Universidad de Buenos Aires
Facultad de Ciencias Económicas
Ciencias Exactas y Naturales e Ingeniería

CARRERA DE ESPECIALIZACIÓN EN
SEGURIDAD INFORMÁTICA

TRABAJO FINAL DE ESPECIALIZACIÓN

TEMA

SEGURIDAD A TRAVÉS DE HONEYPOTS

TÍTULO

ANÁLISIS DE T-POT, UN HONEYPOT OPEN SOURCE

AUTOR: FABRICIO GABRIEL TORRICO BARAHONA

TUTOR DE TRABAJO FINAL: DR. PEDRO HECHT

2022

COHORTE 2020

DECLARACIÓN JURADA DE ORIGEN DE LOS CONTENIDOS

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.

FIRMADO

Fabricio Gabriel Torrico Barahona
CI: 6180245 LP, BOLIVIA

RESUMEN

El presente trabajo final de especialización tiene por objetivo analizar y describir en detalle la arquitectura del *honeypot* T-Pot. Para ello se inicia con una introducción teórica asociada a sistemas señuelos o *honeypots*, permitiendo al lector tener un entendimiento claro y preciso del tema. Posteriormente se analiza la empresa que lidera el desarrollo de T-POT, entendiendo de ese modo la dimensión real del proyecto que se analizará. Tras ello y tomando en cuenta que T-POT basa su arquitectura en el uso de contenedores Docker, se inicia con el análisis pormenorizado de cada contenedor, que puede ser un honeypot, un NSM (Network Security Monitor) o una herramienta de apoyo; posteriormente se analiza el funcionamiento sinérgico de cada uno de los componentes, lo cual permite entender la arquitectura y excelente uso de tecnologías que permite hacer de T-POT uno de los *honeypots* más usados y difundidos a nivel mundial. Finalmente y tras mostrar las ventajas de T-POT, se plantean una serie de mejoras que podrán ser estudiadas o desarrolladas en futuros trabajos de investigación.

Palabras clave: seguridad, riesgo, amenazas, vulnerabilidades, ciberataques, red de datos, infraestructura tecnológica, servicios, virtualización, contenedores.

ÍNDICE

ÍNDICE DE FIGURAS

AGRADECIMIENTOS

El presente trabajo final de especialización está dedicado a Dios, a mis amados papás y hermana Guisela, Dante y Gabriela. Mi más grato agradecimiento a ellos que son el pilar fundamental y quienes me dan la fortaleza y alegría día a día.

Agradecido también con mis abuelitos Walter, Alcira, Juan y Lidia, a todos mis familiares, a Mischell, amigos y todas aquellas personas que están a mi lado apoyándome día a día.

CUERPO INTRODUCTORIO

I. INTRODUCCIÓN

Si utiliza algún tipo de tecnología en sus labores diarias, seguro escuchó hablar de antivirus; si tiene conocimientos en el área de informática, probablemente entienda la funcionalidad de cortafuegos (*firewall*), certificados digitales, redes privadas virtuales (*VPN),* sistemas de detección o prevención de intrusos *(IDS o IPS)*, Gestión de Eventos e Información de Seguridad (*SIEM)* entre otros; pero inclusive si está inmerso en temas de seguridad, es probable que nunca haya escuchado el término *honeypot* o en su defecto no sepa a cabalidad su uso y finalidad.

Es interesante revisar estadísticas [1] y evidenciar que a pesar de la gran cantidad de herramientas de seguridad disponibles en el mercado, más del 90% de las organizaciones de salud han reportado al menos una brecha de ciberseguridad en los últimos años; el 62.7% de las empresas creen que los ciberataques han aumentado desde el año 2020 debido a la pandemia de COVID-19; los ataques de malware en el año 2020 aumentaron en un 358% respecto a años anteriores y en un 435% respecto al 2019, además que el 93% del malware observado el año 2019 es polimórfico (capaz de modificar su programación para evitar ser detectado). Pues a pesar de que las empresas gastan un estimado de 2.4 millones de dólares en defensa y que un 91% han aumentado sus costos en seguridad informática durante el 2021, los ataques siguen afectando a todo tipo de empresas, con estadísticas que indican que un 43% de los ciberataques afectan a pequeños negocios; pero eso no es todo, a una compañía le toma aproximadamente 6 meses detectar una brecha de seguridad. Y es que en general las medidas implementadas son de carácter defensivo o correctivo, cuando lo ideal sería implementar seguridad proactiva que permita conocer al enemigo, saber cómo actúa, qué herramientas emplea, con quienes se comunica, su nivel de conocimiento y pericia en diversos tipos de ataques, etc. Muchas de estas inquietudes pueden ser resueltas con una correcta implementación de sistemas señuelos o *honeypots*.

El presente trabajo de especialización desarrolla de manera clara y ordenada los conceptos teóricos relacionados con *honeypots*, su finalidad, clasificación, ventajas y desventajas de su implementación, además de analizar en profundidad T-Pot, un "*honeypot* que ejecuta diversos *honeypots*" en un único equipo de cómputo.

II. OBJETIVOS

El presente trabajo contempla los siguientes objetivos.

II a. OBJETIVO GENERAL

Analizar y describir en detalle la arquitectura del honeypot T-Pot, permitiendo un entendimiento de cada uno de sus componentes y la forma en que trabajan de manera conjunta.

II b. OBJETIVOS ESPECÍFICOS

Los objetivos específicos son:

- Mostrar la utilidad de los honeypot, su clasificación, implementación, ventajas y desventajas.
- Analizar en profundidad la arquitectura y herramientas que componen el honeypot T-Pot.
- Plantear posibles mejoras o desarrollo de nuevos módulos para el honeypot T-Pot.

III. ALCANCES

Los alcances del proyecto son los siguientes:

- Se realizará un desarrollo teórico de los *honeypots*.
- Se recopilará información pública de la empresa Deutsche Telekom.
- Se analizará la arquitectura y herramientas de T-Pot a través de la documentación del proyecto e implementación en un entorno controlado.
- Se plantearán mejoras al T-Pot de manera descriptiva, sin desarrollar ni implementar las propuestas.

IV. METODOLOGÍA

El trabajo plantea una metodología exploratoria y descriptiva, tomando un enfoque cualitativo, donde el método de recolección y análisis de datos sigue un enfoque cuasi experimental.

CUERPO PRINCIPAL

1. HONEYPOTS

Como se mencionó en la introducción, el personal dedicado a la seguridad sabe a cabalidad la definición y finalidad de herramientas como antivirus, *firewalls*, UTMs, *IDS, IPS, SIEM,* entre otros; lo cual no ocurre con los *honeypots,* esto probablemente se deba a que no existe una definición clara ni ampliamente aceptada por la comunidad de seguridad; como se explica en [2], “Algunos piensan que un *honeypot* es una herramienta para el engaño, otros lo consideran un arma para atraer a los piratas informáticos y otros creen que es simplemente otra herramienta de detección de intrusos. Unos creen que un *honeypot* debería emular vulnerabilidades, otros lo ven simplemente como una cárcel y algunos ven los *honeypots* como sistemas de producción controlados en los que los atacantes pueden entrar”.

La finalidad de los siguientes puntos es establecer una definición consensuada de *honeypot*, explicar su funcionalidad, clasificación y mencionar algunas ventajas y desventajas; de modo tal que el lector llegue con una base sólida y claridad en los conceptos que serán empleados en el análisis de T-Pot.

1.1. DEFINICIÓN

Existen diversas definiciones de *honeypot,* entre las más relevantes:

- “Un *honeypot* es un recurso de seguridad cuyo valor radica en ser investigado, atacado o comprometido” [2]
- “Un *honeypot* es un recurso informático monitoreado de cerca que queremos que sea investigado, atacado o comprometido” [3]
- “El *honeypot* consiste en crear una herramienta de seguridad informática dentro de una red con el único objetivo de figurar como un señuelo, de forma que los ataques externos se centren en él y no en los archivos o los puntos débiles reales de tu negocio” [4]
- “Los *honeypot* son sistemas hardware o herramientas software que simulan ser equipos vulnerables para poder exponerlos sin ningún

riesgo y permitir el análisis de todos los ataques efectuados sobre ellos" [5]

- "Un *honeypot*, o sistema trampa o señuelo, es una herramienta de la seguridad informática dispuesto en una red o sistema informático para ser el objetivo de un posible ataque informático, y así poder detectarlo y obtener información del mismo y del atacante" [6]
- "Un *honeypot* es un sistema que podemos configurar en nuestra red con el objetivo de registrar los ataques que éste recibe y que nos puede servir como una alerta temprana para prevenir otros ataques en la red, para saber si somos blanco de algún tipo de ataque o para entender mejor el panorama al que nos enfrentamos. Además, nos permite obtener la procedencia de las amenazas que existen y las técnicas a las cuales nos podemos enfrentar" [7]
- "Es un sistema informático que se "sacrifica" para atraer ciberataques, como un señuelo. Simula ser un objetivo para los hackers y utiliza sus intentos de intrusión para obtener información sobre los cibercriminales y la forma en que operan o para distraerlos de otros objetivos" [8]

Tras analizar las diferentes definiciones y funcionalidades de los *honeypots,* se plantea la siguiente definición que resume de manera clara y concisa la finalidad de los sistemas estudiados.

Un *honeypot,* también llamado sistema trampa o señuelo, es aquel recurso de seguridad que es empleado para investigación o producción, que espera tener algún grado de interacción con los atacantes y cuyo valor radica en ser atacado, comprometido e investigado.

1.2. FUNCIONALIDAD

De [2] y [3] se concluye que los *honeypot* no tienen valor de producción, ninguna persona o recurso debe interactuar con ellos; por lo que cualquier intento de contactarlo es sospechoso.

Dada la premisa, un *honeypot* debe publicar aquellos servicios que se desean monitorear, los mismos deben ser accesibles desde la red (ya sea interna o Internet según su objetivo). La funcionalidad radica en que nadie hace uso de ese servicio y debe estar constantemente monitoreado, por lo

que cualquier actividad que se detecte es sin duda alguna una amenaza (sondas, gusanos, *botnets*, ataques masivos, *script kiddies*, ataques dirigidos, etc.)

Una vez detectada algún tipo de actividad, el objetivo es analizar los hallazgos para poder detectar aspectos importantes como ser: de dónde vienen los cibercriminales, el nivel de amenaza, el modus operandi empleado, los datos o aplicaciones que buscan, la eficacia de las medidas de seguridad locales para detener los ciberataques, entre otros. [8]

1.3. CLASIFICACIÓN

Se destacan dos clasificaciones principales.

- Según su finalidad
- Según su nivel de interacción

1.3.1. SEGÚN SU FINALIDAD

Dada la finalidad del *honeypot*, se clasifican en *honeypots* de producción y de investigación [9].

Honeypot de producción. Sistemas empleados para investigar el motivo por el cual los ciberdelincuentes ejecutan cierto tipo de ataques. La finalidad es averiguar por qué se han fijado en esa empresa, e intentar desviar o mitigar el riesgo de dichos ataques en su red interna.

Honeypot de investigación. Sistemas diseñados para obtener información sobre la comunidad *blackhat*, generalmente utilizados por organizaciones sin ánimos de lucro e instituciones educativas. Su misión principal es investigar las amenazas que pueden enfrentar las organizaciones, como quiénes son los atacantes, cómo están organizados, qué tipo de herramientas usan para atacar otros sistemas y dónde obtuvieron esas herramientas.

1.3.2. SEGÚN SU NIVEL DE INTERACCIÓN

El nivel de interacción da una escala con la que medir y comparar *honeypots*. Cuanto más pueda hacer un *honeypot* y cuanto más pueda hacer un atacante a un *honeypot*, mayor será la información que se pueda derivar de él. Sin embargo, de la misma manera, cuanto más pueda hacer un atacante

al *honeypot*, más daño potencial puede hacer un atacante [2]. Bajo esta característica, algunos autores (como [3], [4] y [9]) lo dividen en *honeypots* de interacción alta y baja; mientras que otros (como [2] y [7]) agregan una tercera clasificación, denominándola de interacción media. En general, cada uno tiene particularidades que lo caracterizan.

Honeypot de alta interacción. Son equipos que corren servicios, aplicaciones y sistemas operativos reales. El nivel de riesgo que trae consigo su implementación es alto ya que si el atacante logra tomar control, podría usarlo para acceder al resto de la red interna. Su implementación y mantenimiento requieren bastante dedicación. Las ventajas de este tipo de *honeypots* es que es difícil que un atacante descubra que está interactuando con un sistema trampa y la información que recopila es muy amplia y detallada.

Honeypot de baja interacción. Solo emulan vulnerabilidades o servicios de red básicos (como la conectividad TCP/IP, ICMP, NetBIOS, etc.). El nivel de riesgo que trae consigo su implementación es bajo. Son relativamente fáciles de implementar y mantener. La desventaja es que un atacante experimentado podría descubrir que se encuentra frente a un sistema trampa, además que la información recolectada es escasa; se puede detectar un escaneo de red, pero no se podrá obtener más información sobre las técnicas o las intenciones del atacante.

Honeypot de interacción media. Tiene algunos servicios básicos emulados, pero a diferencia de los *honeypot* de baja interacción, pueden dar algún tipo de respuesta al atacante; por ejemplo, devolver un banner del servicio. La información que se genera en este tipo de *honeypot* es considerablemente menor que en uno de alta interacción, pero son más fáciles de mantener e implementar.

1.4. VENTAJAS Y DESVENTAJAS

Los *honeypots* son sin duda alguna una excelente herramienta de seguridad informática, pero como cualquier solución tiene beneficios y perjuicios. Autores como [2], [5] y [8] citas las siguientes ventajas:

Valor de los datos. A diferencia de dispositivos como *firewalls, IDS o IPS,* los *honeypot* recopilan pocos datos, por lo que son de muy alto valor. El análisis de los datos recopilados es muy fácil y permite una reacción más rápida. Además, proporcionan información sobre el atacante, su metodología, sus herramientas e inclusive su propósito y grado de conocimiento.

Pocos recursos de hardware. Dispositivos de seguridad como *firewalls, IDS, IPS* o *SIEM* pueden llegar a requerir gran capacidad de cómputo, tarjetas de red que soporte altas velocidades, mucho espacio de almacenamiento; en general, recursos de hardware de última tecnología. Eso no ocurre con los *honeypots*, debido a que capturan y monitorean poca actividad, generalmente no tienen problemas de agotamiento de recursos, funcionarán a la perfección con pocos recursos virtualizados o inclusive reutilizando equipos que queden obsoletos para su uso.

Simplicidad. Probablemente la simplicidad es la mayor ventaja de los *honeypots*. No es necesario desarrollar algoritmos sofisticados, ni bases de datos de firmas que mantener, ni bases de reglas que configurar incorrectamente. Simplemente se lo debe implementar, sentarse y esperar. Si bien algunos *honeypots*; especialmente los de investigación, pueden ser más complejos, todos operan con la misma premisa simple: comprobar si alguien se conecta al sistema señuelo. Como los profesionales de seguridad experimentados dicen, cuanto más simple es el concepto, más confiable es. Con la complejidad vienen las configuraciones erróneas, las averías y los fallos.

Retorno de inversión. Soluciones de seguridad como antivirus, antispam, *firewalls, IPS, VPNs,* etc. pueden llegar a ser muy difíciles de justificar ante la gerencia y muchas veces se convierten en víctimas de su propio éxito debido a que al proteger la infraestructura de ataques, estos parecen no existir. Un *honeypot* permite justificar no solo su valor, sino también el del resto de los otros dispositivos de seguridad, ya que evidencia que las amenazas están presentes y los ataques son constantes.

Ayuda al personal de TI y seguridad. Puede distraer a los atacantes y dar tiempo al personal de TI y seguridad a proteger lo que realmente importa.

Puede ser empleado como ambiente de pruebas de seguridad ante amenazas de atacantes reales. Los ataques y resultados pueden ser compartidos con miembros de TI y seguridad de otras empresas.

Pero es necesario mencionar que no es la solución a todos los problemas, autores como [2], [4], [5] y [8] mencionan las siguientes desventajas que trae consigo la implementación de *honeypots:*

Campo de visión estrecho. La mayor desventaja de los *honeypots* es que tienen un campo de visión estrecho, solo ven aquella actividad que está dirigida contra ellos. Si un atacante irrumpe en una red y ataca una variedad de sistemas, un *honeypot* será inconsciente de la actividad a menos que sea atacado directamente.

Huellas digitales. Otra desventaja de los *honeypots*, especialmente muchas versiones comerciales, es la toma de huellas digitales. La toma de huellas dactilares se produce cuando un atacante puede identificar la verdadera identidad de un *honeypot* porque tiene ciertas características o comportamientos esperados. Por ejemplo, un *honeypot* puede estar diseñado para emular un servidor web NT IIS pero tiene ciertas características que lo identifican como un servidor Unix Solaris, estas identidades contradictorias pueden actuar como una firma para un *honeypot*.

Riesgo. Un *honeypot* puede introducir riesgos al entorno en el cual es implementado, esto debido a que tras atacar el *honeypot,* el atacante podría utilizarlo para atacar, infiltrarse o dañar otros sistemas u organizaciones.

Personal y tiempo para análisis y mantenimiento. Un *honeypot* no es una solución que se configure una vez y se la deje en producción, requiere mantenimiento y supervisión para poder sumar valor a la entidad, así como personal y tiempo dedicado al análisis de ataques recibidos.

2. DEUTSCHE TELEKOM AG

Para poder dimensionar y entender la verdadera magnitud del *honeypot* a analizar, se desarrolla a continuación algunos aspectos relevantes de Deutsche Telekom, empresa que lidera el desarrollo del proyecto T-Pot.

Deutsche Telekom fue fundada en 1995, es uno de los principales proveedores mundiales de tecnología de la información y telecomunicaciones, con sede en Alemania y presencia en más de 50 países. Cuenta con alrededor de 211.000 empleados en todo el mundo y genera ingresos anuales de 80,5 billones de euros, distribuidos en una estructura de accionistas compuesta por un 68,1% de capital flotante, 14,5% propiedad de la República Federal de Alemania y 17,4% propiedad del grupo bancario KfW. En la actualidad no se dedica únicamente a las telecomunicaciones, pasó de ser una compañía telefónica tradicional a un tipo de compañía de servicios, la cual tiene divisiones especializadas en temas como seguridad informática, nube, internet de las cosas, servicio al cliente, IT, consultorías, etc. [10]

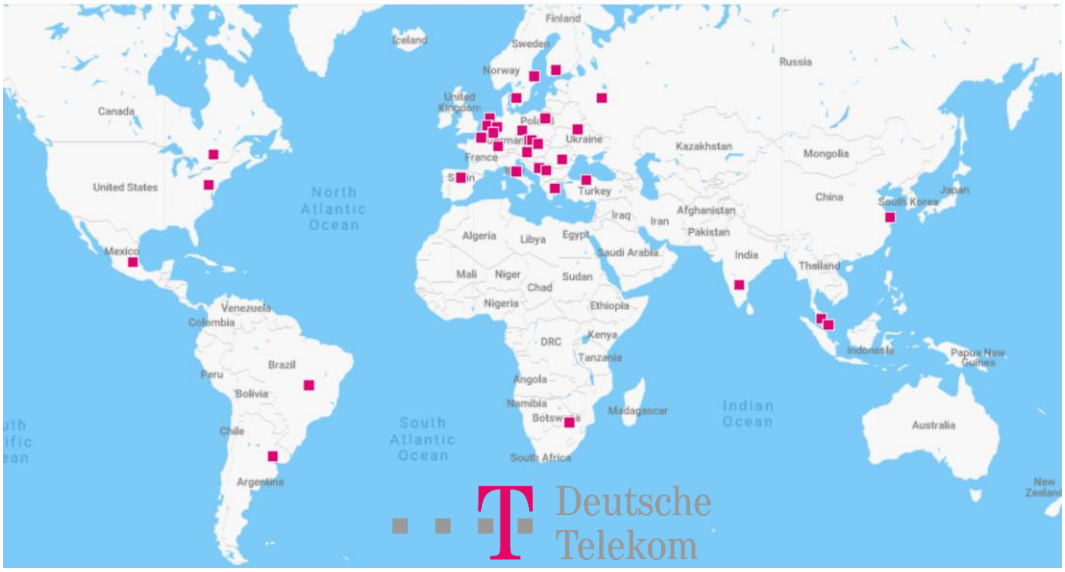


Figura 1: Países donde opera Deutsche Telekom
Fuente: Sitio Web de Deutsche Telekom [10]

2.1. TELEKOM SECURITY

Una de las divisiones de Telekom es Telekom Security, empresa líder del mercado de soluciones de seguridad de TI en Alemania y que reúne la gama completa de experiencia en ciberseguridad del grupo. Esta división está marcando nuevos hitos en este campo con la ayuda de un nuevo Centro de

Operaciones de Seguridad y Defensa Cibernética integrado, el más grande y moderno de su tipo en Europa. La experiencia combinada de su red de más de 1200 especialistas en seguridad internacional y una amplia cartera, que va desde paquetes de seguridad para uso personal de individuos hasta ciberdefensa para empresas de alta tecnología, hacen de Telekom la primera opción para clientes (particulares, pymes y grandes empresas) que buscan productos y soluciones diseñadas para proteger todos los aspectos de sus redes de TI, desde teléfonos inteligentes hasta infraestructuras corporativas. [11]

Entre los productos relacionados con seguridad que ofrece Telekom resaltan los siguientes [12]:

- Seguridad de la red
    - Protección de la red empresarial
    - Protección de red personalizada
    - Seguridad alojada administrada
    - Protección de Internet Pro
    - Protección DDoS
    - Paquete: *Business Router* y *Business Network Protect*
    - Paquete de protección digital empresarial
- Seguridad en el dispositivo final
    - Endpoint Protect Business
    - Paquete de seguridad completo
    - Mobile Protect Pro
- Protección de identidad y acceso
    - Firma digital
    - Clave digital para identidades digitales
    - Certificados digitales
    - Contraseña segura de un solo uso (OTP)
- Seguridad de datos y nube
    - Email Protect Pro
    - APT Protect Pro (nube)
    - APT Protect Pro (en las instalaciones del cliente)
    - Puerta de enlace de correo electrónico cifrado

    - Cifrado en la nube
    - Cloud Protect Pro
- Servicio de ciberdefensa
    - Ciberdefensa gestionada
    - Ciberdefensa inteligente

2.2. SOCIOS DE NEGOCIO

Deutsche Telekom trabaja de la mano con más de 2000 socios líderes en el mercado e invierte en más de 100 start-ups, como ser Amazon Web Services, VM Ware, Microsoft, Cisco, Huawei, Symantec, Hewlett Packard Enterprise, IBM, Ericsson, Chack point, Adobe, Lenovo, Global Data, ISG, entre otros. [13]

2.3. PROYECTOS DE CÓDIGO ABIERTO

Además de ofrecer servicios de paga, Deutsche Telekom tiene diferentes proyectos de código abierto (*Open Source*) que están publicados en la plataforma de alojamiento de código para el control de versiones y colaboración *Git Hub*. Se describe a continuación algunos de los proyectos activos:

T-Pot. Albergado con el nombre de proyecto “tpotce” y descrito como una plataforma *honeypot* todo en uno.

