulnerabilidades descubiertas y se contará con las últimas funcionalidades implementadas por el fabricante.
- En el caso de las organizaciones, la política de actualizaciones debe contemplar todas las casuísticas posibles como pueden ser los mantenimientos programados y los no programados
- Físicamente, se han de tomar algunas medidas como comprobar la cubierta protectora del aparato y cuán difícil es acceder a sus componentes internos. Se revisará el estado y acceso a puertos USB u otro tipo de puertos específicos de administración, vigilando su protección y su apagado si no fueran necesarios.

- Toda información que se almacene localmente en el dispositivo se hará cifrada para protegerla ante accesos no autorizados.

6.3. Trabajo futuro

Como futura línea de trabajo resultaría interesante trabajar con otros *HoneyPots* que emulen servicios diferentes a los que presentaba Cowrie y además alojarlos repartidos en servicios *cloud* como AWS por ejemplo. Creando distintos *pipelines* con logistash se podría que alimentar de datos muy valiosos nuestra implementación de elasticsearch.

Durante el proyecto se ha apreciado la falta de binarios descargados tras la fase de explotación inicial. Los *droppers*, aun siendo códigos simples, son capaces de detectar el entorno *HoneyPot*. Quizá con más modificaciones sobre el sistema Cowrie, como el sistema de archivos completo, se podría provocar más descargas de archivos en nuestro sistema trampa.

Glosario

Botnet: Se trata de una red de ordenadores controladas por un tercer actor y con la capacidad de obedecer las órdenes de este.

Cloud: Cloud computing es la disponibilidad bajo demanda de recursos de computación como servicios a través de Internet.

Cowrie: Honeypot destinado a simular interacciones de nivel medio a través de los protocolos de comunicación SSH y Telnet. Recibe los comandos enviados por el atacante, verifica que están disponibles, que son posibles y los simula. Además, en caso de llevar a cabo descarga de ficheros, estos son almacenados en una carpeta específica para posteriormente poderlos analizar.

Criptomoneda: Es un medio digital de intercambio que utiliza criptografía fuerte para asegurar las transacciones financieras, controlar la creación de unidades adicionales y verificar la transferencia de activos.

DoS (Denial of Service): Ataque informático a un sistema de computadores o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos.

DDoS: Se trata de un ataque de denegación de servicios realizado de manera distribuida.

DMZ: Zona desmilitarizada, es una red local que se ubica entre la red interna de una organización y la red externa (internet). La DMZ se utiliza habitualmente para ubicar servidores que deben ser accesibles desde el exterior, como por ejemplo servidores de Web, de correo o DNS.

Dropper: Es un tipo de troyano diseñado para "instalar" algún tipo de malware (virus, puerta trasera, etc.) en un sistema de destino.

Firewall: Configuración de seguridad en los router locales de usuario que restringe el tráfico de internet entrante, saliente o dentro de la misma red local. Funciona bloqueando o permitiendo tráfico de forma selectiva.

Fork: Se traduce al castellano como bifurcación. Cuando hacemos un fork de un repositorio, se hace una copia exacta en crudo (en inglés «*bare*») del repositorio original que podemos utilizar como un repositorio git cualquiera. Después de hacer fork tendremos dos repositorios git idénticos pero con distinta URL que pueden cada uno evolucionar de forma totalmente autónoma.

FTP: Protocolo de transferencia de ficheros, que se utiliza para transferir todo tipo de archivos entre equipos conectados a una red.

Honeynet: Tipo de Honeypot que consiste en una red diseñada para ser comprometida por ciberdelincuentes. Se utiliza para estudiar las técnicas utilizadas por los atacantes que han comprometido la seguridad de la red.

HoneyPot: Es un mecanismo de seguridad de computadora configurado para detectar, desviar o, de alguna manera, contrarrestar los intentos de uso no autorizado de sistemas de información.

HTTP: Protocolo de transferencia de hipertexto, protocolo de comunicación que permite la transferencia de información a través de archivos HTML en internet.

HTTPS: Protocolo de transferencia de hipertexto seguro, es la versión segura de HTTP, esta encriptado para aumentar la seguridad de las transferencias de datos.