PEBA - Python EWS Backend API. Servicio *backend* ligero de *Python* para recopilar y procesar eventos de ataque capturados por demonios *honyeypot*, en particular aquellos que se ejecutan en la plataforma T-Pot. Además, puede servir como una herramienta de recopilación de datos centralizada para instalaciones T-Pot distribuidas.

EWSPOSTER. Herramienta escrita en *Python* para recopilar registros y alertas de diferentes honeypots (por ejemplo, Glastopf, Honeytrap, Dionaea, Cowrie, Kippo, eMobility, Conpot, Elasticpot, Mailoney, RDPY, VNClowPot, Heralding, Ciscoasa, Tanner y Clutton) y transmíteselos a PEBA.

Explo. Herramienta simple para describir problemas de seguridad web en un formato legible por humanos y máquinas.

Misp-Warning-Lists. Fork “manual” del repositorio original del *MISP (Malware Information Sharing Platform*), desarrollado para proporcionar actualizaciones diarias para las listas, así como corregir y mantener los *scritps* utilizados para generar las listas a través de la plataforma TIP de Telekom.

3. T-POT

Una vez revisados los conceptos que enmarcan los *honeypots* y plasmada la magnitud de la empresa Deutsche Telekom, se procede a describir y analizar T-Pot.

T-Pot es un *honeypot* de código abierto cuyo proyecto es liderado por Telekom Security; los ataques que recibe alimentan los datos del "*Security Dashboard"* disponible en [14] y surgió por la necesidad de tener una herramienta que sea fácil de implementar, de bajo mantenimiento y que combine algunas de las mejores tecnologías de *honeypot* en un solo sistema. [15]

3.1. HISTORIA Y VERSIONES

Durante muchos años la empresa Telekom Security había configurado varios *honeypots* en sus redes domésticas, redes de acceso a Telekom y en las instalaciones de sus socios en todo el mundo. Lo cual les hizo notar que muchas personas estaban interesadas en ejecutar algún tipo de sensor de *honeypot*, pero terminaban abrumadas por el procedimiento de instalación y mantenimiento, lo cual los impulsó a crear una plataforma *multi-honeypot* que sea fácil de implementar, de bajo mantenimiento y que combine algunas de las mejores tecnologías de *honeypot* en un solo sistema. Es así que en marzo de 2015 hacen pública la herramienta y ponen a disposición de la comunidad el código fuente como una versión beta [15]. Desde entonces se realizaron muchas mejoras y actualizaciones, se publicaron las versiones 16.10, 16.10.1, 17.10, 18.11, 19.03, 19.03.1, 19.03.3, 20.06.0 y 20.06.1, para finalmente llegar a la versión 20.06.2 publicada el 22 de febrero de 2021; que es la última versión al momento de elaborar el presente documento, pero que sufre actualizaciones permanentes en la plataforma de desarrollo colaborativo GitHub.

3.2. ARQUITECTURA

La arquitectura y modo en que se implementan varios *honeypot* en un solo sistema hace de T-Pot una de las soluciones más implementadas y con más apoyo a nivel mundial. En los siguientes puntos se analizará de manera

separada la arquitectura, *honeypots*, herramientas y elementos que componen el sistema señuelo estudiado en su versión 20.06.2.

T-Pot 20.06 en adelante está probado para ser instalado en el sistema operativo Linux Debian 10 (*Buster)*; los demonios *honeypot,* así como otros componentes que soporta se ejecutan en contenedores Docker y hacen uso de Docker-Compose, lo cual permite correr múltiples demonios y herramientas en una misma interfaz de red, además de reducir la huella digital de las soluciones y restringir a cada honeypot dentro de su propio entorno. [16]

Diferentes *honeypots,* la solución ELK Stack y herramientas NSM (Network Security Monitor) interactúan de manera conjunta y sinérgica para dar lugar a T-Pot. Adicionalmente, ciertas instalaciones incluyen las herramientas Cockpit, Cyberchef, Elasticsearch Head, Spiderfoot y Suricata, las cuales coadyuvan en el análisis, monitoreo y administración de la solución de *honeypot*, así como en la etapa de envío de hallazgos a la comunidad.

El Anexo “A” plasma de manera gráfica la arquitectura de T-Pot y se analiza a continuación cada uno de los elementos que lo componen.

3.2.1. HONEYPOTS

La versión 20.06.2 de T-Pot incluye versiones *dockerizadas* de los *Honeypots* adbhoney, ciscoasa, citrixhoneypot, conpot, cowrie, ddospot, dicompot, dionaea, elasticpot, endlessh, glutton, heralding, hellpot, honeypots, honeypy, honeysap, honeytrap, ipphoney, log4pot, mailoney, medpot, rdpy, redishoneypot, snare y tanner. Se analiza a continuación cada uno de ellos.

3.2.1.1. ADBHoney

El protocolo Android Debug Bridge (ADB) permite la comunicación con un dispositivo que tiene Android como sistema operativo (celulares, tabletas, televisores, etc.) e implementa varios comandos diseñados para ayudar al desarrollador. La comunicación se la realiza a través de un cable USB, con amplios mecanismos de autenticación y protección; sin embargo, la conexión TCP/IP no tiene ningún tipo de autenticación y deja al dispositivo propenso a todo tipo de ataques. [17] [18]

ADBHoney es un *honeypot* de baja interacción que emula el protocolo Android Debug Bridge (ADB) sobre TCP/IP (puerto 5555), con la ayuda de Python. Permite la ejecución de comandos “adb connect”, “adb push” y “adb shell”, pero a través del shell solo se podrá ejecutar el comando “ls”, el cual devolverá el contenido del archivo “responses.py”. [19] Los logs y archivos cargados al *honeypot* se almacenan en las rutas “/opt/adbhoney/log” y “/opt/adbhoney/dl” del contenedor y están mapeados a los volúmenes del sistema anfitrión (Debian con T-Pot) “/data/adbhoney/log” y “/data/adbhoney/downloads” respectivamente.

3.2.1.2. CiscoASA Honeypot

El Cisco ASA (Adaptive Security Appliance) es un dispositivo de seguridad que combina capacidades de firewall, antivirus, prevención de intrusiones y red privada virtual (VPN). Proporciona una defensa proactiva contra amenazas que detiene los ataques antes de que se propaguen por la red. [20]

CiscoASA Honeypot es un *honeypot* de baja interacción que emula; con la ayuda de Python, un dispositivo Cisco ASA capaz de detectar la vulnerabilidad con CVE 2018-0101, una vulnerabilidad de ejecución remota de código y DoS. [21] Publica los puertos 8443 (TCP) y 5000 (UDP) y los logs se almacenan en la ruta “/var/log/ciscoasa” del *honeypot,* la cual está mapeada al volumen “/data/ciscoasa/log” del sistema anfitrión.

3.2.1.3. Citrix Honeypot

Citrix ADC; producto de red principal de Citrix Systems, es un Controlador de Entrega de Aplicaciones (por sus siglas en inglés Application Delivery Controller). Se trata de una herramienta que mejora la velocidad de entrega y la calidad de las aplicaciones al usuario final, realiza tareas de optimización de tráfico, equilibrio de carga L4-L7, aceleración de aplicaciones web, realiza varios tipos de almacenamiento en caché y compresión, puede convertirse en un servidor proxy, incluye firewall de aplicaciones NetScaler, capacidades de encriptación SSL, puede procesar solicitudes SSL, ofrecer operaciones VPN y de micro aplicaciones VPN, puede administrar el tráfico durante los ataques DDoS, asegurándose de que el tráfico llegue a las

aplicaciones críticas. Además, los registros de actividad de la red de Netscaler se incorporan al servicio de análisis basado en la nube de Citrix y se utilizan para analizar e identificar riesgos de seguridad. [22]

Citrix Honeypot es un *honeypot* de baja interacción que; con la ayuda de Python, emula un dispositivo Citrix ADC que detecta y registra los intentos de exploración y explotación de la vulnerabilidad con CVE 2019-19781, misma que permite un ataque de *Path Traversal*. [23] [24] Publica el puerto 443 y los logs se almacenan en la ruta "/opt/citrixhoneypot/logs" del *honeypot,* volumen mapeado a "/data/citrixhoneypot/logs" del sistema anfitrión.

3.2.1.4. Conpot

Sistema de Control Industrial (ICS por sus siglas en inglés "*Industrial Control System*") es un término colectivo que se utiliza para describir diferentes tipos de sistemas de control e instrumentación asociada, que incluye los dispositivos, sistemas, redes y controles utilizados para operar y/o automatizar procesos industriales. Dependiendo de la industria, cada ICS funciona de manera diferente y está diseñado para administrar tareas de manera electrónica de manera eficiente. En la actualidad, los dispositivos y protocolos utilizados en un ICS se utilizan en casi todos los sectores industriales e infraestructura crítica, como las industrias de fabricación, transporte, generación de energía, telecomunicaciones, tratamiento de agua, procesamiento químico, de gas, petróleo, etc. De los diferentes tipos de ICS, lo más comunes son los sistemas SCADA (Supervisión, Control y Adquisición de Datos) y DCS (Sistemas de Control Distribuido). [25]

El sistema de Supervisión, Control y Adquisición de Datos SCADA (por sus siglas en inglés *Supervisory Control And Data Acquisition*) es una herramienta de automatización y control industrial utilizada en los procesos productivos que puede controlar, supervisar, recopilar datos, analizarlos y generar informes a distancia mediante una aplicación informática. Su principal función es la de evaluar los datos con el propósito de subsanar posibles errores. Formalmente se define como una agrupación de aplicaciones informáticas instaladas en un ordenador denominado Máster o MTU, destinado al control automático de una actividad productiva a distancia que

está interconectada con otros instrumentos llamados de campo como son los autómatas programables (PLCs) y las unidades terminales remotas (RTUs). [26]

Conpot es un *honeypot* de baja interacción que con la ayuda de Python emula Sistemas de Control Industrial SCADA, está diseñado para ser fácil de implementar, modificar y ampliar. Proporciona una gama de protocolos de control industrial comunes, lo que permite construir un sistema capaz de emular infraestructuras complejas para convencer a un adversario de que acaba de encontrar un enorme complejo industrial. También proporciona la posibilidad de servir como servidor una interfaz hombre-máquina personalizada para aumentar la superficie de ataque de los *honeypots*. Los tiempos de respuesta de los servicios se pueden retrasar artificialmente para imitar el comportamiento de un sistema bajo carga constante. Debido a que proporcionamos pilas completas de protocolos, se puede acceder a Conpot con HMI productivas o se puede ampliar con hardware real. Conpot se desarrolla bajo el paraguas de "*Honeynet Project*" y sobre los hombros de un par de gigantes. [27] Tiene diferentes plantillas y T-Pot implementa cada una de ellas a través de diferentes instancias Docker, las cuales publican diferentes puertos. La plantilla "default" simula un PLC Siemens S7-200 con dos dispositivos esclavos y publica los puertos 80, 102, 161 (UDP), 502, 21, 44818 y 47808 (UDP); la plantilla "IEC104" simula el dispositivo IEC 60870-5-104 y publica el puerto 2404; la plantilla "guardian_ast" emula un dispositivo diseñado para el cumplimiento y control de inventario para tanques de almacenaje, monitorea los niveles de las bombas, sistemas de bombeo y el inventario de tanques como los utilizados en las gasolineras y publica el puerto 10001; la plantilla "ipmi" crea un dispositivo de Interfaz de Administración de Plataformas Inteligentes (Intelligent Platform Management Interface, IPMI) que permite a un operador gestionar remotamente servidores a nivel de hardware y publica el puerto 623 (UDP); la plantilla "kamstrup_382" es un clon de un medidor de energía eléctrica inteligente modelo Kamstrup 382 y publica los puertos 1025 y 50100. [28] Los logs los guarda en la ruta "/var/log/conpot", misma que está mapeado a "/data/conpot/log" del sistema anfitrión.

3.2.1.5. Cowrie

Telnet, desarrollado en 1969, es un protocolo (inseguro) que (por defecto) utiliza el puerto 23 (TCP) y proporciona una interfaz de línea de comandos para la comunicación con un dispositivo o servidor remoto, generalmente se emplea para la administración remota, pero también es usado para la configuración inicial de un dispositivo. [29]

SSH o *Secure Shell*, es un protocolo de red criptográfico que (por defecto) utiliza el puerto 22 (TCP) y proporciona una interfaz de administración que típicamente es usada para inicio de sesión y ejecución remota de comandos, pero puede ser empleado para proteger cualquier servicio de red. [30] [31]

Cowrie es un *honeypot* SSH y Telnet de interacción media a alta diseñado para registrar ataques de fuerza bruta y la interacción de shell realizada por el atacante. En el modo de interacción media (shell) emula un sistema UNIX en Python, en el modo de interacción alta (proxy) funciona como un proxy SSH y telnet para observar el comportamiento del atacante a otro sistema. [32] La implementación de T-Pot publica los puertos 22 y 23 y mapea los volúmenes "/data/cowrie/downloads", "/data/cowrie/keys", "/data/cowrie/log" y "/data/cowrie/log/tty" a las siguientes unidades del contenedor "/home/cowrie/cowrie/dl", "/home/cowrie/cowrie/etc", "/home/cowrie/cowrie/log" y "/home/cowrie/cowrie/log/tty".

3.2.1.6. DDoSPot

Los ataques de red distribuidos se conocen como DDoS - *Distributed Denial of Service*. Este tipo de ataques aprovecha los límites de capacidad específicos que se aplican a cualquier recurso de red; se ejecuta enviando varias solicitudes al recurso atacado, con la intención de desbordar su capacidad para administrar varias solicitudes y de evitar que este funcione correctamente para los clientes legítimos. [33]

DDoSPot es una plataforma trampa para rastrear y monitorear ataques de denegación de servicio distribuido (DDoS) basados en UDP. La plataforma actualmente admite los siguientes servicios/servidores de honeypot en forma de complementos relativamente simples llamados pots: servidor DNS,

servidor NTP, servidor SSDP, servidor Chargen, servidor UDP aleatorio/simulado. [34] Su implementación publica los puertos 19 (UDP), 53 (UDP), 123 (UDP), 1900 (UDP) y mapea los volúmenes “/data/ddospot/log”, “/data/ddospot/bl” y “/data/ddospot/db” a las unidades “/opt/ddospot/ddospot/logs”, “/opt/ddospot/ddospot/bl” y “/opt/ddospot/ddospot/db” del contenedor.

3.2.1.7. Dicompot

Imágenes y Comunicaciones Digitales en Medicina DICOM (por sus siglas en inglés Digital Imaging and Communications in Medicine) es el estándar internacional que establece las reglas, que permiten el intercambio de imágenes médicas e información asociada entre equipos de diferentes proveedores, computadoras y hospitales. Los archivos DICOM se pueden intercambiar entre dos entidades que pueden recibir imágenes y datos del paciente en formato DICOM. [35] [36] También se conoce como estándar NEMA PS3 y estándar ISO 12052: 2017. [37]

Dicompot es un *honeypot* que emula un servidor DICOM completamente funcional, haciendo uso del lenguaje de programación “Go”. [38] Publica el puerto 11112 (TCP) y guarda los logs en “/var/log/dicompot”, que está mapeado a “/data/dicompot/log” del sistema anfitrión. Para realizar pruebas se deben emplear clientes como “Horos”, “OsiriX”, “Navegatium”, “MicroDicom”, entre otros.

3.2.1.8. Dionaea

Dionaea es un *honeypot* de baja interacción que tiene el objetivo de atrapar y obtener una copia del malware que explota vulnerabilidades expuestas por los diferentes servicios emulados. Es el sucesor de otro *honeypot* denominado Nepenthes, está escrito en C, incorpora Python como lenguaje de scripting, utiliza la biblioteca “libemu” para emular la ejecución de instrucciones y detectar *shellcodes*; además, la última versión cuenta con soporte para IPv6 y TLS. [39] [40] [41]

Dionaea implementado en T-Pot, emula los siguientes servicios. [39]

- FTP (*File Transfer Protocol*), publica los puertos 20 y 21 y permite crear directorios, cargar y descargar archivos. Mapea el volumen "/opt/dionaea/var/dionaea/roots/ftp" a "/data/dionaea/roots/ftp".
- HTTP (*Hypertext Transfer Protocol*), publica los puertos 81 y 443 y mapea el volumen "/opt/dionaea/var/dionaea/roots/www" a "/data/dionaea/roots/www".
- MSSQL (*Microsoft SQL Server*), este módulo implementa el protocolo Tabular Data Stream que es utilizado por Microsoft SQL Server. Publica el puerto 1433 y permite que los clientes inicien sesión. Puede decodificar las consultas que se ejecutan en la base de datos, pero como no hay una base de datos, Dionaea no puede responder y no hay más acciones.
- MySQL, el módulo está respaldado por sqlite como base de datos y publica el puerto 3306.
- PPTP (*Point-to-Point Tunneling Protocol*), por defecto emula una conexión con equipos Linux, pero puede configurarse con Cisco PIX, DrayTek, Microsoft o MikroTik, en cualquier caso publica el puerto 1723.
- SIP (*Session Initiation Protocol*), este módulo implementa SIP como protocolo VoIP. A diferencia de otros honeypots de VoIP, no se conecta a un registrador / servidor de VoIP externo, simplemente espera los mensajes SIP entrantes (por ejemplo, OPTIONS o incluso INVITE), registra todos los datos como incidentes de honeypot y / o volcados de datos binarios (tráfico RTP) y reacciona en consecuencia. Publica los puertos 5060, 5060 (UDP) y 5061 y mapea el volumen "/opt/dionaea/var/dionaea" a "/data/dionaea", donde se encuentra el archivo "sipaccounts.sqlite", que contiene cuentas SIP.
- SMB (*Server Message Block*), es el principal protocolo de Dionaea debido a su historial de errores explotables remotos y a que es un objetivo muy popular para los gusanos. Hace uso de *scapy* (una herramienta de manipulación de paquetes para redes informáticas) adaptada a python3 y además de los ataques conocidos a las pymes, Dionaea admite la carga de archivos en recursos compartidos de

pymes. Por defecto publica el puerto 445, pero puede ser modificado en el archivo de configuración.

- TFTP (*Trivial File Transfer Protocol*), este módulo proporciona un servidor TFTP, publica el puerto 69 (UDP) para servir archivos y mapea el volumen "/opt/dionaea/var/dionaea/roots/tftp" a "/data/dionaea/roots/tftp".
- Adicionalmente, Dionaea permite configurar los servicios MongoDB, MQTT (*Message Queing Telemetry Transport*), Memcache, impresoras y UPnP, para lo cual publica los puertos 42, 135, 1883 y 27017. Además de mapear los volúmenes "/opt/dionaea/var/dionaea/roots/upnp", "/opt/dionaea/var/dionaea/binaries", ":/opt/dionaea/var/log" y "/opt/dionaea/var/dionaea/rtp" a los directorios de sistema anfitrión "/data/dionaea/roots/upnp", "/data/dionaea/binaries", "/data/dionaea/log" y "/data/dionaea/rtp".