IDS: Intrusion Detection System, aplicación que se utiliza para detectar accesos no autorizados a un sistema o red, es decir, sistemas que monitorizan tráfico entrante y lo cotejan con una base de datos actualizada de firmas de ataque conocidas.

IoT: Es un sistema de dispositivos informáticos interrelacionados, máquinas digitales y mecánicas, objetos, animales o personas que cuentan con identificadores únicos (UID) y la capacidad de transferir datos a través de una red sin necesidad de interacción de persona a persona o de persona a computadora.

IP: Internet Protocol, protocolo de comunicación de datos digitales que permite el desarrollo y transporte de paquetes de datos. Está clasificado funcionalmente en la capa de red del modelo internacional OSI.

ISP: Internet Service Provider, empresa que brinda conexión a internet a sus clientes.

IPTables: Modulo del núcleo de Linux que se encarga de filtrar los paquetes de red. Se encarga de determinar que paquetes de datos entran al sistema y cuales son filtrados.

Keylogger: Software que monitoriza y graba en un registro la actividad del teclado del usuario.

Log: Acontecimiento en un sistema informático que afecta a un proceso en particular y queda registrado en un archivo.

Login: Es el acto de introducir las credenciales (Usuario y contraseña) en un sistema informático.

Minero: Software malicioso encargado de resolver operaciones matemáticas relacionado con las criptomonedas para lucrar con nuestra capacidad de computo a un tercer actor.

VirusTotal: Es un sitio web creado por la empresa de seguridad española Hispasec Sistemas.

Malware: Son los tipos de software malicioso que intentan infectar un ordenador, dispositivo móvil u otros dispositivos conectados a internet, con múltiples finalidades, como por ejemplo extraer información personal o contraseñas, cifrar un disco duro o evitar que un usuario acceda a su dispositivo.

Man-in-the-middle: del inglés “Hombre en el medio”, es un tipo de ataque basado en interceptar la comunicación entre 2 o más interlocutores, pudiendo suplantar la identidad

de uno u otro según lo requiera para ver la información e incluso modificarla a su antojo, de tal forma que las respuestas recibidas en los extremos pueden estar dadas por el atacante y no por el interlocutor legítimo.

MySQL: Es un sistema de bases de datos de Oracle que se utiliza en todo el mundo para gestionar bases de datos.

Root: Cuenta de superusuario Linux. Cuenta que posee todos los privilegios y permisos necesarios para realizar cualquier operacion sobre el sistema operativo.

Router: Dispositivo que permite interconectar redes con distinto prefijo en su dirección IP. Su función es establecer la mejor ruta para llegar a otra red remota donde se encuentra el dispositivo destino.

SCP: Secure Copy Protocol, protocolo que se utiliza para la transferencia de archivos de forma segura. Este protocolo permite la transferencia de datos entre equipos a través del protocolo SSH.

SHA-256: Función de 256 del algoritmo SHA-2. Este tipo de algoritmos son utilizados para verificar la identidad de las copias de datos sin riesgo de colisiones.

SMTP: Simple Mail Transfer Protocol, protocolo o conjunto de reglas de comunicación que utilizan los servidores de correo electrónico para enviar y recibir e-mails.

SSH: Protocolo de red que permite el acceso remoto a sistemas informáticos por medio de un canal seguro, a través de una conexión cifrada. Su puerto de servicio es el tcp 22.

SPAM: Correo electrónico masivo no deseado. Es decir, un email que se envía a multitud de personas sin aprobación previa, con el objetivo de promover un producto, servicio o hasta una estafa en particular.

TCP: Transmision Control Protocol, protocolo ubicado en la capa de transporte del modelo OSI, cuyo objetivo es crear conexiones dentro de una red para el intercambio de datos.

Telnet: Protocolo de red que permite el acceso remoto a sistemas informáticos por medio de un canal en el que la información se envía en claro. Su puerto de servicio es el tcp 23.

UDP: User Datagram Protocol, protocolo de la capa de transporte del modelo OSI, basado en la transmisión sin conexión de datagramas.

UML (Unified Modeling Language): Es un lenguaje de modelado visual de software, indispensable para la arquitectura y la ingeniería de software y sistemas.

Wearables: Es un dispositivo electrónico que se usa en el cuerpo humano y que interactúa con otros aparatos para transmitir o recoger algún tipo de datos.

Bibliografía

[1] *¿Qué es el Internet de las cosas?* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.redhat.com/es/topics/internet-of-things/what-is-iot

[2] *Internet de las cosas* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://es.wikipedia.org/wiki/Internet_de_las_cosas

[3] *¿Qué es IoT (Internet de las cosas)?* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://aws.amazon.com/es/what-is/iot/

[4] *internet of things (IoT)* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[5] *Megalista: Ejemplos de dispositivos IoT (2023)* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://opensistemas.com/dispositivos-iot/

[6] *Seguridad en la instalación y uso de dispositivos IoT* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia-de-seguridad-iot.pdf

[7] *El número de ataques a dispositivos IoT se duplica en un año* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.kaspersky.es/about/press-releases/2021_el-numero-de-ataques-a-dispositivos-iot-se-duplica-en-un-ano

[8] *Botnet* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.incibe.es/aprendeciberseguridad/botnet

[9] *¿Qué es una botnet?* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.proofpoint.com/es/threat-reference/botnet

[10] *Mirai (Malware).* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en:

https://es.wikipedia.org/wiki/Mirai_(malware)

[11] *Hajime hace estragos en IoT* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.redestelecom.es/seguridad/noticias/1097666002503/hajime-estragos-iot.1.html

[12] *Reaper, otra botnet que surge gracias a dispositivos IoT que no son seguros* [en línea] [fecha de consulta: febrero 2023]. Disponible en: https://www.redeszone.net/2017/10/22/reaper-otra-botnet-surge-gracias-dispositivos-iot-no-seguros/

[13] *El ataque de los botnets* [en línea] [fecha de consulta: febrero 2023]. Disponible en: https://www.danysoft.com/los-12-peores-botnets/

[14] *¿Qué es y para qué sirve un Honeypot?* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en:

https://www.redeszone.net/tutoriales/seguridad/que-es-honeypot/

[15] *Honeypot* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.sofistic.com/productos/honeypot/

[16] *Honeypot* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://es.wikipedia.org/wiki/Honeypot#Definici%C3%B3n_del_t%C3%A9rmino

[17] http://repositorio.espe.edu.ec/bitstream/21000/421/1/T-ESPE-021875.pdf

[18] *Fernández, AFMdA. (2019) - "Seguridad en Internet de las Cosas: Diseño y desarrollo de un honeypot para el análisis de ataques a IoTs"*. Universitat Oberta de Catalunya. [en línea][fecha de consulta: Noviembre 2023]. Disponible en: https://openaccess.uoc.edu/bitstream/10609/96546/6/albenizTFM0619memoria.pdf

[19] *HoneyPots Parte1-Kippo* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://thehackerway.com/2015/03/24/honeypots-parte-1-kippo/

[20] *IoTPot: A novel honeypot for revealing current IoT threats* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.researchgate.net/publication/303181974_IoTPOT_A_novel_honeypot_for_revealing_current_IoT_threats

[21] *Telnet IoT honeypot* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://github.com/Phype/telnet-iot-honeypot

[22] *Conoce el Honeypot Cowrie, compatible con servicios SSH y Telnet* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.hacking.land/2017/05/conoce-el-honeypot-cowrie-compatible.html?m=1

[23] *T-Pot: Una colmena de Honeypots para atraparlos a todos* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.hacking.land/2017/07/t-pot-una-colmena-de-honeypots-para.html?m=1#google_vignette

[24] *Instalar Honeypot T-Pot en una máquina virtual* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en:https://blog.elhacker.net/2021/01/instalar-honeypot-t-pot-en-una-maquina-virtual-tpotce-cowrie-docker-dionea.html

[25] *¿Qué es Ubuntu? Una guía rápida para principiantes* [en línea] [fecha de consulta: Noviembre 2023].
Disponible en: https://www.hostinger.es/tutoriales/que-es-ubuntu#%C2%BFQue_es_Ubuntu

[26] *¿Qué es la pila ELK?* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://aws.amazon.com/es/what-is/elk-stack/

[27] https://www.elastic.co/es/elastic-stack

[28] *¿Qué es el sandboxing? ¿Cómo funciona el software de sandbox en la nube?* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.avast.com/es-es/business/resources/what-is-sandboxing#pc

[29] *5 plataformas de sandbox online* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://keepcoding.io/blog/plataformas-de-sandbox-online/