3.2.1.9. ElasticPot

Elasticsearch es un motor de analítica y análisis distribuido, gratuito y abierto para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y no estructurados. Conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad, Elasticsearch es el componente principal del Elastic Stack, un conjunto de herramientas gratuitas y abiertas para la ingesta, el enriquecimiento, el almacenamiento, el análisis y la visualización de datos. [42]

ElasticPot es un honeypot que simula un servidor Elasticsearch vulnerable abierto a Internet. Utiliza ideas de otros *honeypots*, como ADBHoneypot (para compatibilidad con complementos de salida), Citrix Honeypot (para estructura general), Elastichoney, (para un ejemplo general de un honeypot de Elasticsearch). ElasticpotPY (para la idea de usar respuestas con script almacenadas en archivos) y Delilah (para ideas adicionales sobre qué emular). [43] Publica el puerto 9200 y mapea el volumen "/opt/elasticpot/log" a "/data/elasticpot/log" del sistema anfitrión.

3.2.1.10. EndleSSH

SSH o Secure Shell, es un protocolo de red criptográfico que (por defecto) utiliza el puerto 22 (TCP) y proporciona una interfaz de administración que típicamente es usada para inicio de sesión y ejecución remota de comandos, pero puede ser empleado para proteger cualquier servicio de red. [30] [31]

Endlessh es un *tarpit* (servicio de red que intencionalmente inserta demoras en su protocolo, ralentizando a los clientes obligándolos a esperar [44]) SSH que envía muy lentamente un banner SSH aleatorio e interminable, manteniendo a los clientes SSH bloqueados durante horas o incluso días. El propósito es dejar que los *script kiddies* se atasquen en este *tarpit* en lugar de molestar a un servidor real. [45] En la implementación de T-Pot mapea el puerto 2222 del contenedor al puerto 22 del sistema publicado y mapea el volumen “/data/endlessh/log” al directorio “/var/log/endlessh”.

3.2.1.11. Glutton

Glutton es un *honeypot* que actúa como proxy entre atacante y otro *honeypot*, proporcionando la capacidad de capturar, registrar y analizar el tráfico enviado. Básicamente, escucha todos los puertos y luego actúa de acuerdo con un archivo de reglas “rules.yaml”. Actualmente tiene soporte de proxy para SSH y TCP, además de ofrecer la posibilidad de inicio de sesión para el protocolo SSH. Para manipular los paquetes y cumplir la funcionalidad de proxy hace uso de varias librerías, siendo la principal “freki”, la cual manipula los paquetes en modo de usuario haciendo uso de NFQueue (un objetivo de *iptables* e *ip6tables* que delega la decisión sobre los paquetes a un software de espacio de usuario). [46] [47] [48] El contenedor Docker configurado en T-Pot emplea el modo de red “host”, es decir que comparte la pila de red con el sistema anfitrión y no se le asigna una dirección IP propia. Además, mapea el volumen “/var/log/glutton” a “/data/glutton/log” y el archivo “/opt/glutton/rules/rules.yaml” a través de “/root/tpotce/docker/glutton/dist/rules.yaml”. Este NFQ se levanta cuando se realiza la instalación de tipo “NextGen”.

3.2.1.12. Heralding

Heralding es un *honeypot* de baja interacción que tiene el objetivo de recopilar credenciales, para ello emula con la ayuda de Python diferentes servicios que solicitan inicio de sesión. [49] Los servicios publicados son FTP en el puerto 21, SSH en el 22, Telnet en el 23, SMTP en el 25, HTTP en el 80, HTTPS en el 443, POP3 en el 110, POP3S en el 995, IMAP en el 143, IMAPS en el 993, Socks5 en el 1080, MySQL en el 3306, RDP en el 3389, PostgreSQL en el 5432 y VNC en el 5900. Para registrar los logs, mapea el volumen "/var/log/heralding" a "/data/heralding/log" del sistema anfitrión.

3.2.1.13. HellPot

HellPot es un *honeypot* basado en Heffalump que "envía al infierno" a los bots HTTP rebeldes. En particular, implementa un archivo de configuración "config.toml", tiene registro JSON y viene con ganancias de rendimiento significativas. Los clientes (con suerte bots) que ignoren robots.txt y se conecten a la instancia de HellPot sufrirán consecuencias eternas ya que HellPot enviará un flujo infinito de datos que está lo suficientemente cerca de ser un sitio web real que podrían quedarse hasta que "su alma se desgarre y dejen de existir". Bajo el capó de este "sufrimiento eterno" hay un motor de *markov* que arroja fragmentos de "El nacimiento de la tragedia (helenismo y pesimismo)" de Friedrich Nietzsche al cliente mediante fasthttp. [50] En la implementación de T-Pot mapea el puerto 8080 del contenedor al puerto 80 del sistema publicado y el volumen "/data/hellpot/log" al directorio "/var/log/hellpot".

3.2.1.14. Honeypots

Honeypots en una solución desarrollada en Python que permite la implementación de 23 *honeypots* en un solo paquete PyPI con los cuales se puede monitorear el tráfico de la red, las actividades de bots y credenciales introducidas. Los resultados obtenidos se pueden registrar en una base de datos postgres, archivos, terminal o syslog y los servicios que emula son: dns, ftp, httpproxy, http, https, imap, mysql, pop3, postgres, redis, smb, smtp, socks5, ssh, telnet, vnc, mssql, elastic, ldap, ntp, memcache, snmp, y oracle. [51] La implementación en T-Pot publica los puertos 21, 22, 23, 25, 53 (UDP),

80, 110, 143, 389, 443, 445, 1080, 1433, 3306, 5432, 5900, 6369, 8080, 9200 y mapea el volumen “/data/honeypots/log” al directorio “/var/log/honeypots” del contenedor.

3.2.1.15. HoneyPy

HoneyPy es un *honeypot* de interacción baja o media, determinada por la funcionalidad de los complementos empleados. Está desarrollado en Python2 y emula los servicios “echo” en los puertos 7 TCP y UDP, Telnet Unix en el puerto 2323, Telnet Windows en el 2324 y Elasticsearch en el 9200, pero además puede emular otros servicios como DNS, NTP, SMTP, TFTP, Web etc. [52] Si bien T-Pot emplea otros *honeypot* para publicar los servicios provistos por defecto, el volumen “/opt/honeypy/log” está mapeado a “/data/honeypy/log” para usarlo cuando se lo necesite.

3.2.1.16. HoneySAP

SAP o Sistemas, Aplicaciones y Productos en Procesamiento de Datos (por sus siglas en inglés *Systems Applications and Products in Data Processing*) es el nombre de una empresa multinacional alemana fundada en 1972 por antiguos empleados de IBM. Dicha empresa desarrolla software ERP (Planificación de Recursos Empresariales o *Enterprise Resource Planning* por sus siglas en inglés), el cual consta de varios módulos totalmente integrados que cubren prácticamente todos los aspectos de la gestión empresarial (recursos humanos, productivos, logísticos, etc.). El sistema SAP es el número uno en el mercado de ERP. En 2010, SAP tenía más de 140.000 instalaciones en todo el mundo, más de 25 soluciones comerciales específicas de la industria y más de 75.000 clientes en 120 países, lo cual hace que sea un objetivo de ataque muy codiciado. [53] [54]

HoneySAP es un *honeypot* de baja interacción centrado en la investigación, específico para servicios SAP. El objetivo principal es permitir que los profesionales de la seguridad, investigadores y organizaciones conozcan las técnicas y motivaciones detrás de los ataques contra los sistemas SAP. [55] La implementación en T-POT, publica el servicio “*SAPRouterService*” en el puerto 3299 y mapea el volumen “/opt/honeysap/log” a “/data/honeysap/log”.

3.2.1.17. Honeytrap

Honeytrap es un *honeypot* de baja interacción, escrita en “C” y destinado a detectar ataques contra servicios TPC y UDP. En su configuración predeterminada, se ejecuta como un demonio e inicia los procesos del servidor cuando se realiza un intento de conexión a un puerto. Tiene diferentes modos de operación disponibles que controlan cómo se manejan las conexiones. En modo “normal”, envía datos arbitrarios proporcionados en archivos de plantilla como un medio básico para emular protocolos conocidos. Otro modo de operación popular es el llamado “modo espejo”, en el que las conexiones entrantes se devuelven al iniciador, este truco elimina la necesidad de emular el protocolo en muchos casos. Un tercer modo, es el modo “proxy”, el cual permite el reenvío de sesiones específicas a otros sistemas, por ejemplo, *honeypots* de alta interacción. Para tomar los intentos de conexión entrantes, se hace uso de la función “NFQueue” de “iptables”, la cual a través de una regla coloca los segmentos TCP-SYN entrantes en una cola donde pueden ser recogidos por Honeytrap. [56] [57]

El contenedor Docker de Honeytrap configurado en T-Pot, emplea el modo de operación “normal” y la red está configurada en modo “host”, es decir que comparte la pila de red con el sistema anfitrión y no se le asigna una dirección IP propia. Por otra parte, mapea los volúmenes “/opt/honeytrap/var/attacks”, “/opt/honeytrap/var/downloads” y “/opt/honeytrap/var/log” a los directorios “/data/honeytrap/attacks”, “/data/honeytrap/downloads” y “/data/honeytrap/log” respectivamente. Este NFQ se levanta en cuando se realiza la instalación de tipo “Standard”, “Sensor”, “Industrial” o “Collector”.

3.2.1.18. IPPHoney

El Protocolo de Impresión de Internet o IPP (por sus siglas en inglés “Internet Printing Protocol”) es un protocolo de Internet especializado en la comunicación entre los dispositivos del cliente (computadoras, teléfonos móviles, tabletas, etc.) e impresoras (o servidores de impresión). Permite a los clientes enviar uno o más trabajos de impresión a la impresora o al servidor de impresión y realizar tareas como consultar el estado de una impresora,

obtener el estado de los trabajos de impresión o cancelar trabajos de impresión individuales. Como todos los protocolos basados en IP, IPP puede ejecutarse localmente o por Internet. A diferencia de otros protocolos de impresión, IPP también admite el control de acceso, autenticación y cifrado, lo que lo convierte en un mecanismo de impresión mucho más capaz y seguro. [58]

IPPHoney es un *honeypot* que simula una impresora que admite el Protocolo de Impresión de Internet (IPP) y está expuesta a Internet. Utiliza ideas de varios otros honeypots, como ADBHoneypot (para compatibilidad con complementos de salida), Citrix Honeypot (para estructura general) y Elasticpot. [59] La implementación en T-Pot publica el puerto 631 y mapea el volumen “/opt/ipphoney/log” al directorio “/data/ipphoney/log”.

3.2.1.19. Log4Pot

La vulnerabilidad Log4Shell tiene asociado el CVE-2021-44228 y se trata de una vulnerabilidad de tipo ejecución remota de código que afecta el software Apache Log4J. [60]

Log4Pot es un honeypot para la vulnerabilidad Log4Shell (CVE-2021-44228) que mapea los puertos 80, 443, 8080, 9200 y 25565 al puerto 8080 del contenedor, así como los volúmenes “/data/log4pot/log” y “/data/log4pot/payloads” a los directorios “/var/log/log4pot/log” y “/var/log/log4pot/payloads” respectivamente.

3.2.1.20. Mailoney

SMTP (Protocolo Simple de Transferencia de Correo o Simple Mail Transfer Protocol por sus siglas en inglés) es un protocolo de comunicación estándar de Internet para la transmisión de correo electrónico. Los servidores de correo y otros agentes de transferencia de mensajes utilizan SMTP para enviar, recibir y/o retransmitir correo saliente entre remitentes y receptores de correo electrónico. [61] [62]

Mailoney es un *honeypot* SMTP, escrito en Python y que tiene tres de módulos principales. El primero es “open_relay”, el cual intentará registrar todo el texto de los correos electrónicos que se intentan enviar; el segundo es “postfix_creds”, que registra las credenciales de los intentos de inicio de

sesión y el tercer módulo es “schizo_open_relay”, el cual registra toda la interacción con Mailoney. [63] La implementación de T-Pot publica el puerto 25 y mapea el volumen “/opt/mailoney/logs” al directorio “/data/mailoney/log” del equipo anfitrión.

3.2.1.21. Medpot

FHIR o recursos de interoperabilidad de atención médica rápida (Fast Healthcare Interoperability Resources por sus siglas en inglés) fue creado por HL7 (Health Level Seven International) y es un estándar que describe formatos y elementos de datos (conocidos como "recursos"), además de una interfaz de programación de aplicaciones (API) para intercambiar registros médicos electrónicos (EHR). Lo que hace especial a FHIR, es que fue adoptado y promovido por Apple y CMS como su mecanismo de interfaz de atención médica preferido, ambas empresas lanzaron una aplicación de soluciones móviles que permite a los pacientes conectarse y administrar sus registros clínicos de forma segura, haciendo uso de FHIR. [64] [65]

Medpot es un *honeypot* que haciendo uso del lenguaje “Go”, emula la interfaz de programación de aplicaciones (API) FHIR. [66] Su implementación en T-Pot publica el puerto 2575 y registra los logs en “/data/medpot/log/”, directorio que está mapeado al volumen “/var/log/medpot” del *honeypot* implementado con Docker.

3.2.1.22. RDPY

Protocolo de Escritorio Remoto o RDP (por sus siglas en inglés Remote Desktop Protocol) es un protocolo propietario desarrollado por Microsoft que proporciona al usuario una interfaz gráfica para conectarse a otra computadora a través de una conexión de red (puerto 3389 TCP y UDP). [67] [68]

RDPY es un *honeypot* completamente implementado en Python (excepto el algoritmo de descompresión de mapa de bits que se implementa en C con fines de rendimiento) que emula el protocolo Microsoft RDP del lado del cliente y del servidor. RDPY se basa en el motor de red impulsado por eventos Twisted, admite la capa de seguridad RDP estándar, RDP sobre SSL y autenticación NLA (a través del protocolo de autenticación ntlmv2). Por otra

parte, proporciona los siguientes binarios RDP y VNC: RDP Man In The Middle proxy que registra la sesión, Honeypot de RDP, captura de pantalla de RDP, cliente RDP, cliente VNC, captura de pantalla de VNC y reproductor de RSS. [69] Su implementación en T-Pot publica el puerto 3389 y mapea el volumen "/var/log/rdpy" al directorio "/data/rdpy/log" del host.

3.2.1.23. RedisHoneypot

Redis es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente. Está escrito en ANSI C por Salvatore Sanfilippo, quien es patrocinado por Redis Labs y está liberado bajo licencia BSD por lo que es considerado software de código abierto. Los clientes se conectan a un servidor Redis creando una conexión TCP al puerto 6379 y mediante un protocolo denominado RESP (Protocolo de serialización de REdis). Si bien el protocolo se diseñó específicamente para Redis, se puede usar para otros proyectos de software cliente-servidor. [70] [71]

RedisHoneyPot es un sistema *honeypot* altamente interactivo que admite el protocolo Redis. Está desarrollado en lenguaje Golang y simula la ejecución de los siguientes comandos: *ping, info, set, get, del, exists, keys, flushall, flushdb, save, select, dbsize, config* y *slaveof.* [72] La implementación en T-Pot publica el puerto 6379 y mapea el volumen "/data/redishoneypot/log" a "/var/log/redishoneypot".

3.2.1.24. Snare y Tanner

Snare es un *honeypot* de aplicaciones web (sucesor de otro *honeypot* denominado Glastopf) que emula vulnerabilidades (conocidas como "superficie de ataque") a las que un usuario no autorizado puede acceder y posiblemente explotar. Una de sus características es que cuando los atacantes le toman las huellas digitales, este muestra que es un servidor de aplicaciones web Nginx, lo cual le ayuda a no ser detectado como un *honeypot.* Además, viene con un programa en Python denominado "clone.py", que permite clonar sitios web y usarlos como superficies de ataque. [73] [74]

Por su parte, Tanner es un servicio de clasificación y análisis de datos remotos, que evalúa las solicitudes HTTP y compone una respuesta. Puede clasificar los ataques en función de sus firmas como LFI (Inyección de archivos locales), RFI (Inyección remota de archivos), XSS (Cross-site Scripting), CMD_EXEC (Ejecución de comandos) y SQLI (Inyección de lenguaje de consulta estructurado). Utiliza Redis como servidor de base de datos por defecto, sin embargo también puede almacenar el tráfico web que se le proporciona en una base de datos MongoDB. [74] [75]

Ambas soluciones trabajan juntas, cada evento que recibe Snare es enviado a Tanner, el cual lo analiza, clasifica, evalúa y decide cómo Snare debe responder al cliente en función a reglas configuradas, lo cual permite que el *honeypot* produzca respuestas dinámicas que mejoran su camuflaje. [74]

La implementación en T-Pot publica el puerto 80 y mapea los volúmenes "/var/log/tanner" y "/opt/tanner/files" a los directorios "/data/tanner/log" y "/data/tanner/files" respectivamente. En cuanto a los sitios web que publica, lo hace de manera al azar de un conjunto de sitios predefinidos y cargados durante la instalación de T-Pot.

3.2.2. ELK STACK

ELK Stak es una solución liderada por la empresa Elastic y compuesta por tres proyectos de código abierto: Elasticsearch, Logstash y Kibana. Los tres productos forman una pila de un extremo al otro y conforman una herramienta de análisis de datos en tiempo real, que proporciona información procesable de casi cualquier tipo de fuente de datos. [76] [77] [78] [79] [80]

Logstash es un pipeline de procesamiento de datos del lado del servidor, que ingesta datos de una multitud de fuentes simultáneamente, los transforma y luego los envía para que algún motor de búsqueda lo utilice. [81]

Elasticsearch es un motor de búsqueda y análisis distribuido que acepta todos los tipos de datos (textuales, numéricos, geoespaciales, estructurados y no estructurados). Está desarrollado a partir de Apache Lucene y es conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad. [42]

Kibana es una herramienta que permite a los usuarios visualizar los datos de Elasticsearch en cuadros y gráficos. [82]

Adicionalmente, el año 2015 se introdujeron una gran colección de agentes ligeros conocidos como Beats, los cuales permiten enviar los datos a Elasticsearch. Tras introducir los agentes, la solución pasó a llamarse Elastic Stack y su arquitectura se muestra en la siguiente figura. [76]

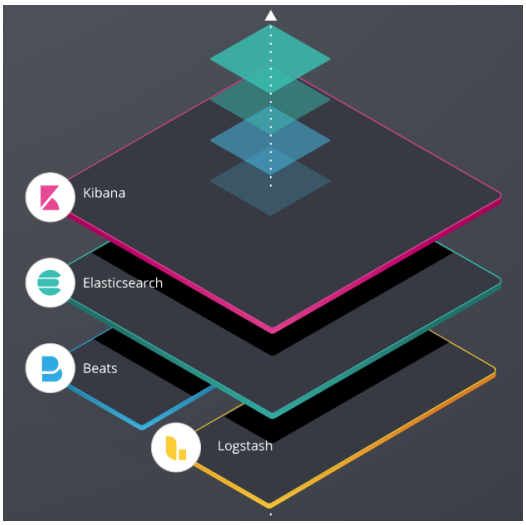


Figura 2: Arquitectura de Elastic Stack
Fuente: Sitio Web de la empresa Elastic

En general, Elastic Stack es empleado por grandes empresas como Uber, Cisco, Mercado Libre, Tinder, Telefónica, Entel, Orange y Audi entre otras para analizar todo tipo de datos. [80] Por su parte, T-Pot lo emplea para poder analizar y mostrar de manera gráfica los "*logs*" generados por cada uno de los *honeypots*.

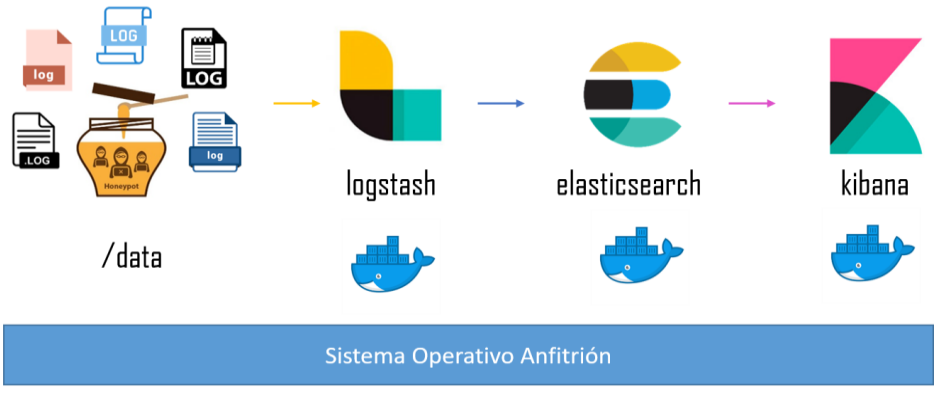


Figura 3: Flujo para la visualización de logs con Kibana en T-Pot
Fuente: Elaboración propia

La figura previa muestra el flujo que siguen los *logs* de los diferentes *honeypots* para ser visualizados con Kibana. Primero, cada *honeypot* almacena sus registros de actividad en el directorio "/data/nombreDelHoneypot" del Sistema Operativo anfitrión, donde accede Logstash, procesa los datos y los almacena en la base de datos de Elasticsearch, a la cual accede Kibana para mostrar en forma de cuadros y gráficos los ataques registrados por cada *honeypot*. De manera similar a los *honeypot*, cada herramienta del ELK Stack es implementada en un contenedor Docker independiente.

3.2.3. NETWORK SECURITY MONITOR (NSM)

T-Pot emplea FATT, p0f y Suricata como herramientas para la toma de huellas digitales y monitoreo de seguridad de red. Se analiza a continuación estas tres soluciones.

3.2.3.1. Fatt

FATT (por sus siglas en inglés Fingerprint All The Things), es un script basado en pyshark, que permite extraer metadatos de red y huellas digitales de archivos de captura de paquetes (pcap) o tráfico de red en vivo. Se usa principalmente para monitorear honeypots, pero también es empleado para análisis forense de redes. Admite una serie de protocolos de red como HTTP, SSH, gQUIC, SSL/TLS y RDP, lo que lo hace muy eficaz en la ejecución de operaciones de análisis basadas en la red. Tiene la capacidad de ejecutar procedimientos de búsqueda de amenazas dentro de un sistema de red y los métodos de toma de huellas digitales que incorpora son JA3 (huella digital de cliente / servidor TLS), HASSH (huella digital de cliente / servidor SSH), RDFP (huella digital RDP experimental para el protocolo de seguridad RDP estándar) y otro método que extrae huellas digitales de encabezados HTTP. [83] [84] T-Pot lo implementa como un contendor Docker, en el cual el modo de red es "Host", es decir que comparte la pila de red con el sistema anfitrión y no se le asigna una dirección IP propia. Por su parte, mapea el volumen "/opt/fatt/log" a "/data/fatt/log".

3.2.3.2. p0f

P0f (acrónimo de Passive Operating System Fingerprinting) es una herramienta escrita en C, que utiliza una variedad de mecanismos sofisticados y puramente pasivos para la toma de huellas digitales de tráfico, permite identificar a las partes detrás de cualquier comunicación TCP / IP (a menudo tan solo un SYN normal) sin interferir de ninguna manera. La versión 3 es una reescritura completa de la base de código original, que incorpora un número significativo de mejoras a la toma de huellas dactilares a nivel de red e introduce la capacidad de razonar sobre cargas útiles a nivel de aplicación (por ejemplo, HTTP). [85] Algunas de las capacidades de p0f son [86]:

- Identificación altamente escalable y extremadamente rápida del sistema operativo y el software en ambos puntos de una conexión TCP básica, especialmente en configuraciones donde las sondas NMap están bloqueadas, son demasiado lentas, no confiables o simplemente activarían alarmas.
- Medición del tiempo de actividad del sistema y de la conexión a la red, la distancia (incluida la topología detrás de NAT o filtros de paquetes), las preferencias de idioma del usuario, etc.
- Detección automatizada de configuraciones de conexión compartida / NAT, equilibrio de carga y proxy a nivel de aplicación.
- Detección de clientes y servidores que falsifican declaraciones declarativas como X-Mailer o User-Agent.

T-Pot implementa la p0f en un contendor Docker con modo de red “Host” y mapea el volumen “/var/log/p0f” al directorio “/data /p0f/log” del sistema operativo anfitrión.

3.2.3.3. Suricata

Suricata es un motor de detección de amenazas de red, maduro, rápido, robusto, gratuito y de código abierto. El motor Suricata es empleado para la detección de intrusiones en tiempo real (IDS), prevención de intrusiones en línea (IPS), monitoreo de seguridad de red (NSM) y procesamiento pcap fuera de línea. Suricata inspecciona el tráfico de la red utilizando un poderoso y extenso lenguaje de firmas y reglas, además de tener

un poderoso soporte de scripting Lua para la detección de amenazas complejas. Con formatos de entrada y salida estándar como YAML y JSON, las integraciones con herramientas SIEM existentes como Splunk, Logstash / Elasticsearch, Kibana y otras bases de datos se vuelven fáciles. [87]

La implementación de T-Pot consiste en un contendor Docker con modo de red “Host” y el volumen “/var/log/suricata” está mapeado al directorio “/data/suricata/log”.

3.2.4. HERRAMIENTAS

Adicionalmente, T-Pot incorpora las siguientes herramientas que coadyuvan en el análisis, monitoreo y administración de la solución de *honeypot*, así como en la etapa de envío de hallazgos a la comunidad.

3.2.4.1. Cockpit

Cockpit es una herramienta de administración de servidores que proporciona una consola web fácil de usar, a través de la cual se puede administrar de manera sencilla uno o varios sistemas Linux. La herramienta permite ver aspectos del rendimiento del sistema y realizar cambios de configuración; si bien la lista de tareas puede depender del tipo particular de Linux que esté utilizando, algunas de las actividades principales que se puede realizar a través de Cockpit son las siguientes: supervisión de la actividad del sistema como ser CPU, memoria, almacenamiento, red, cuentas de usuario, servicios, aplicaciones, actualizaciones de software, uso de línea de comandos, volcado de kernel, configuraciones de seguridad, administración de máquinas virtuales, contenedores, etc. [88] [89] [90]

La configuración por defecto de T-Pot publica Cockpit en el puerto 64294 (en lugar del 9090), configuración que es realizada durante la la instalación a través del archivo “/etc/systemd/system/cockpit.socket.d/ listen.conf”. Es importante mencionar que no permite el acceso con el usuario root, el cual es necesario para realizar cambios en configuraciones y administración de contenedores, por lo cual (si se desea hacer uso de la herramienta) se debe editar el archivo “/etc/pam.d/Cockpit”, comentar la línea “auth requisite pam_succeed_if.so uid >= 1000” o modificarla por “auth requisite pam_succeed_if.so uid >= 0”.

3.2.4.2. CyberChef

CyberChef es una aplicación web sencilla e intuitiva para realizar todo tipo de operaciones "cibernéticas" dentro de un navegador web. Estas operaciones incluyen codificación simple como XOR o Base64, encriptación más compleja como AES, DES y Blowfish, creación de binarios y hexdumps, compresión y descompresión de datos, cálculo de hashes y sumas de verificación, análisis de IPv6 y X.509, cambio de codificaciones de caracteres y mucho más. La herramienta está diseñada para permitir a los analistas técnicos y no técnicos manipular datos de formas complejas sin tener que lidiar con herramientas o algoritmos complejos. [91]

T-Pot implementa CyberChef a través de la ejecución de un contendor Docker, el cual publica la herramienta en el puerto 64299 y Heimdall Application Dashboard (panel web que se detallará en un subtítulo posterior) haciendo uso de Nginx como servidor web, trabaja como proxy reverso redireccionando el puerto 64299 al puerto 64297 y a la ruta “/cyberchef”, es decir “https://direccion.ip.del.honeypot:64297/cyberchef/”.

3.2.4.3. SpiderFoot

SpiderFoot es una herramienta de reconocimiento que consulta automáticamente más de 100 fuentes de datos públicas (OSINT) para recopilar información sobre direcciones IP, nombres de dominio, direcciones de correo electrónico, nombres y más. La solución se puede utilizar de forma ofensiva (por ejemplo, en un ejercicio del equipo rojo o en una prueba de penetración) para el reconocimiento de su objetivo o de forma defensiva para recopilar información sobre lo que una persona u organización podrían haber expuesto a través de Internet. Para usarlo, simplemente se debe especificar el objetivo a investigar, elegir qué módulos habilitar y SpiderFoot recopilará datos para comprender todas las entidades y cómo se relacionan entre sí. [92] [93]

La implementación en T-Pot se la hace a través de un contendor Docker, publicando SpiderFoot en el puerto 64303. Heimdall Application Dashboard haciendo uso de Nginx como proxy reverso, redirecciona el puerto

64303 al 64297 y a la ruta “/spiderfoot”, por lo que para acceder a SpiderFoot se lo hace a través de “https://direccion.ip.del.honeypot:64297/spiderfoot/”.

3.2.4.4. EWS Poster

EWS Poster permite recopilar registros y alertas de diferentes honeypots, convertirlas al protocolo EWS y enviarlas a un recolector externo. [94] T-Pot hace uso de ews.py (herramienta desarrollada en Python) [95] implementándola en un contenedor Docker que mapea el directorio “/data”, donde se encuentran todos los registros de los diferentes *honeypots* y envía todos los datos capturados a un backend de la comunidad, que posteriormente son utilizados para alimentar a Sicherheitstacho (Security Dashboard de Telekom). [16]

3.2.4.5. Heimdall Application Dashboard y Nginx

Heimdall Application Dashboard es un panel que permite crear enlaces a diferentes aplicaciones web. Se caracteriza por su sencillez y elegancia, además de ser apoyado y respaldado por el grupo linuxserver.io. [96]

T-Pot implementa Heimdall como Dashboard de acceso a Cockpit, Cyberchef, Elasticsearch Head, Kibana, Spyderfoot, GitHub y SecurityMeter. Lo hace empleando Nginx como servidor web, el cual trabaja como proxy reverso y accede a los diferentes puertos de cada contenedor que alberga las herramientas previamente mencionadas. Se accede al servicio a través del puerto 64297, publicado por un contenedor Docker que lleva por nombre “Nginx” y mapea los volúmenes “/etc/nginx/cert/”, “/etc/nginx/nginxpasswd” y “/var/log/nginx/” a los directorios “/data/nginx/cert/”, “/data/nginx/conf/nginxpasswd” y “/data/nginx/log/” respectivamente.

3.3. ANÁLISIS DEL FUNCIONAMIENTO

Una vez examinados los componentes de T-Pot por separado, se procede a estudiar cómo funcionan todos ellos de manera conjunta, para ello se divide el análisis en tres etapas: instalación, preparación de componentes y ejecución de T-Pot, para finalmente estudiar el funcionamiento sinérgico de la solución.

3.3.1. INSTALACIÓN

En esta primera etapa se analizan las tareas realizadas en el proceso de instalación de T-Pot.

Para un mejor entendimiento, primero se detallarán brevemente los requisitos del sistema, los modos y tipos de instalación, para posteriormente entrar al análisis de las tareas realizadas por el instalador.

3.3.1.1. Conceptos previos a la instalación

La instalación de T-Pot es bastante sencilla, lo único que requiere es conexión a Internet de manera transparente (sin uso de Proxy). Pero previo a ello, el usuario debe tener claro dónde, de qué modo y tipo se realizará la instalación.

En lo que refiere a dónde, T-Pot puede ser instalado en un hardware real, en una máquina virtual (probado en Virtual Box y VMWare) o en la nube (como Amazon Web Services, Google Cloud y Open Telekom Cloud con o sin la ayuda de Ansible o Terraform). En cualquiera de los casos, requiere al menos 4vCPUs, 8GB de RAM y 128 GB de espacio libre en disco duro.

Respecto al modo de instalación, este puede ser a través de una imagen ISO (ya sea prediseñada o creada por el usuario) o como software en un sistema operativo Debian 10 (Buster) ya existente.

Finalmente, existen 6 tipos de instalación para seleccionar: *Standard, Sensor, Industrial, Collector, NextGen* y *Medical.* La diferencia entre cada una de ellas son los *honeypots* y herramientas que se instalan, el detalle de cada uno puede ser revisado en el Anexo “B”.

Por su perte, el Anexo “C” muestra la instalación de T-Pot y sugerencias para la solución de problemas que podrían presentarse.

3.3.1.2. Análisis de la instalación

Independientemente del modo y tipo de instalación, T-Pot ejecuta las sentencias que se encuentran en el script “install.sh”, las cuales en términos generales realizan acciones agrupadas en seis fases.

- De manera previa a ejecutar las fases, verifica si existe el archivo “install.log” en el directorio raíz “/”. En caso que exista, aborta la instalación debido a que el instalador solo puede ser ejecutado una vez.
- En la fase I se define todas las variables globales a utilizar durante la ejecución del script. Algunas de ellas son las URL de Debian, Python, GitHub, Docker y listbot de Deutsche Telekom, la ruta a diferentes archivos de configuración, los paquetes a verificar e instalar, los mensajes que se mostrarán al usuario, etc.
- En la fase II se define todas las funciones a utilizar durante la ejecución del script. Las mismas ayudan en la generación de banners, generación de nombres de usuario, verificación de logueo con usuario root, paquetes instalados, acceso a sitios remotos, servicios e instalación de dependencias.
- En la fase III, denominada de Pre Instalación, se verifica que el usuario logueado es root y que se tienen instalados ciertos paquetes definidos en una variable global en la fase uno.
- La fase IV se denomina “Preparación del entorno de instalación”y se verifica la versión de Debian, la cual debe ser Stretch (9) o Buster (10) estable, caso contrario detendrá la instalación. Verifica también los parámetros que se pasaron para la ejecución del script, los cuales pueden ser --conf y el archivo para la configuración sin interacción humana; o --type y el tipo de instalación ya sea este “user”, “auto” o “iso”. Finalmente, en caso que el tipo de instalación sea “user”, alerta al usuario que los puertos conocidos (FTP, SSH, Telnet, SMTP, HTTP, etc) serán usados por los *honeypots.*
- La fase V lleva por título “Interacción entre el usuario y el instalador”, se establecen los parámetros de configuración del proxy (solo en caso que el tipo de instalación sea “iso”); se solicita el tipo de instalación (*Standard, Sensor, Industrial, Collector, NextGen* o *Medical*); se pide introducir una contraseña para el usuario “tsec” (solo en caso que el tipo de instalación sea “iso”), además del usuario y contraseña para la interfaz web.

- Finalmente en la fase VI, si no se produjo ningún error en todas las fases previas, se procede con la instalación de T-Pot, donde se realizan las siguientes tareas:
    - Se crea en la raíz del sistema operativo los archivos “install.err” e “install.log”, donde irán los errores y logs producidos en la instalación.
    - Se muestra un banner con el texto “Installing ...” y se verifica las dependencias.
    - Si todo es correcto y el tipo de instalación es diferente a “*Sensor”,* se configura los credenciales introducidos para la interfaz web y se genera un certificado SSL auto firmado.
    - En caso de ser necesario, se configura el servidor NTP, el protocolo 802.1x y las interfaces inalámbricas.
    - Se desactiva el roaming SSH, se instala Elasticdump y Elasticsearch-curator.
    - Se clona en el directorio /opt/tpot el código albergado en “https://github.com/telekom-security/tpotce” (por lo que siempre se instalará la última versión estable)
    - Se crea el usuario y grupo tpot con gid y uid 2000, sin directorio home y se deshabilita el login.
    - Se establece el *hostname*, para ello se selecciona una palabra al azar del archivo a.txt y otra del archivo n.txt ubicados en el directorio /opt/tpot/host/usr/share/dict/ y se concatenan ambas; esto ocurre siempre y cuando la instalación no sea en entornos *cloud*, en ese caso se mantiene el *hostname* original.
    - Se establece los puertos 64294 y 64295 para los servicios Cockpit y SSH respectivamente, además de impedir el acceso a Cockpit con el usuario root.
    - Se crea un enlace simbólico del tipo de instalación seleccionada “/opt/tpot/etc/compose/tipo_instalacion.yml” en la ruta “/opt/tpot/etc/tpot.yml”, esto con el objetivo de descargar e iniciar solo aquellas imágenes de Docker que correspondan al tipo de instalación requerida (*Standard, Sensor, Industrial, Collector, NextGen* o *Medical*); por lo tanto, del archivo “tpot.yml” se

selecciona el nombre de cada una de las imágenes y se las descarga de Docker Hub a través del comando “docker pull”.

- Se modifica el archivo “/etc/apt/apt.conf.d/10periodic” para actualizar la lista de paquetes diariamente y eliminar los paquetes obsoletos semanalmente. Además, se edita los siguientes archivos: “/etc/sysctl.conf” para que el sistema se reinicie tras un “*kernel panic”;* “/etc/fail2ban/jail.d/tpot.conf” para configurar fail2ban y evitar ataques de fuerza bruta en Cockpit, ssh y administración web; “/etc/systemd/network/99-default.link” para corregir el error detallado en “https://github.com/systemd/systemd/issues/3374”; “/etc/crontab” para ejecutar trabajos en cron que permita comprobar diariamente si hay imágenes Docker actualizadas y descargarlas, eliminar diariamente todos los índices de Elasticsearch y Logstash con tiempo de creación mayor a 90 días, cargar cada hora en una ruta específica aquellos archivos que se descargaron vía ftp con Dionaea, reiniciar de lunes a sábado el servidor y buscar paquetes actualizados todos los domingos, actualizarlos y reiniciar el servidor.
- Se crea todos los directorios dentro de “/data”, donde se guardarán los archivos de cada uno de los *honeypots* y herramientas implementadas como contenedores.
- Se copia el archivo “tpot.service” en el directorio “/etc/systemd/system/” y ejecuta “systemctl enable tpot” con el objetivo de que el servicio tpot se inicie cada vez que levanta el sistema.
- Se verifica y configura permisos a directorios y archivos, así como otros parámetros de grub, console-setup, prompt, ip y agregación del directorio “/opt/tpot/bin” al path, de modo tal que se pueda acceder a los archivos desde cualquier ruta.
- Se ejecuta el script “updateip.sh”, limpia y elimina directorios y archivos que ya no son necesarios para finalmente reiniciar el sistema operativo.

La siguiente figura plasma las fases que sigue el instalador de T-Pot.

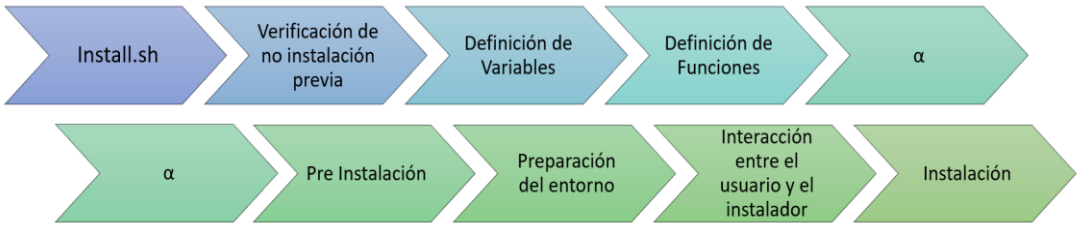


Figura 4: Fases del Instalador de T-POT
Fuente: Elaboración propia

3.3.2. PREPARACIÓN DE COMPONENTES Y EJECUCIÓN DE T-POT

Una vez finalizada la instalación, después de todo reinicio del host o reinicio particular del servicio “tpot”, se realiza una serie de tareas que consisten en la preparación de todos los componentes y ejecución de T-Pot como tal. En este apartado se analizará la definición del servicio y las tareas ejecutadas al iniciar y detener el mismo.

Al revisar el archivo que controla el servicio denominado “tpot” y que se encuentra en la ruta “/etc/systemd/system/tpot.service” (Ver Anexo D) se evidencia que primero se definen las opciones genéricas (dentro de “[Unit]”) estableciendo “tpot” como la descripción del servicio, se establece que el servicio “docker.service” es un requisito y debió haber sido ejecutado previamente. Posteriormente define las opciones específicas para el tipo de extensión (dentro de “[Service]”) estableciendo que el servicio se reiniciará cuando el proceso de servicio finalice, se elimine o agote el tiempo de espera (Restart=always); así mismo, que debe esperar 5 segundos antes de reiniciar el servicio (RestartSec=5) y el “TimeoutStartSec” y “TimeoutStopSec” no tendrá un límite de tiempo (“TimeoutSec=infinity”). Tras definir los valores, ejecuta las siguientes tareas antes de iniciar el servicio (“ExecStartPre”):

- Obtiene las direcciones IP interna y externa para añadirlas a /etc/issue y /data/ews/conf/ews.ip respectivamente, esto a través de la ejecución del script “/opt/tpot/bin/updateip.sh”.
- Ejecuta el script “/opt/tpot/bin/clean.sh” que permite limpiar el estado del directorio “/data” si la persistencia esté deshabilitada o rotar y comprimir los *logs* que allí se encuentren en caso de estar habilitada.
- Elimina contenedores, imágenes y volúmenes antiguos.

- Habilita el modo promiscuo en p0f y Suricata.
- Configura *iptables* para aceptar las conexiones entrantes hacia los puertos de cada *honeypot* y evitar que sean reenviadas a *glutton* o *honeytrap*. Para ello ejecuta el script “/opt/tpot/bin/rules.sh” pasando como parámetros el archivo “/opt/tpot/etc/tpot.yml” (que es un enlace simbólico al archivo que contiene las configuraciones e imágenes de Docker a levantar según el tipo de instalación inicial) y la palabra “set” (que le indica al script que debe añadir las reglas a *iptables*).

Una vez terminadas las tareas descritas previamente, se ejecuta el servicio (“ExecStart”) y haciendo uso de docker-compose levanta los contenedores descritos en “/opt/tpot/etc/tpot.yml”, finalizando de ese modo la ejecución del servcio *“tpot”*.

Como es común en la definición de todo servicio, también se establecen las tareas a ejecutar cuando el servicio es detenido. En el caso de “/etc/systemd/system/tpot.service” se ejecuta docker-compose para detener y borrar contenedores y volúmenes descritos en “/opt/tpot/etc/tpot.yml”. Tras detener el servicio (“ExecStopPost”) se ejecuta el script “/opt/tpot/bin/rules.sh” pasando los parámetros “/opt/tpot/etc/tpot.yml” y “unset” (que le indica al script que debe borrar las reglas de iptables creadas al iniciar el servicio).

3.3.3. FUNCIONAMIENTO SINÉRGICO DE T-POT

Una vez iniciado el servicio “tpot”, todos los componentes y contenedores estarán operando para que la solución reciba ataques, los envíe a la comunidad y el administrador del *honeypot* pueda analizar todos los eventos. En este apartado se analiza el funcionamiento sinérgico de cada componente para hacer que T-Pot funcione como un todo.