[30] *Shodan: qué es y para qué se puede usar este buscador de dispositivos conectados a Internet* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://keepcoding.io/blog/plataformas-de-sandbox-online/

[31] *5 buscadores de dispositivos y servicios conectados a Internet* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.welivesecurity.com/la-es/2023/03/01/buscadores-dispositivos-servicios-conectados-internet/

[32] *Geolocalizando IPs con Wireshark y GeoIP (Parte I).* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://red.computerworld.es/actualidad/geolocalizando-ips-con-wireshark-y-geoip-parte-i

Bibliografía no referenciada en la memoria, pero sí consultada para comprensión general del TFG e implementación

[33] *Telnet IoT honeypot* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://github.com/Phype/telnet-iot-honeypot

[34] *Installing Cowrie in seven steps* [en línea] [fecha de consulta: Diciembre 2023]. Disponible en: https://cowrie.readthedocs.io/en/latest/INSTALL.html#

[35] *López, AL. [AlbertoLopez TECH TIPS]. (2021). [DMZ] ¿Qué es una DMZ? ► Zona DESMILITARIZADA en redes informáticas | Alberto López.* [YouTube].[fecha de consulta: Diciembre 2023]. Disponible en: https://www.youtube.com/watch?v=YR8xaXvGcWc

[36] *Fernández, LF (2020) - La encriptación End-to-End es el próximo objetivo de los dispositivos IoT* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://www.redeszone.net/noticias/seguridad/encriptacion-end-to-end-dispositivos-iot/

[37] *Dirección ip estática.* Disponible en: https://servidordebian.org/es/jessie/config/network/static_ip

[38] *Pérez, IPM. Hacking en IoT, acceso al firmware de un dispositivo* [en línea]. Disponible en: https://www.futurespace.es/hacking-en-iot/

[39] *Seguridad en dispositivos IoT.* [en línea] [fecha de consulta: Noviembre 2023]. Disponible en: https://gurudelainformatica.es/seguridad-en-dispositivos-iot

[40] *Verdejo Alvarez, GVA. – "CAPÍTULO 4: SEGURIDAD EN REDES IP: Honeypots y Honeynets".* [en línea][fecha de consulta: Noviembre 2023]. Disponible en: https://www.cs.upc.edu/~gabriel/files/DEA-es-4HoneypotsyHoneynets.pdf

[41] *Installing the Elastic Stack* [en línea][fecha de consulta: Diciembre 2023]. Disponible en: https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html

[42] *¿QUÉ ES ELK? ElasticSearch, Logstash y Kibana* [en línea][fecha de consulta: Diciembre 2023]. Disponible en: https://openwebinars.net/blog/que-es-elk-elasticsearch-logstash-y-kibana/

[43] *How to send Cowrie output to an ELK stack* [en línea][fecha de consulta: Diciembre 2023]. Disponible en: https://cowrie.readthedocs.io/en/latest/elk/README.html

[44] *Cazando Malware (Parte II)* [en línea][fecha de consulta: Diciembre 2023]. Disponible en: https://www.hackplayers.com/2017/11/cazando-malware-parte-ii.html

[45] *Introducción práctica a Logstash* [en línea][fecha de consulta: Diciembre 2023]. Disponible en: https://www.elastic.co/es/blog/a-practical-introduction-to-logstash

Anexo

Instalación de Cowrie en 7 pasos

Paso 1: Instalar dependencias del sistema

Primero instalamos el soporte de todo el sistema para entornos virtuales Python y otras dependencias. Los paquetes reales de Python se instalan más tarde.

En sistemas basados en Debian (última verificación en Debian 10, 2021-04-29):

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

Paso 2: Crear un usuario

Se recomienda encarecidamente utilizar un identificador de usuario no root:

```
  $ sudo adduser --disabled-password cowrie
  Adding user 'cowrie' ...
  Adding new group 'cowrie' (1002) ...
  Adding new user 'cowrie' (1002) with group 'cowrie' ...
  Changing the user information for cowrie
  Enter the new value, or press ENTER for the default
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
  Is the information correct? [Y/n]

  $ sudo su - cowrie
```