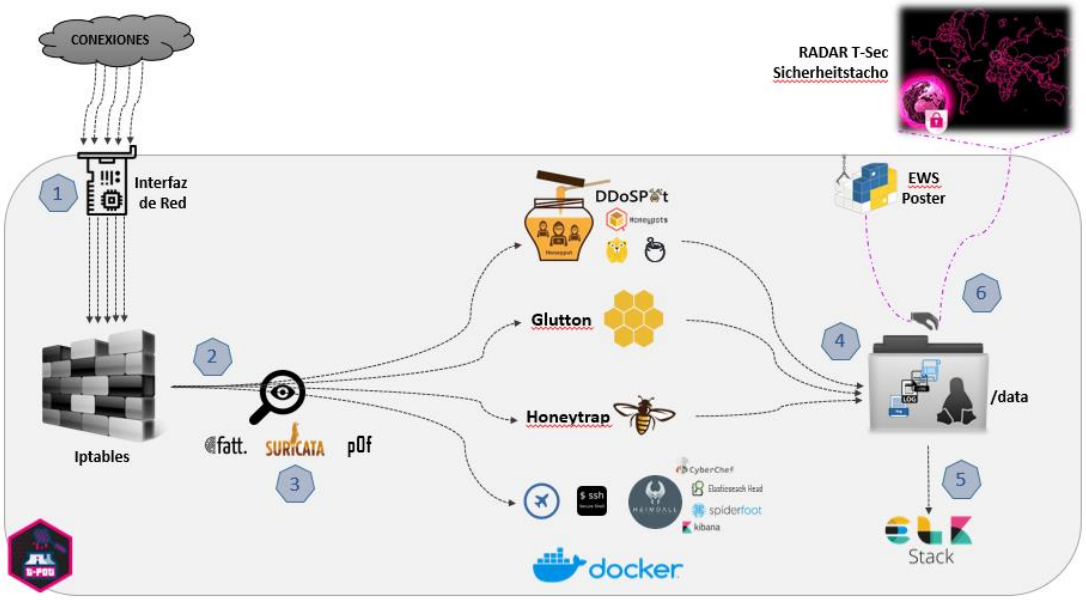


Figura 5: Funcionamiento Sinérgico de T-POT
Fuente: Elaboración propia

La figura 5 muestra de manera gráfica el flujo que siguen los datos, que para un entendimiento preciso puede ser dividido en seis etapas:

1. Todos los datos que ingresan a T-Pot a través de la interfaz de red del host son enviados al módulo del *kernel* de linux "*iptables".*
2. En base a las reglas configuradas al momento de iniciar el servicio según el tipo de instalación, el tráfico será enviado a uno de los siguientes contenedores:
   2.1. Al *honeypot* que escucha en el puerto y protocolo consultado.
   2.2. En caso de que el protocolo y puerto consultado no esté siendo atendido por ningún *honeypot*, la solicitud se enviará a Glutton si el tipo de instalación es NextGen o a Honeytrap si la instalación es Standard, Sensor, Industrial o Collector.
   2.3. Los puertos 64294, 64295 y 64297 están destinados a la administración, correspondiendo a cockpit, ssh y Heimdall respectivamente.
3. El tráfico reenviado es inspeccionado por los siguientes NSM:
   3.1. Fatt. Inspecciona todo el tráfico, excepto el que va dirigido al puerto 64294.
   3.2. Suricata. Inspecciona todo el tráfico.
   3.3. p0f. Inspecciona todo el tráfico.

4. Toda interacción que se tiene entre los *honeypots* y el atacante es *logeada* de manera persistente en el directorio "/data" del sistema anfitrión y permanece allí por 30 días, valor que puede ser cambiado por el administrador editando el archivo "/opt/tpot/etc/logrotate/logrotate.conf".
5. Los datos almacenados en "/data" son procesados por ELK Stack, permitiendo su análisis e interpretación al administrador.
6. Los datos capturados son enviados a un *backend* de la comunidad, el cual utiliza los datos para alimentar a Sicherheitstacho (el radar T-Sec que muestra los ataques cibernéticos que ocurren alrededor del mundo). Este aporte a la comunidad está activado de forma predeterminada, pero el administrador puede optar por no participar en el envío de datos, para lo cual debe eliminar todo lo relacionado con EWSposter del archivo "/opt/tpot/etc/tpot.yml".

3.4. ADMINISTRACIÓN, USO DE HERRAMIENTAS Y REVISIÓN DE ATAQUES

Una vez ejecutado el servicio, se tienen los siguientes métodos de interactuar con T-Pot para poder administrarlo, hacer uno de herramientas y revisar los ataques que recibe.

- La primera es a través de Cockpit, accediendo al puerto 64294 desde un navegador web.
- La segunda es a través de una conexión SSH a través del puerto 64295.
- La tercera es empleada para el análisis de los datos obtenidos por los *honeypots*, accediendo al puerto 64297 vía web. A través de ese puerto se accede a Heimdall y se selecciona Kibana u otra herramienta de apoyo como Cyberchef, Spyderfoot o Elasticsearch Head.

4. PLANTEAMIENTO DE MEJORAS A T-POT

Como se evidencia en el análisis de T-POT, se trata de un honeypot bastante completo, funcional y con mejoras permanentes. Pero a pesar de ello se identificaron pequeñas observaciones que se pueden corregir o aportes para mejorar la solución. Se menciona a continuación algunas de ellas.

La documentación oficial, publicada en el archivo README.md de GitHub [16], no se encuentra actualizada con las características de la última versión de T-Pot a la fecha (20.06.2 publicada el 22 de febrero de 2021). El diagrama de la arquitectura no está actualizado con los honeypots añadidos recientemente (DDOSpot, EndlesSSH, HellPot, Honeypots, Log4pot, RedisHoneypot); sumado a esto, los honeypots que se ejecutan en cada tipo de instalación según la documentación (Standard, Sensor, Industrial, Medical, Collector y NextGen) no corresponden a los que realmente se ejecutan en la implementación de T-Pot.

Si el atacante es una persona (no un *bot*) o se trata de un ataque dirigido a la empresa, es muy probable que no solo se ejecuten tareas automatizadas sino que también se hará una revisión manual de los servicios, por lo que sería interesante desarrollar módulos a través de los cuales los usuarios puedan subir datos que caractericen a su empresa. Así por ejemplo, en lugar de usar sitios web predefinidos en algún CMS (WordPress, Drupal, OwnCloud, GitLab, etc) como sucede actualmente, se podría usar la opción que tiene *Snare* para clonar el sitio web oficial de la empresa que implementa T-Pot (haciendo uso del comando “clone --target http://www.sitio-de-la-empresa.com”. Así mismo, se podría analizar la implementación de servicios de *Data Sanitization* para poder cargar datos reales sin poner en riesgo la información de la empresa o de los clientes.

Si bien los *honeypots* implementados cubren gran cantidad de servicios y todas las solicitudes de conexión a los puertos TCP y UDP son respondidos por algún *honeypot, Glutton o Honeytrap*; se sugiere revisar al menos los siguientes *honeypots* orientados a bases de datos y sistemas web, tecnologías ampliamente usadas y atacadas en la actualidad. MongoDB-HoneyProxy, NoSQLpot, MysqlPot, pghoney, Express honeypot, Laravel Application Honeypot, Nodepot, phpmyadmin_honeypot, tomcat-manager-honeypot y revisar [97] para otros *honeypots* de código abierto.

En muchas ocasiones una empresa no requiere que sus sitios o sistemas sean accedidos desde cualquier lugar del mundo, pero reciben solicitudes de conexión desde países como Rusia, India o China; si bien esto se lo puede detectar a través del panel de *Kibana*, se recomienda analizar la

implementación de un panel similar al *Security Dashboard* [14] que permita ver de manera gráfica de qué países se originan los intentos de conexión al T-Pot implementado en la empresa.

CONCLUSIONES

Como se evidencia a lo largo del documento, la arquitectura y lógica en el uso de tecnologías que implementa T-Pot permiten hacer de él una solución escalable, a la cual se pueden agregar otros *honeypots* sin afectar el funcionamiento de sus componentes. El hecho de emplear contenedores, crea ambientes aislados y seguros, impidiendo que los mismos sean usados como herramientas de ataque al ser comprometidos. Está claro que T-POT es una herramienta muy bien diseñada y un caso de estudio muy interesante de analizar y copiar para la implementación de otras soluciones. Así mismo, el proyecto como tal es todo un éxito y la comunidad encargada de su desarrollo es bastante activa, implementando mejoras constantemente para el uso de cualquier persona interesada; tal vez lo único que se pueda mejorar es la actualización de la documentación, pero al ser un proyecto tan dinámico, es entendible que no siempre esté a la par del desarrollo. Finalmente, al ser una herramienta tan fácil de implementar, recomiendo su uso en entornos laborales, permitiendo así tener conocimiento de los intentos de conexiones y ataques que se tienen, lo cual puede ser útil al momento de justificar inversión en tecnología y seguridad; además, el hecho se ser una solución Open Source, permite estudiar el código para agregar mejoras o personalizaciones que requiera la empresa o incluso como base para otros proyectos.

GLOSARIO

Antivirus : Programa cuyo objetivo es buscar, detectar, bloquear, desinfectar archivos y prevenir una infección de virus informáticos,

APT : Amenaza Persistente Avanzada, por sus siglas en inglés Advanced Persistent ThreaT.

Botnet : Nombre genérico que denomina a cualquier grupo de PC infectados y controlados por un atacante de forma remota.

DDoS : Ataque de denegación de servicio distribuido, por sus siglas en inglés Distributed Denial of Service.

Firewall : También denominado cortafuegos, es la parte de un sistema o una red informática que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

FQDN : Fully Qualified Domain Name, es un nombre de dominio completo que incluye el nombre de la computadora y el nombre de dominio asociado a ese equipo.

Gusano : Programa de software malicioso que puede replicarse a sí mismo en ordenadores o a través de redes.

Hardening de servidores : Proceso de asegurar un sistema mediante la reducción de vulnerabilidades en el mismo, esto se logra eliminando software, servicios, usuarios, etc; innecesarios en el sistema; así como cerrando puertos que tampoco estén en uso.

Host virtual : Mecanismo que permite mantener múltiples nombres de host en un solo servidor.

IDS : Sistema de Detección de Intrusos (Intrusion Detection System). Programa de detección de accesos no autorizados a un computador o a una red.

IPS : Sistema de Prevención de Intrusos (Intrusion Prevention System). Programa que ejerce el control de acceso en una red informática para proteger a los sistemas computacionales de ataques y abusos.

Iptables : Módulo del kernel de Linux que se encarga de filtrar los paquetes de red en base a las tablas de filtrado, de NAT y la tabla Mangle.

OSINT : Inteligencia de código abierto (Open Source Intelligence por sus siglas en inglés) son datos disponibles en el dominio público que pueden revelar información interesante sobre un objetivo. Esto incluye DNS, Whois, páginas web, DNS pasivo, listas negras de spam, metadatos de archivos, listas de inteligencia de amenazas, así como servicios como SHODAN, HaveIBeenPwned? y más.

OTP : Contraseña de un solo uso, por sus siglas en inglés One-Time Password.

Proxy : Servidor (programa o dispositivo) que hace de intermediario en las peticiones de recursos que realiza un cliente (A) a un servidor (C).

Prueba de concepto : Implementación, a menudo resumida o incompleta, de un método o de una idea, realizada con el propósito de verificar que el concepto o teoría en cuestión es susceptible de ser explotada de una manera útil.

Script : Archivo contiene una secuencia de comandos e instrucciones a ser ejecutadas por una coputadora.

Script Kiddie : Individuo no calificado que utiliza scripts o programas desarrollados por otros para atacar sistemas informáticos y redes.

SIEM : Gestión de Información y Eventos de Seguridad (Security Information and Event Management). Categoría de software que tiene como objetivo otorgar a las organizaciones información útil sobre potenciales amenazas de seguridad de sus redes críticas de negocio, a través de la estandarización de datos y priorización de amenazas.

Terminal : Software o dispositivo electrónico que se emplea para interactuar con un computador.

UTM : Gestión Unificada de Amenazas (Unified Threat Management). Dispositivo de red único con múltiples funciones, entre ellas: antivirus, firewall (cortafuegos), IDS, IPS, NAT, VPN, antispam, antiphishing, antispyware, filtro de contenidos, etc.

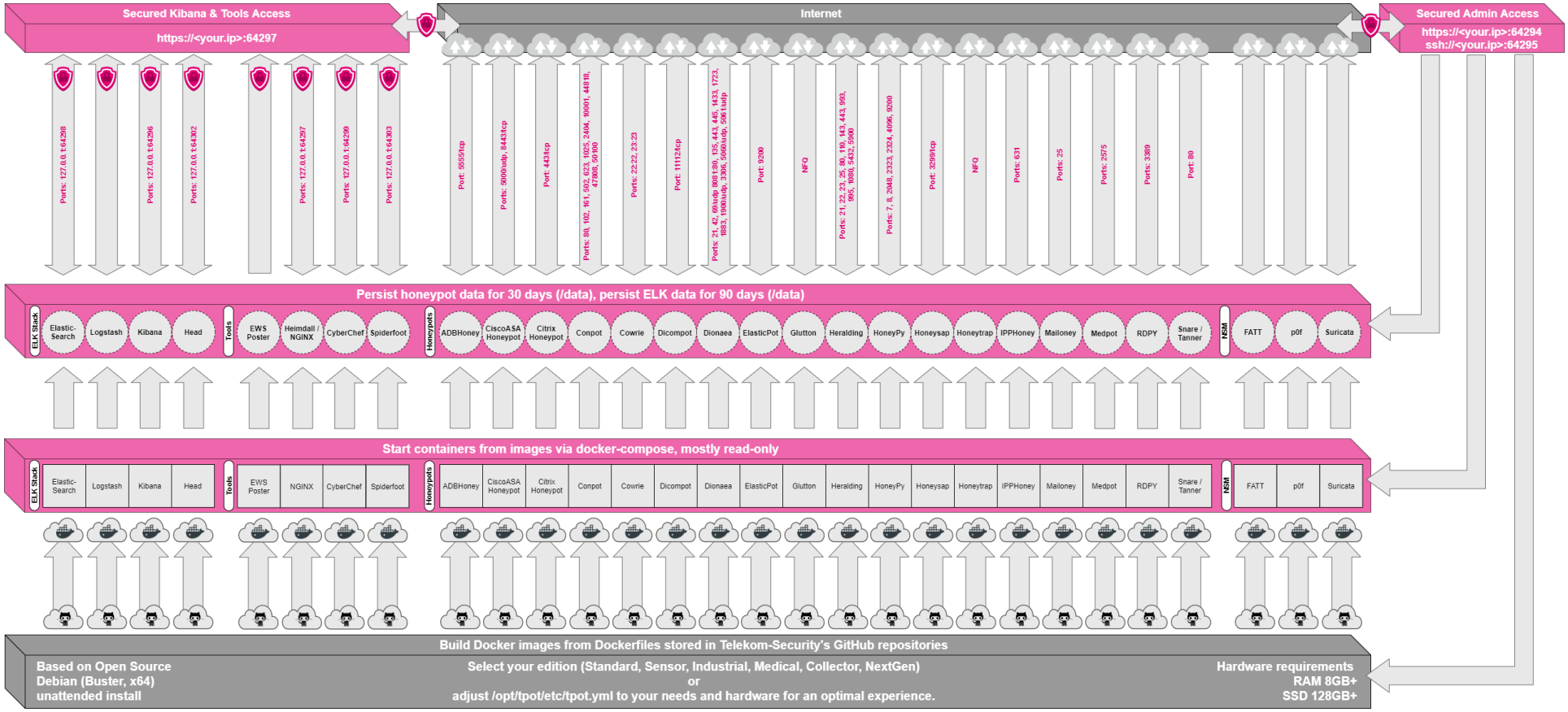
VPN : Red Privada Virtual (Virtual Private Network). Tecnología de red de ordenadores que permite una extensión segura de la red de área local (LAN) sobre una red pública o no controlada como Internet.

Vulnerabilidad : Falla que permite que una amenaza se convierta en un riesgo.

ANEXOS

ANEXO A

ARQUITECTURA DE T-POT


Secured Kibana & Tools Access
https://<your.ip>:64297
Internet
Secured Admin Access
https://<your.ip>:64294
ssh://<your.ip>:64295
Ports: 127.0.0.1:64298
Ports: 127.0.0.1:64296
Ports: 127.0.0.1:64302
Ports: 127.0.0.1:64297
Ports: 127.0.0.1:64299
Ports: 127.0.0.1:64303
Port: 5555/tcp
Ports: 9000/udp, 8443/tcp
Port: 4431/tcp
Ports: 80, 102, 161, 502, 623, 1025, 2404, 10001, 44818, 47808, 50100
Ports: 22:22, 23:23
Port: 11112/tcp
Ports: 21, 42, 69/udp, 80/tcp, 135, 443, 445, 1433, 1723, 1883, 1900/udp, 3306, 5060/udp, 5061/udp
Port: 9200
NFQ
Ports: 21, 22, 23, 25, 80, 110, 143, 443, 993, 995, 1080, 5432, 5900
Ports: 7, 9, 2048, 2323, 2324, 4096, 9200
Port: 2598/tcp
NFQ
Ports: 631
Ports: 25
Ports: 2575
Ports: 3389
Port: 80
Persist honeypot data for 30 days (/data), persist ELK data for 90 days (/data)
ELK Stack
Elastic-Search
Logstash
Kibana
Head
Tools
EWS Poster
Heimdall / NGINX
CyberChef
Spiderfoot
Honeypots
ADBHoney
CiscoASA Honeypot
Citrix Honeypot
Conpot
Cowrie
Dicompot
Dionaea
ElasticPot
Glutton
Heralding
HoneyPy
Honeysap
Honeytrap
IPPHoney
Mailoney
Medpot
RDPY
Snare / Tanner
NSM
FATT
p0f
Suricata
Start containers from images via docker-compose, mostly read-only
ELK Stack
Elastic-Search
Logstash
Kibana
Head
Tools
EWS Poster
NGINX
CyberChef
Spiderfoot
Honeypots
ADBHoney
CiscoASA Honeypot
Citrix Honeypot
Conpot
Cowrie
Dicompot
Dionaea
ElasticPot
Glutton
Heralding
HoneyPy
Honeysap
Honeytrap
IPPHoney
Mailoney
Medpot
RDPY
Snare / Tanner
NSM
FATT
p0f
Suricata
Build Docker images from Dockerfiles stored in Telekom-Security's GitHub repositories
Based on Open Source
Debian (Buster, x64)
unattended install
Select your edition (Standard, Sensor, Industrial, Medical, Collector, NextGen)
or
adjust /opt/tpot/etc/tpot.yml to your needs and hardware for an optimal experience.
Hardware requirements
RAM 8GB+
SSD 128GB+

ANEXO B

HONEYPOTS EMPLEADOS SEGÚN TIPO DE INSTALACIÓN DE T-POT

			Tipo de Instalación											
			Standard		Sensor		Industrial		Collector		NextGen		Medical	
Tipo	**Contenedor (C)**	**Puerto (P)**	**C**	**P**	**C**	**P**	**C**	**P**	**C**	**P**	**C**	**P**	**C**	**P**
HONEYPOTS	adbhoney	5555	✓	✓	✓	✓					✓	✓		
	ciscoasa	5000 (UDP)	✓	✓	✓	✓					✓	✓		
		8443		✓								✓		
	citrixhoneypot	443	✓	✓	✓	✓					✓	✓		
	conpot	69 (UDP)	✓		✓		✓	✓			✓			
		80						✓						
		102						✓						
		502						✓						
		21						✓						
		44818						✓						
		47808 (UDP)			✓		✓	✓			✓			
		161 (UDP)		✓		✓		✓				✓		
		2404		✓		✓		✓				✓		
		10001		✓		✓		✓				✓		
		623 (UDP)		✓		✓		✓				✓		
		1025		✓		✓		✓				✓		
		50100		✓		✓		✓				✓		
	cowrie	22	✓	✓	✓	✓	✓	✓						
		23		✓		✓		✓						
	ddospot	19 (UDP)									✓	✓		
		53 (UDP)										✓		
		123 (UDP)										✓		
		1900 (UDP)										✓		
	Dicompot	11112	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓

			Tipo de Instalación											
			Standard		Sensor		Industrial		Collector		NextGen		Medical	
Tipo	Contenedor (C)	Puerto (P)	C	P	C	P	C	P	C	P	C	P	C	P
	dionaea	20		✓		✓						✓		
		21		✓		✓						✓		
		42		✓		✓						✓		
		69 (UDP)		✓		✓						✓		
		81		✓		✓						✓		
		135		✓		✓						✓		
		445		✓		✓						✓		
		1433	✓	✓	✓	✓					✓	✓		
		1723		✓		✓						✓		
		1883		✓		✓						✓		
		3306		✓		✓						✓		
		5060		✓		✓						✓		
		5060 (UDP)		✓		✓						✓		
		5061		✓		✓						✓		
		27017		✓		✓						✓		
	elasticpot	9200	✓	✓	✓	✓					✓	✓		
	endlessh	22									✓	✓		
	glutton	Modo host									✓	✓		
	heralding	21								✓				
		22								✓				
		23								✓				
		25								✓				
		80	✓		✓		✓		✓	✓	✓			
		110		✓		✓				✓		✓		
		143		✓		✓				✓		✓		
		443								✓				

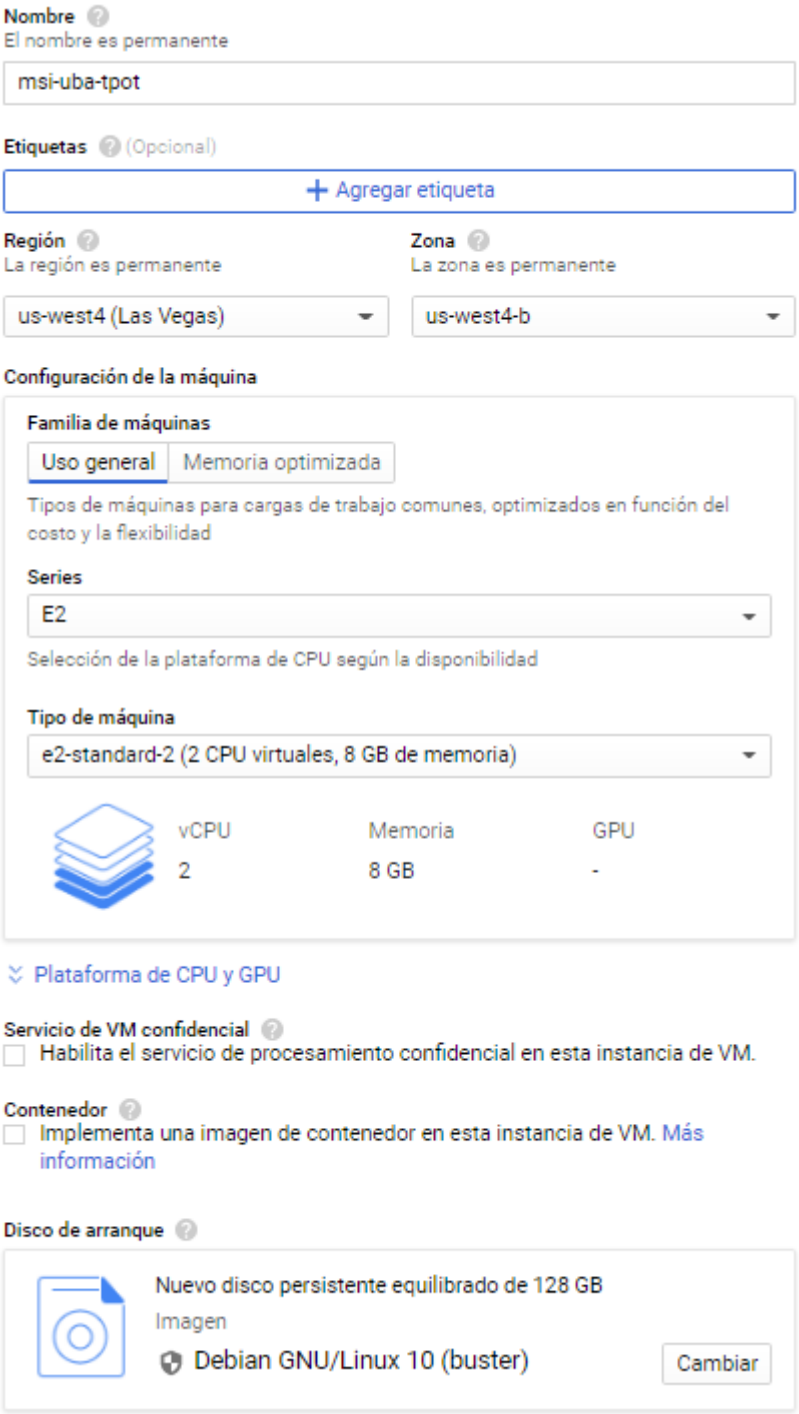
			Tipo de Instalación											
			Standard		Sensor		Industrial		Collector		NextGen		Medical	
Tipo	Contenedor (C)	Puerto (P)	C	P	C	P	C	P	C	P	C	P	C	P
		465		✓		✓				✓		✓		
		993		✓		✓				✓		✓		
		995		✓		✓				✓		✓		
		1080		✓		✓				✓		✓		
		3306								✓				
		3389								✓				
		5432		✓		✓				✓		✓		
		5900		✓		✓		✓		✓		✓		
	hellpot	80									✓	✓		
	honeypots													
	honeypy	7				✓						✓		
		8				✓						✓		
		2048				✓						✓		
		2323			✓	✓					✓	✓		
		2324				✓						✓		
		4096				✓						✓		
	honeysap	3299	✓	✓	✓	✓	✓	✓			✓	✓		
	honeytrap	Modo host	✓	✓	✓	✓	✓	✓	✓	✓				
	ipphoney	631									✓	✓		
	log4pot													
	mailoney	25	✓	✓	✓	✓					✓	✓		
	medpot	2575	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
	rdpy	3389	✓	✓	✓	✓	✓	✓			✓	✓		
	redishoneypot	6379									✓	✓		
	snare	80	✓	✓	✓	✓								
	tanner	---	✓	✓	✓	✓								

			Tipo de Instalación											
			Standard		Sensor		Industrial		Collector		NextGen		Medical	
Tipo	**Contenedor (C)**	**Puerto (P)**	**C**	**P**	**C**	**P**	**C**	**P**	**C**	**P**	**C**	**P**	**C**	**P**
NSM	Fatt	Modo host	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	p0f	Modo host	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Suricata	Modo host	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ELK STACK	Elasticsearch	64298	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
	Elasticsearch Head	64302	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
	Logstash	---	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
	Kibana	64296	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
HERRAMIENTAS	Cyberchef	64299	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
	Spiderfooot	64303	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
	EWS Poster	---	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Heimdall / NGINX	64297	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
		64304		✓				✓		✓		✓		✓

ANEXO C

INSTALACIÓN DE T-POT Y SOLUCIÓN DE EVENTUALES PROBLEMAS

La siguiente instalación se la realizará en Google Cloud Platform. Tras crear una cuenta y obtener 300$US de prueba por 90 días, se procede a crear una instancia de VM con el nombre “uba-msi-tpot” con 8vCPU, 16GB de RAM, 128 GB de disco duro y sistema Operativo Debian 10 (Buster).



A continuación en la opción Redes VPC -> Firewall del menú principal se deben aperturar todos los puertos para recibir ataques a T-Pot, además de puertos en particular para Cockpit, SSH y Web hacia Heimdall.

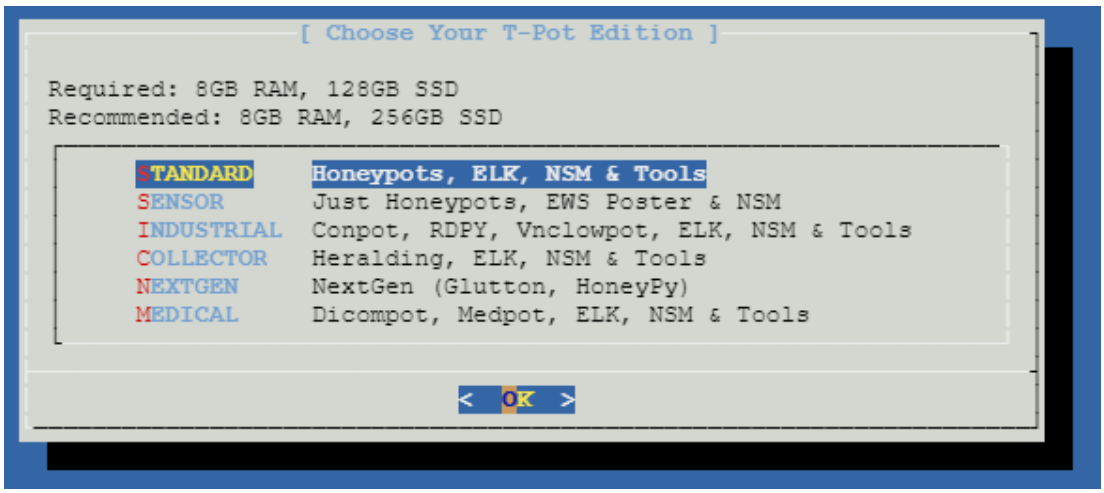
	Nombre	Tipo	Destinos	Filtros	Protocolos/puertos	Acción	Prioridad	Red ↑	Registros
☐	acceso-cockpit	Entrada	Aplicar a todas	Intervalos de IP: 0.0.0.0/0	tcp:64294	Permitir	1000	default	Desactivado
☐	acceso-ssh	Entrada	Aplicar a todas	Intervalos de IP: 0.0.0.0/0	tcp:64295	Permitir	1000	default	Desactivado
☐	acceso-web-heimdall	Entrada	Aplicar a todas	Intervalos de IP: 0.0.0.0/0	tcp:64297	Permitir	1000	default	Desactivado
☐	permitir-honeypots-inferiores	Entrada	Aplicar a todas	Intervalos de IP: 0.0.0.0/0	tcp:0-64293 udp:0-64293	Permitir	1000	default	Desactivado
☐	permitir-honeypots-superiores	Entrada	Aplicar a todas	Intervalos de IP: 0.0.0.0/0	tcp:64298-65535 udp:64298-65535	Permitir	1000	default	Desactivado

Tras ello, acceder a la instancia e iniciar con la instalación de T-Pot. Para ello se deben ejecutar los siguientes comandos:

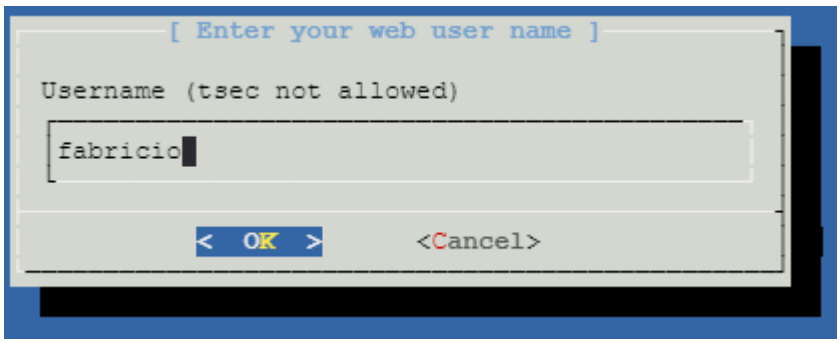
- sudo su
- apt-get update
- apt-get install git
- git clone https://github.com/telekom-security/tpotce
- cd tpotce/iso/installer/
- ./install.sh --type=user

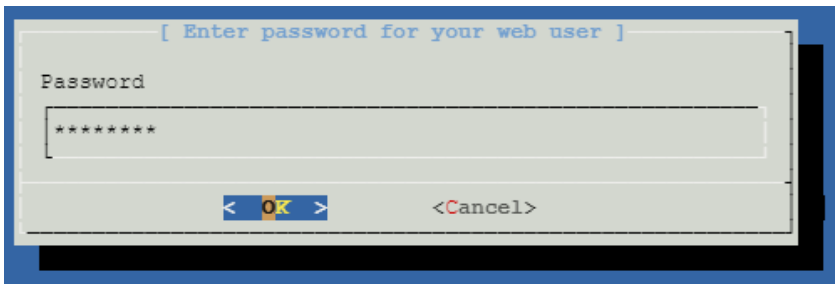
Esa acción ejecutará el instalador y dirigirá por una serie de pasos para instalar T-Pot. Pulsar “y” a la pregunta “Continue?” tras indicar que los puertos estándar serán tomados por T-Pot y se deberá acceder a administrar el servidor (ssh) a través de otros puertos.

Seleccionar el tipo de instalación “Standard”

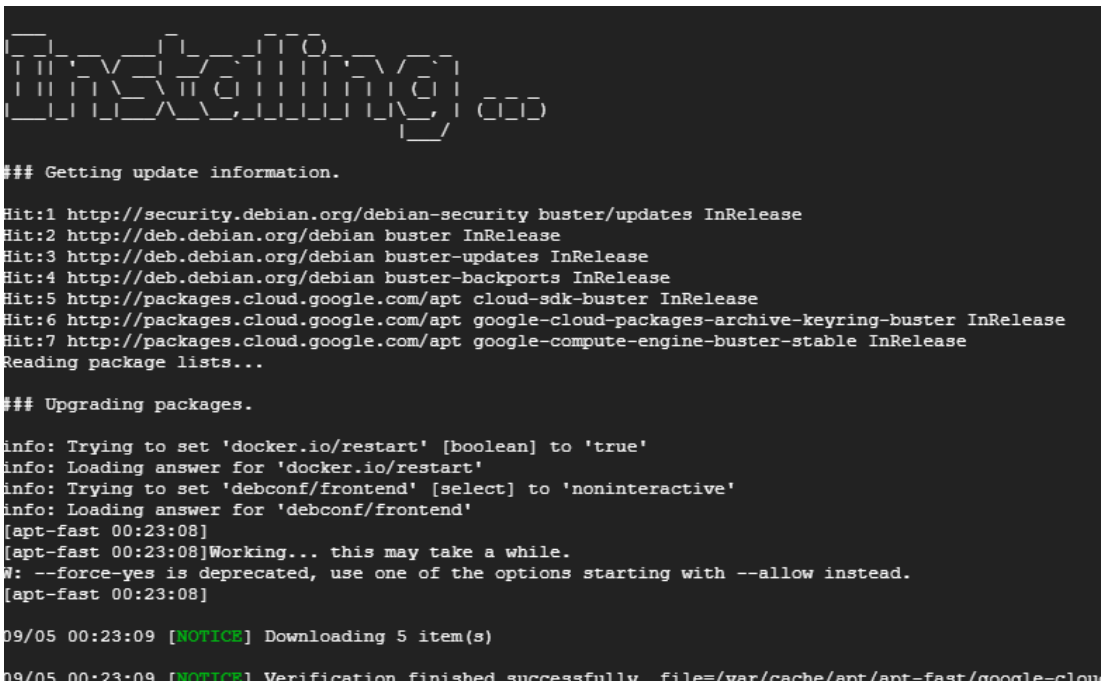


Establecer el nombre de usuario y contraseña, para el ejemplo usaremos “Fabricio”

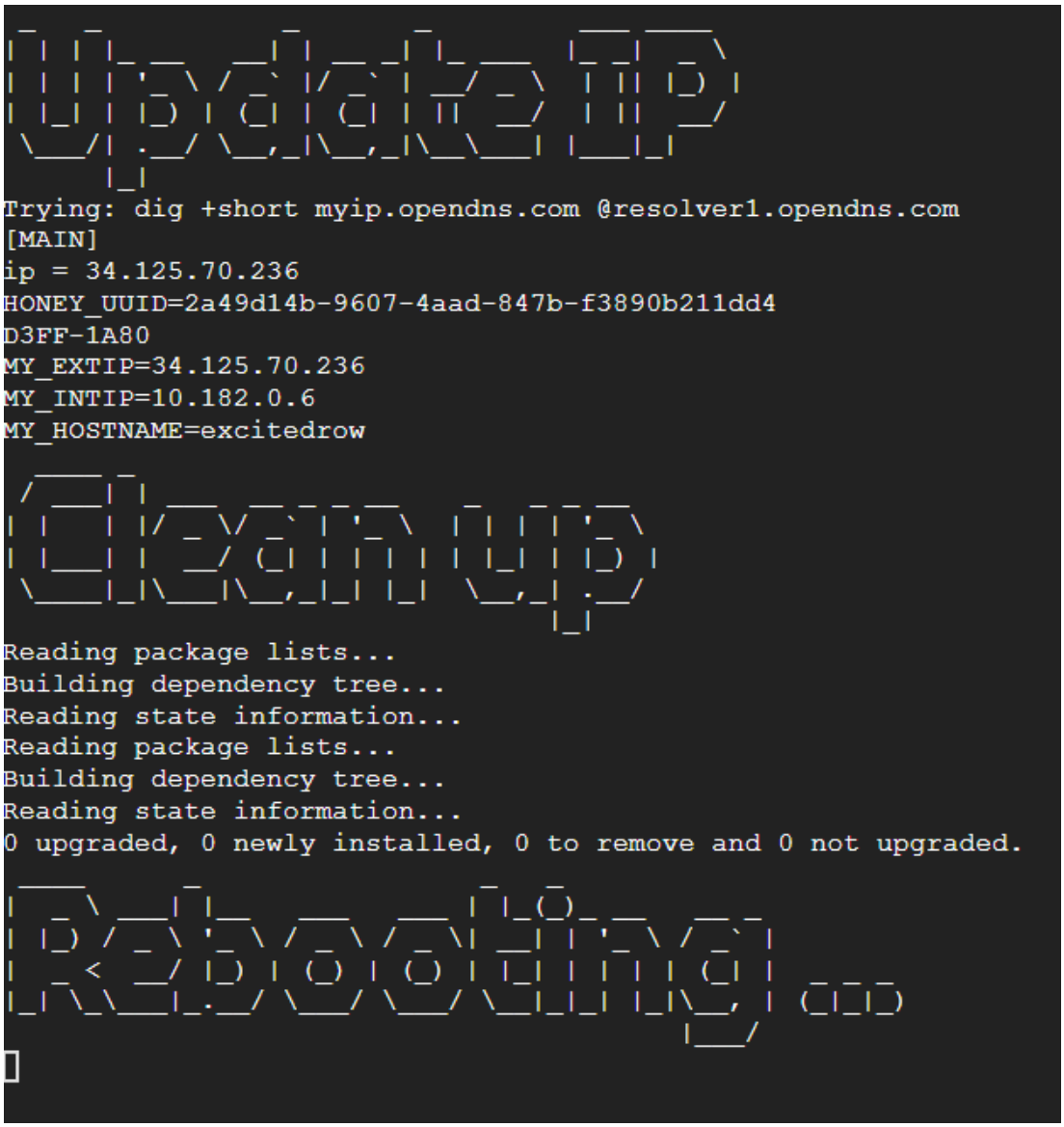




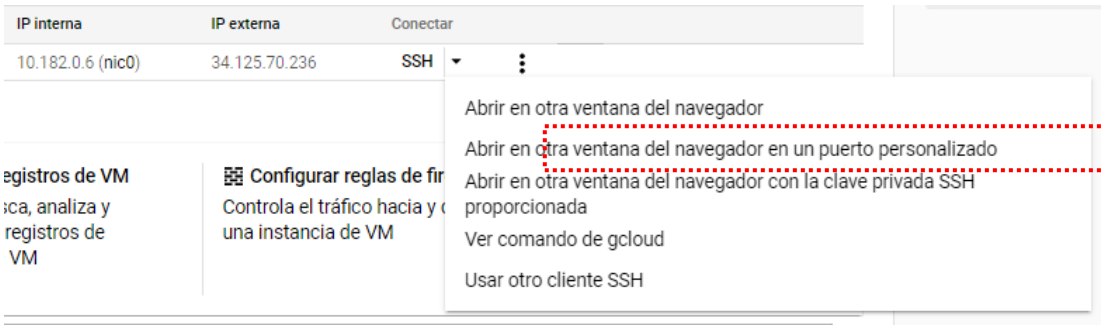
A continuación procederá con la instalación, lo cual puede llevar alrededor de 10 minutos.

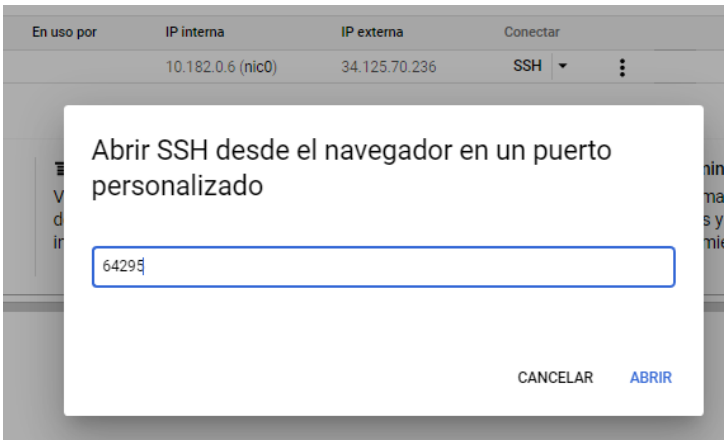


Al finalizar la instalación el servidor se reiniciará.

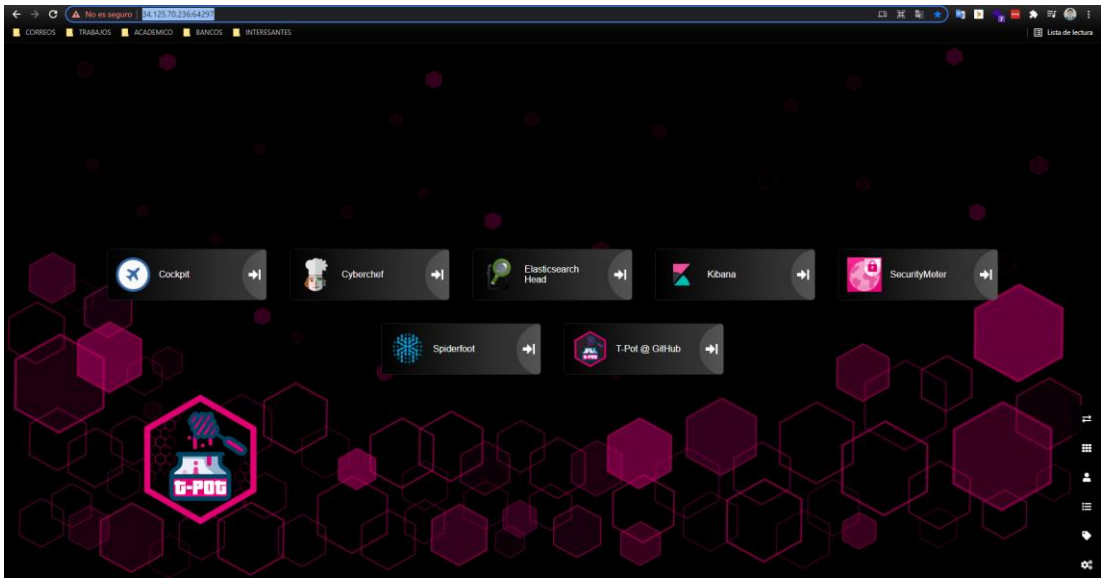


A partir de entonces el acceso ssh será a través del puerto 64295, en el caso de Google Cloud se debe seguir el siguiente procedimiento.