Paso 3: Comprobar el código

Comprueba el código:

```
  $ git clone http://github.com/cowrie/cowrie
  Cloning into 'cowrie'...
  remote: Counting objects: 2965, done.
  remote: Compressing objects: 100% (1025/1025), done.
  remote: Total 2965 (delta 1908), reused 2962 (delta 1905), pack-reused 0
  Receiving objects: 100% (2965/2965), 3.41 MiB | 2.57 MiB/s, done.
  Resolving deltas: 100% (1908/1908), done.
  Checking connectivity... done.

  $ cd cowrie
```

Paso 4: Configurar Virtual Environment

A continuación, debe crear su entorno virtual:

```
  $ pwd
  /home/cowrie/cowrie
```

```
$ python -m venv cowrie-env
New python executable in ./cowrie/cowrie-env/bin/python
Installing setuptools, pip, wheel...done.
```

Active el entorno virtual e instale los paquetes:

```
$ source cowrie-env/bin/activate
(cowrie-env) $ python -m pip install --upgrade pip
(cowrie-env) $ python -m pip install --upgrade -r requirements.txt
```

Paso 5: Instalar la configuración

La configuración de Cowrie se almacena en cowrie.cfg.dist y cowrie.cfg (ubicados en cowrie/etc). Ambos archivos se leen en el arranque, donde las entradas de cowrie.cfg tienen prioridad. El archivo .dist puede ser sobrescrito por actualizaciones, cowrie.cfg no será tocado. Para ejecutar con una configuración estándar, no hay necesidad de cambiar nada. Para habilitar telnet, por ejemplo, cree cowrie.cfg e introduzca sólo lo siguiente:

```
[telnet]
enabled = true
```

Paso 6: Iniciar Cowrie

Inicie Cowrie con el comando cowrie. Puede añadir el directorio cowrie/bin a su ruta si lo desea. Un entorno virtual existente se conserva si está activado, de lo contrario Cowrie intentará cargar el entorno llamado "cowrie-env":

```
$ bin/cowrie start
Activating virtualenv "cowrie-env"
Starting cowrie with extra arguments [] ...
```

Paso 7: Escuchar en el puerto 22 (OPCIONAL)

Hay tres métodos para hacer Cowrie accesible en el puerto SSH por defecto (22): iptables, authbind y setcap.

Iptables

Los comandos de redirección de puertos son a nivel de sistema y necesitan ser ejecutados como root. Una redirección de firewall puede hacer que su servidor SSH existente sea inalcanzable, recuerde mover el servidor existente a un número de puerto diferente primero.

La siguiente regla de firewall redireccionará el tráfico entrante en el puerto 22 al puerto 2222 en Linux:

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

O para telnet:

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

Tenga en cuenta que debe probar esta regla sólo desde otro host; no se aplica a conexiones loopback.

Instalar pila Elastic (ELK)

Instalar ElasticSearch

El paquete Debian para Elasticsearch v8.11.3 puede descargarse desde el sitio web e instalarse de la siguiente manera:

```
  wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.11.3-
amd64.deb
  wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-8.11.3-
amd64.deb.sha512
  shasum -a 512 -c elasticsearch-8.11.3-amd64.deb.sha512
  sudo dpkg -i elasticsearch-8.11.3-amd64.deb
```

Instalar Kibana

El paquete Debian para Kibana v8.11.3 puede descargarse desde el sitio web e instalarse de la siguiente manera:

```
  wget https://artifacts.elastic.co/downloads/kibana/kibana-8.11.3-amd64.deb
  shasum -a 512 kibana-8.11.3-amd64.deb
  sudo dpkg -i kibana-8.11.3-amd64.deb
```

Instalar Logstash

Descargue e instale la clave pública de firma:

```
  wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/usr/share/keyrings/elastic-keyring.gpg
```

Puede que necesite instalar el paquete apt-transport-https en Debian antes de continuar:

```
  sudo apt-get install apt-transport-https
```

Guarda la definición del repositorio en /etc/apt/sources.list.d/elastic-8.x.list:

```
  echo "deb [signed-by=/usr/share/keyrings/elastic-keyring.gpg]
https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee -a
/etc/apt/sources.list.d/elastic-8.x.list
```

Ejecute sudo apt-get update y el repositorio estará listo para su uso. Puede instalarlo con:

```
  sudo apt-get update && sudo apt-get install logstash
```