Tras concluir con la instalación y reiniciar el servidor, T-Pot estará listo para ser usado y se podrá acceder a la interfaz web a través del puerto 64297, en el ejemplo “https://34.125.70.236:64297”.



Si todo sale bien podrá acceder a las herramientas Cockpit, Cyberchef, Elasticsearch Head, SecurityMeter, Spyderfoot y Kibana.

Si desea acceder a la instancia utilizada para la prueba de concepto, utilice los credenciales fabriciotpot:fabriciotpot, tsec:tpot o fabricio-tpot:fabricio-tpot representados como usuario:contraseña.

SOLUCIÓN DE EVENTUALES PROBLEMAS

Como en toda solución tecnológica, es posible que surjan problemas en la instalación o uso. A continuación se detallan problemas que pueden llegar a ocurrir y cómo solucionarlos.

ERRORES EN LA INSTALACIÓN

Si durante la instalación se genera un error relacionado a “apt-fast” deberá instalar el paquete, para ello revisar https://github.com/ilikenwf/apt-fast o ejecutar los siguientes comandos.

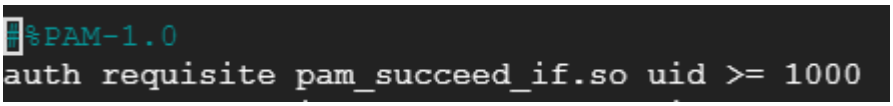
- En /etc/apt/sources.list.d/apt-fast.list, agregar
  - deb http://ppa.launchpad.net/apt-fast/stable/ubuntu bionic main
  - deb-src http://ppa.launchpad.net/apt-fast/stable/ubuntu bionic main
- apt-key adv --keyserver keyserver.ubuntu.com --recv-keys A2166B8DE8BDC3367D1901C11EE2FF37CA8DA16B
- apt-get update
- apt-get install apt-fast

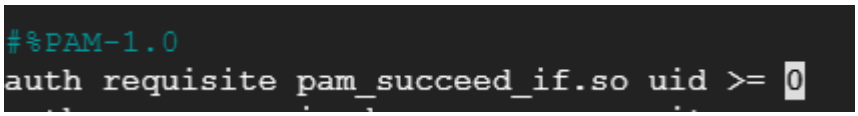
Si durante la instalación se genera el error “reboot: command not found”, agregar el alias reboot='systemctl reboot'.

Si durante la instalación se genera el error “No encuentra el comando addgroup ni adduser”, ejecutar el comando Export PATH=$PATH:/usr/sbin.

PERMITIR LOGUEO DE ROOT EN COCKPIT

Por temas de seguridad, por defecto se bloquea el acceso a la herramienta Cockpit al usuario “root”, en caso de ser necesario puede habilitarlo editando el archivo “/etc/pam.d/cockpit”, sustituyendo el parámetro “>=1000” por “>=0”.



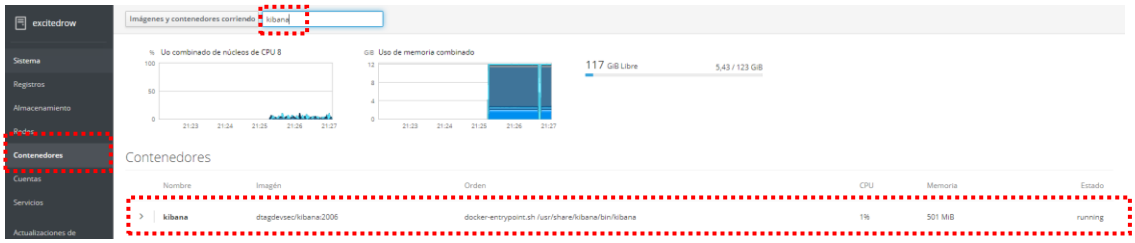


En el caso de Google Cloud también habrá que establecer una contraseña para el usuario root, para ello ejecutar el comando “passwd” e introducir la contraseña, en el ejemplo se usará “fabricio”.

```
[root@excitedrow:/home/fabricio_honeypot_3]# passwd
New password:
Retype new password:
passwd: password updated successfully
```

ERROR DE ACCESO A HERRAMIENTAS DE T-POT

Si las herramientas como Cokpit o Kibana no inician y muestran un error en el navegador puede ser debido a los recursos de hardware, se recomienda destinar al menos 8vCPUs y 16GB de RAM. En caso que los recursos de hardware sean los suficientes y el problema persista se deberá revisar que el contenedor se está ejecutando, una manera sencilla de verificar aquello es ingresar a la herramienta Cockpit que instaló T-Pot y en el menú “Contenedores” verificar que se está ejecutando Kibana.



ANEXO D

DEFINICIÓN DEL SERVICIO T-POT

```
[Unit]
Description=tpot
Requires=docker.service
After=docker.service

[Service]
Restart=always
RestartSec=5
TimeoutSec=infinity

# Get and set internal, external IP infos, but ignore errors
ExecStartPre=-/opt/tpot/bin/updateip.sh

# Clear state or if persistence is enabled rotate and compress logs from /data
ExecStartPre=-/bin/bash -c '/opt/tpot/bin/clean.sh on'

# Remove old containers, images and volumes
ExecStartPre=-/usr/bin/docker-compose -f /opt/tpot/etc/tpot.yml down -v
ExecStartPre=-/usr/bin/docker-compose -f /opt/tpot/etc/tpot.yml rm -v
ExecStartPre=-/bin/bash -c 'docker network rm $(docker network ls -q)'
ExecStartPre=-/bin/bash -c 'docker volume rm $(docker volume ls -q)'
ExecStartPre=-/bin/bash -c 'docker rm -v $(docker ps -aq)'
ExecStartPre=-/bin/bash -c 'docker rmi $(docker images | grep "<none>" | awk \'{print
$3}\')'

# Get IF, disable offloading, enable promiscious mode for p0f and suricata
ExecStartPre=-/bin/bash -c '/sbin/ethtool --offload $(/sbin/ip address | grep "^2: " | awk \'{
print $2 }\' | tr -d [:punct:]) rx off tx off'
ExecStartPre=/bin/bash -c '/sbin/ethtool -K $(/sbin/ip address | grep "^2: " | awk \'{ print
$2 }\' | tr -d [:punct:]) gso off gro off'
ExecStartPre=/bin/bash -c '/sbin/ip link set $(/sbin/ip address | grep "^2: " | awk \'{ print
$2 }\' | tr -d [:punct:]) promisc on'

# Set iptables accept rules to avoid forwarding to honeytrap / NFQUEUE
# Forward all other connections to honeytrap / NFQUEUE
ExecStartPre=/opt/tpot/bin/rules.sh /opt/tpot/etc/tpot.yml set

# Compose T-Pot up
ExecStart=/usr/bin/docker-compose -f /opt/tpot/etc/tpot.yml up --no-color

# Compose T-Pot down, remove containers and volumes
ExecStop=/usr/bin/docker-compose -f /opt/tpot/etc/tpot.yml down -v

# Remove only previously set iptables rules
ExecStopPost=/opt/tpot/bin/rules.sh /opt/tpot/etc/tpot.yml unset

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

ANEXO E

PRUEBAS DE INTERACCIÓN CON LOS HONEYPOTS

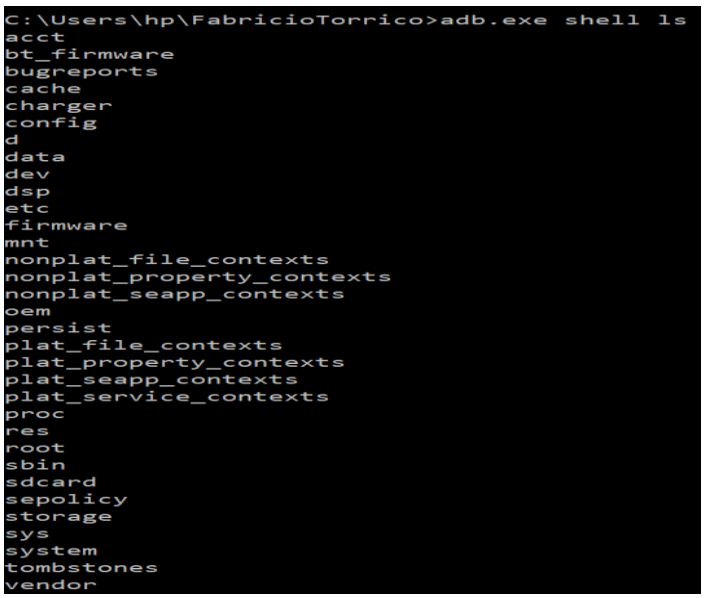
ADBHoney

Para interactuar con el honeypot ADBHoney se hace uso del ejecutable de ADB. Se lo puede hacer instalando el SDK de Android, o directamente descargando únicamente el ejecutable del enlace disponible en [98].

Primero se realiza la conexión a través de la dirección IP del T-Pot.

```
C:\Users\hp\FabricioTorrico>adb.exe connect 104.154.165.69
connected to 104.154.165.69:5555
```

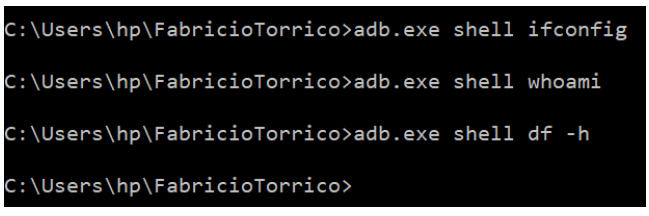
Se procede a listar los directorios del ADBHoney.



Se carga un archivo de prueba.

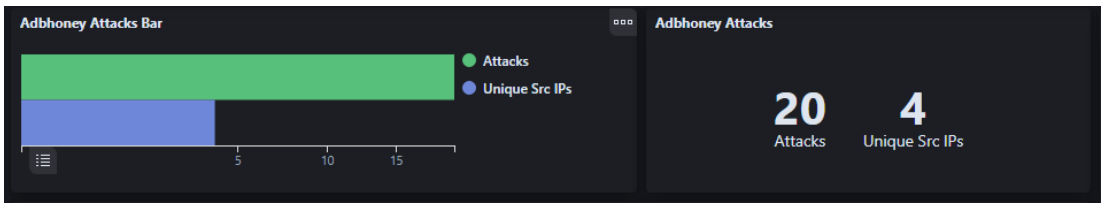
```
C:\Users\hp\FabricioTorrico>adb.exe push pruebaSubir.txt /mnt
pruebaSubir.txt: 1 file pushed, 0 skipped. 0.0 MB/s (25 bytes in 0.000s)
```

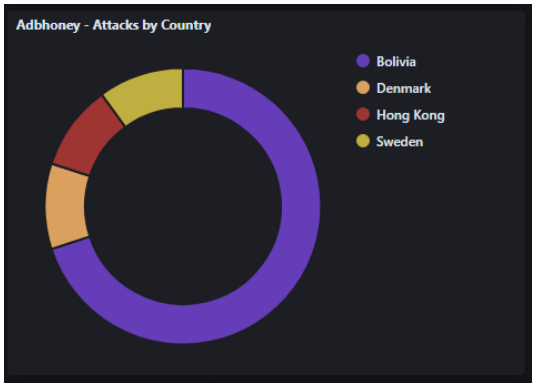
Se ejecutan otros comandos para evidenciar que no son funcionales.

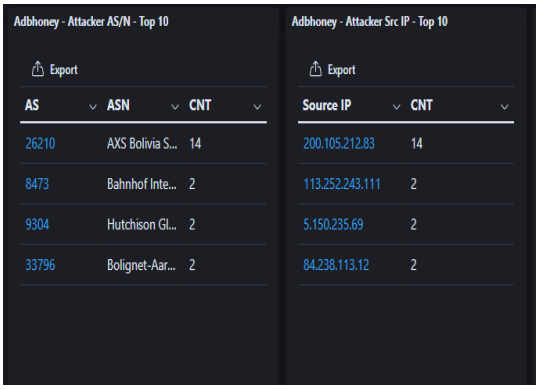


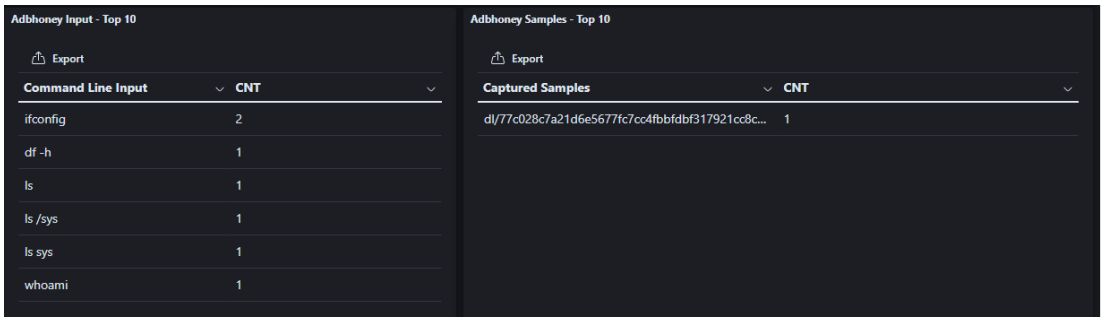
Finalmente, para verificar que el ataque fue registrado, se accede a la interfaz de Kibana de T-Pot. Comprobando de ese modo que en los últimos 15 minutos se recibieron 20 ataques de 4 diferentes direcciones IP ubicadas

en Bolivia, Dinamarca, Hong Kong y Suecia, además de los comandos ejecutados y el archivo de prueba que se subió.









CiscoASA Honeypot

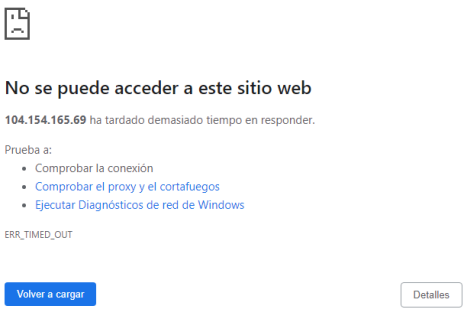
Para explotar la vulnerabilidad con CVE 2018-0101 se usa el exploit publicado en Exploit Database, el cual está escrito en Python.

```
#
# Cisco ASA CVE-2018-0101 Crash PoC
#
# We basically just read @saidelike slides:
# https://www.nccgroup.trust/globalassets/newsroom/uk/events/2018/02/reconbrx2018-robin-hood-vs-cisco-asa.pdf
#
# @zerosum0x0, @jennamagius, @aleph___naught
#

import requests, sys

headers = {}
headers['User-Agent'] = 'Open AnyConnect VPN Agent v7.08-265-gae481214-dirty'
headers['Content-Type'] = 'application/x-www-form-urlencoded'
headers['X-Aggregate-Auth'] = '1'
headers['X-Transcend-Version'] = '1'
headers['Accept-Encoding'] = 'identity'
headers['Accept'] = '*/*'
headers['X-AnyConnect-Platform'] = 'linux-64'
headers['X-Support-HTTP-Auth'] = 'false'
headers['X-Pad'] = '0000000000000000000000000000000000000000'

xml = """<?xml version="1.0" encoding="UTF-8"?>
<config-auth client="a" type="a" aggregate-auth-version="a">
    <host-scan-reply>A</host-scan-reply>
</config-auth>
"""

r = requests.post(sys.argv[1], data = xml, headers = headers, verify=False, allow_redirects=False)

print(r.status_code)
print(r.headers)
print(r.text)
```

Se ejecuta el script.

```
┌──(root💀magician)-[/home/magician]
└─# python3 poc-cve-2018-0101.py http://104.154.165.69
200
{'Content-Type': 'text/html', 'Set-Cookie': 'sess_uuid=b217e589-884e
tp/3.4.4'}
<!DOCTYPE html>

<html dir="ltr" lang="en" prefix="content: http://purl.org/rss/1.0/m
w.w3.org/2000/01/rdf-schema#  schema: http://schema.org/  sioc: http
```

Y se evidencia la denegación de servicio provocada.



Conpot

Para interactuar con el honeypot Conpot, se realiza una conexión a través del puerto 50100.

```
┌──(root💀magician)-[/home/magician]
└─# telnet 34.70.208.38 50100
Trying 34.70.208.38 ...
Connected to 34.70.208.38.
Escape character is '^]'.

Welcome ...
Connected to [00:13:EA:00:00:00]
```

Se ejecuta el comando para obtener ayuda “H”.

```
H
================================================================================
Service Menu
================================================================================
H:   Help [cmd].
Q:   Close connection.
!AC: Access control.
!AS: Alarm Server.
!GC: Get Config.
!GV: Software version.
!SA: Set KAP Server IP and port (*1).
!SB: Set 2nd KAP Server IP and port.
!SC: Set Config (*1).
!SD: Set device name (*1).
!SH: Set KAP Server lookup (DNS or DHCP)
!SI: Set IP (enter either valid IP or 0 to force DHCP)(*1).
!SK: Set KAP watchdog timeout(WDT).
!SN: Set IP for DNS Name servers to use.
!SP: Set IP Ports
!SS: Set Serial Settings.
!RC: Request connect
!RR: Request restart (*1).
!WM: Wink module.
================================================================================
(*1) Forces system restart
================================================================================
Kamstrup (R)
```

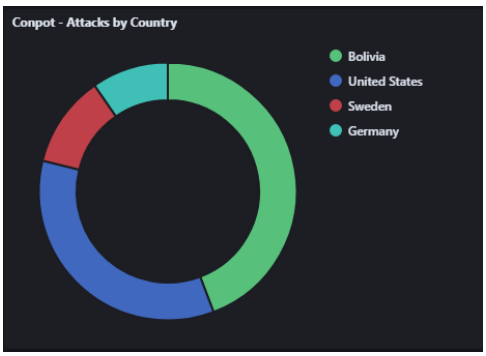
Se verifica la versión del software y sus parámetros configurados.

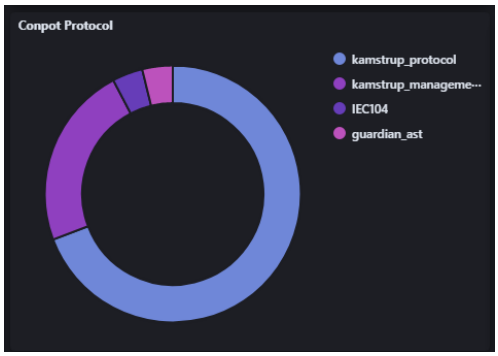
```
!GV

Software Version: 5.5 (E5)
!GC
Device Name         :
Use DHCP            : YES
IP addr.            : 192.168.1.210
IP Subnet           : 255.255.255.0
Gateway addr.       : 192.168.1.1
Service server addr.: 10.232.15.242
Service server hostname.: pwr_ctrl_mgmt01.int.local
DNS Server No. 1: 0.0.0.0
DNS Server No. 2: 0.0.0.0
DNS Server No. 3: 0.0.0.0
MAC addr. (HEX)     : 00:13:EA:00:00:00
Channel A device meterno.: A1 06 A1 02 B7 34 12 00 00 03
Channel B device meterno.: A1 06 A1 02 B7 34 12 00 00 03
Keep alive timer (flash setting): ENABLED 10
Keep alive timer (current setting): ENABLED 10
Has the module received acknowledge from the server: NO
KAP Server port: 50
KAP Local port: 800
Software watchdog: ENABLED 3600
```

Además, se realizan conexiones a los puertos publicados por las diferentes plantillas.

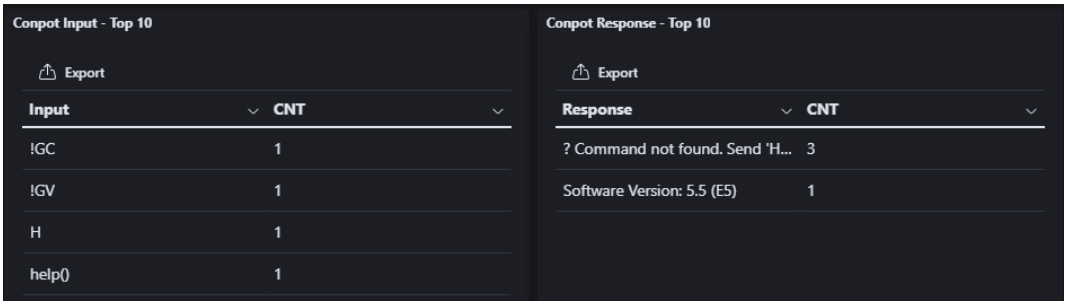
```
┌──(root💀magician)-[/home/magician]
└─# telnet 34.70.208.38 44818
Trying 34.70.208.38 ...
Connected to 34.70.208.38.
Escape character is '^]'.
┌──(root💀magician)-[/home/magician]
└─# telnet 34.70.208.38 47808
Trying 34.70.208.38 ...
Connected to 34.70.208.38.
Escape character is '^]'.
┌──(root💀magician)-[/home/magician]
└─# telnet 34.70.208.38 2404
Trying 34.70.208.38 ...
Connected to 34.70.208.38.
Escape character is '^]'.
┌──(root💀magician)-[/home/magician]
└─# telnet 34.70.208.38 10001
Trying 34.70.208.38 ...
Connected to 34.70.208.38.
Escape character is '^]'.
┌──(root💀magician)-[/home/magician]
└─# telnet 34.70.208.38 1025
Trying 34.70.208.38 ...
Connected to 34.70.208.38.
Escape character is '^]'.
```

Finalmente, se verifica el registro de las conexiones.



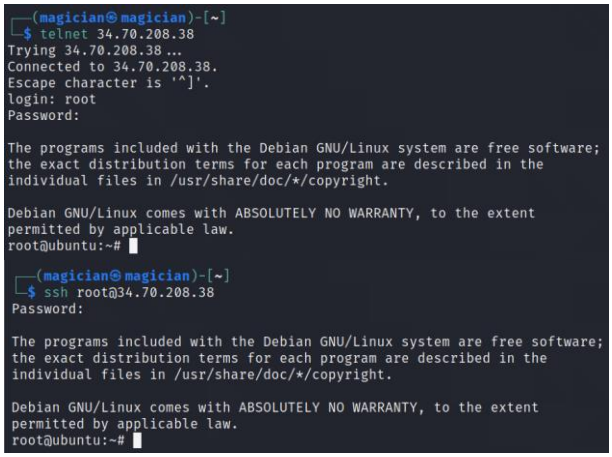


Así como los comandos ejecutados y las respuestas enviadas.

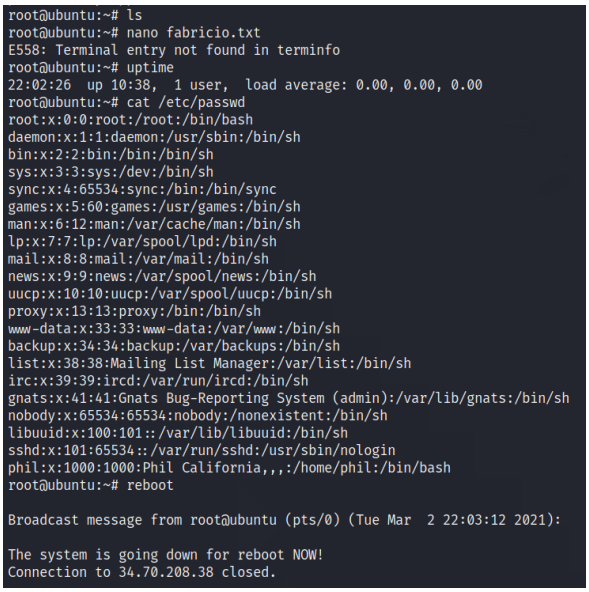


Cowrie

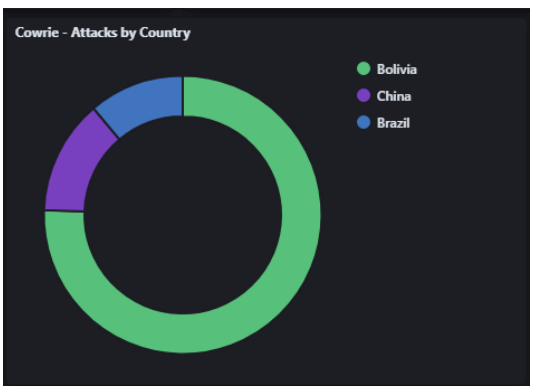
Para probar Cowrie, se realizan conexiones telnet y ssh con la palabra “root” como usuario y contraseña.

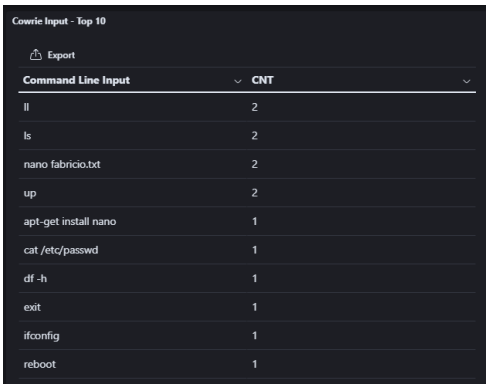


Tras realizar las conexiones se ejecutan diferentes comandos.



Se verifican las conexiones y comandos ejecutados.

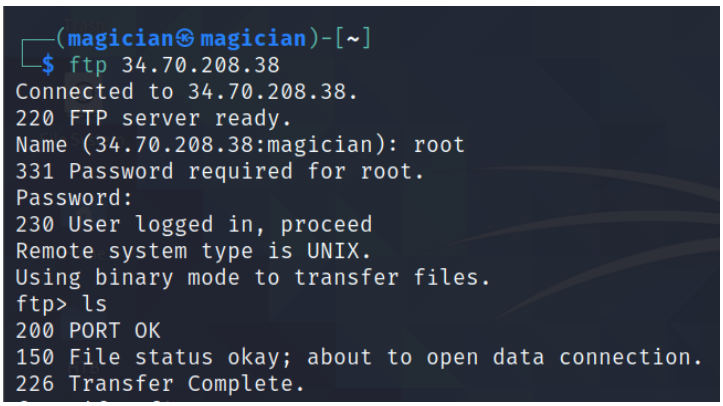




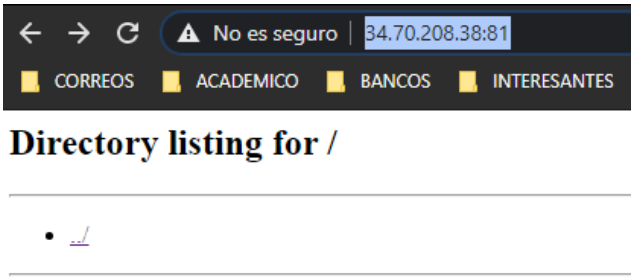
Dionaea

Se procede a probar los diferentes módulos de Dionaea.

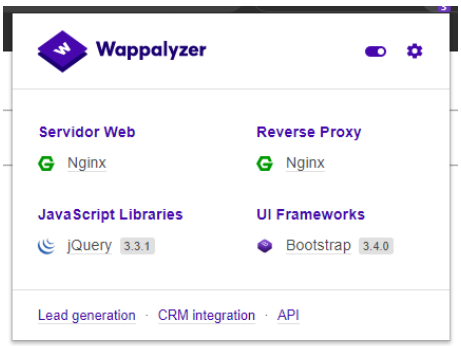
Para probar FTP, se realiza una conexión a través de línea de comandos.



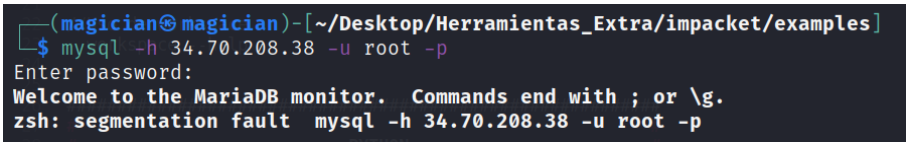
El servicio HTTP se prueba desde un navegador web, con el puerto 81.



Se corrobora que se trata de un servidor Nginx.



Se prueba el servicio MySQL a través de una conexión desde línea de comandos.



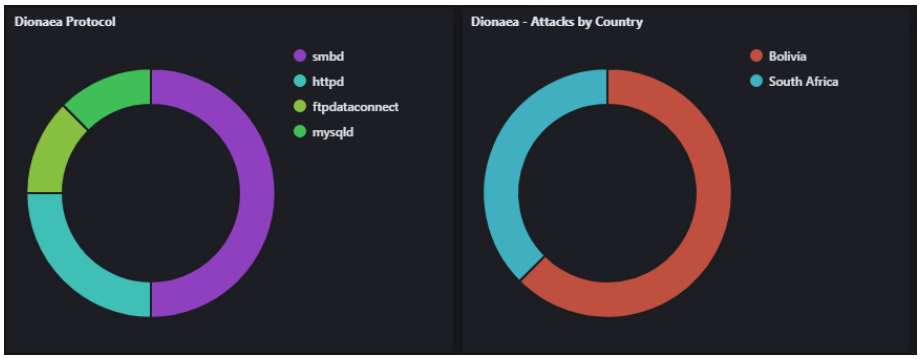
Se realiza una conexión a SMB.

```
┌──(magician㉿magician)-[~/Desktop/Herramientas_Extra/impacket/examples]
└─$ smbclient -N -L \\\\34.70.208.38\\
protocol negotiation failed: NT_STATUS_CONNECTION_DISCONNECTED
```

Y finalmente se prueba el servicio TFTP.

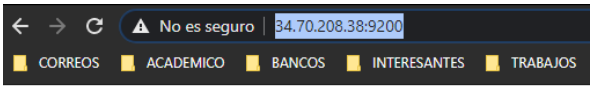
```
┌──(magician㉿magician)-[~/Desktop/Herramientas_Extra/impacket/examples]
└─$ tftp 34.70.208.38
tftp>
```

Se verifica las conexiones a través de los gráficos generados por Kibana.



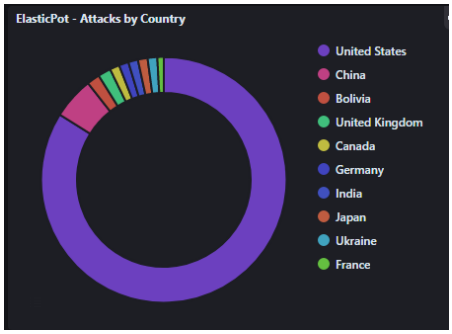
ElasticPot

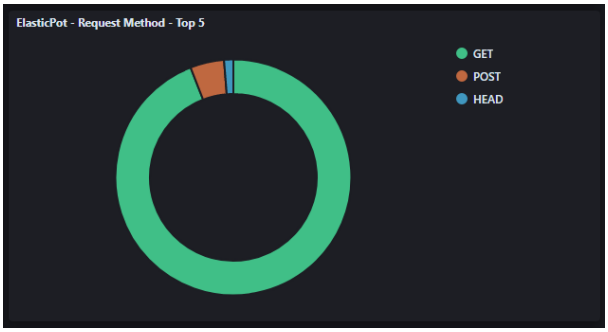
Para probar ElasticPot se accede desde un navegador al puerto 9200.

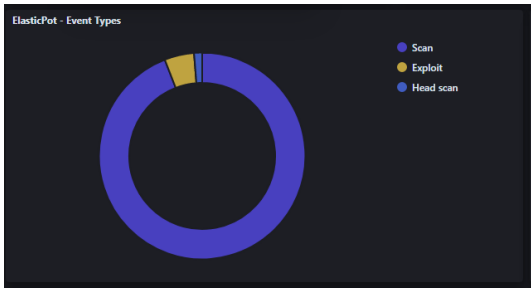


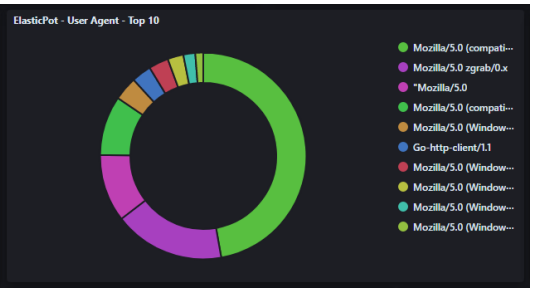

```
{
  "status" : 200,
  "name" : "USNYES01",
  "cluster_name" : "elasticsearch",
  "version" : {
    "number" : "1.4.1",
    "build_hash" : "b88f43fc40b0bcd7f173a1f9ee2e97816de80b19",
    "build_timestamp" : "2015-07-29T09:54:16Z",
    "build_snapshot" : false,
    "lucene_version" : "4.10.4"
  },
  "tagline" : "You Know, for Search"
}
```


Se comprueba las conexiones.



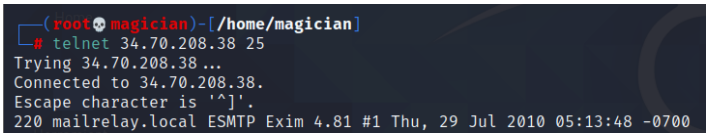




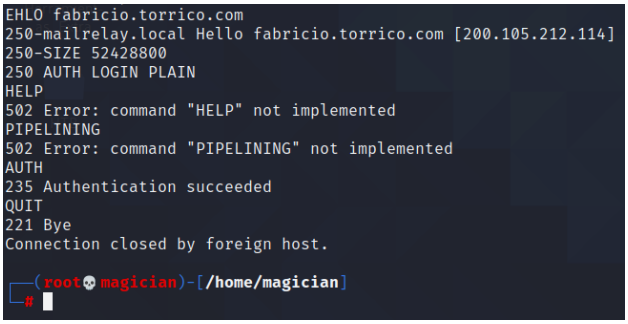


Mailoney

Para probar Mailoney, se realiza una conexión al puerto 25.



Una vez establecida la conexión, se envían algunos comandos al servidor SMTP.

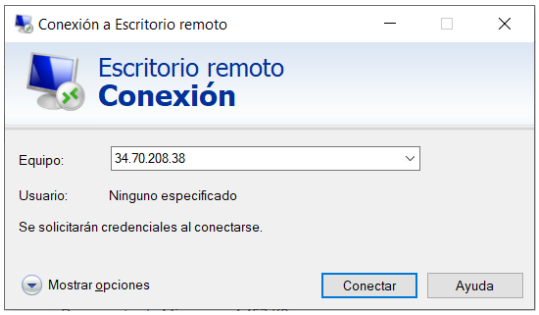


Finalmente, se evidencian los registros a través de Kibana, poniendo especial atención a los comandos enviados al servidor.

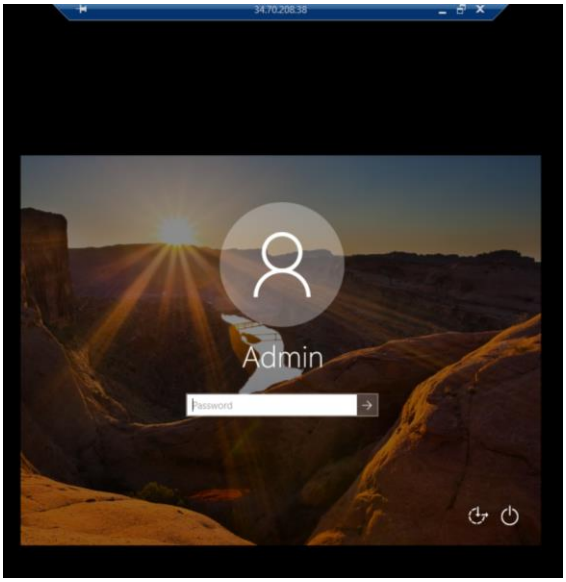
Data	CNT
STARTTLS	3
AUTH	1
EHLO fabricio.torrico.com	1
EHLO masscan	1
EHLO www.censys.io	1
EHLO zg-0226a-42	1
HELP	1
PIPELINING	1
QUIT	1

RDPY

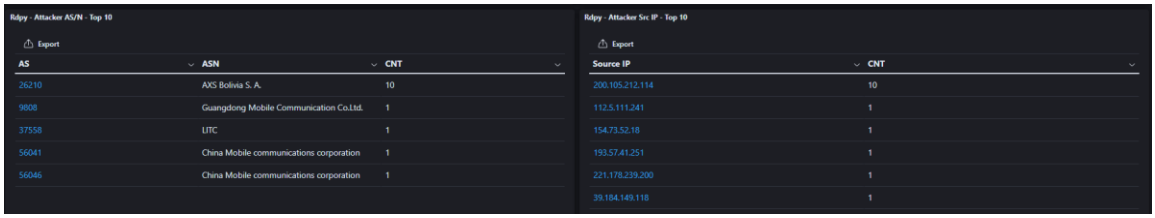
La prueba de RDPY consiste en una conexión de escritorio remoto a la IP asignada a T-Pot.



Se comprueba el acceso a escritorio remoto a una cuenta “Admin”, sin embargo tras pocos segundos la sesión se corta.

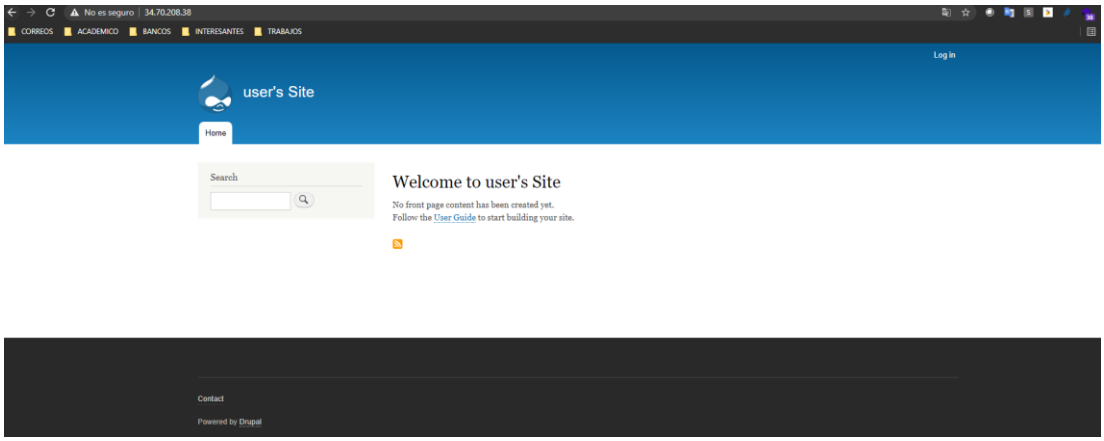


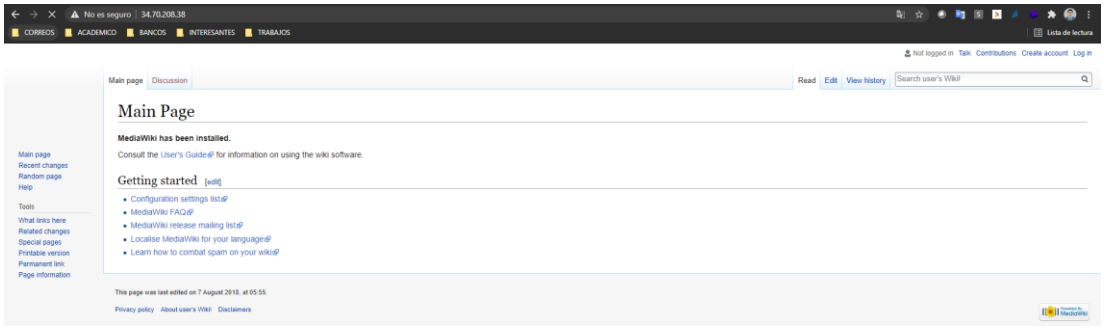
Se comprueban estas conexiones a través de la interfaz de Kibana.

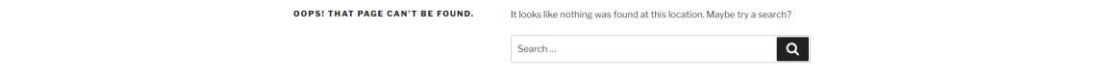


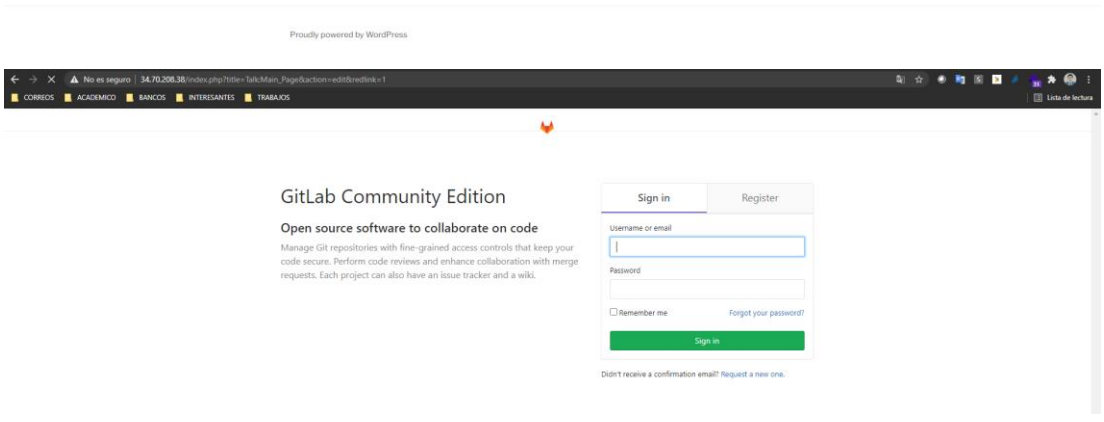
Snare y Tanner

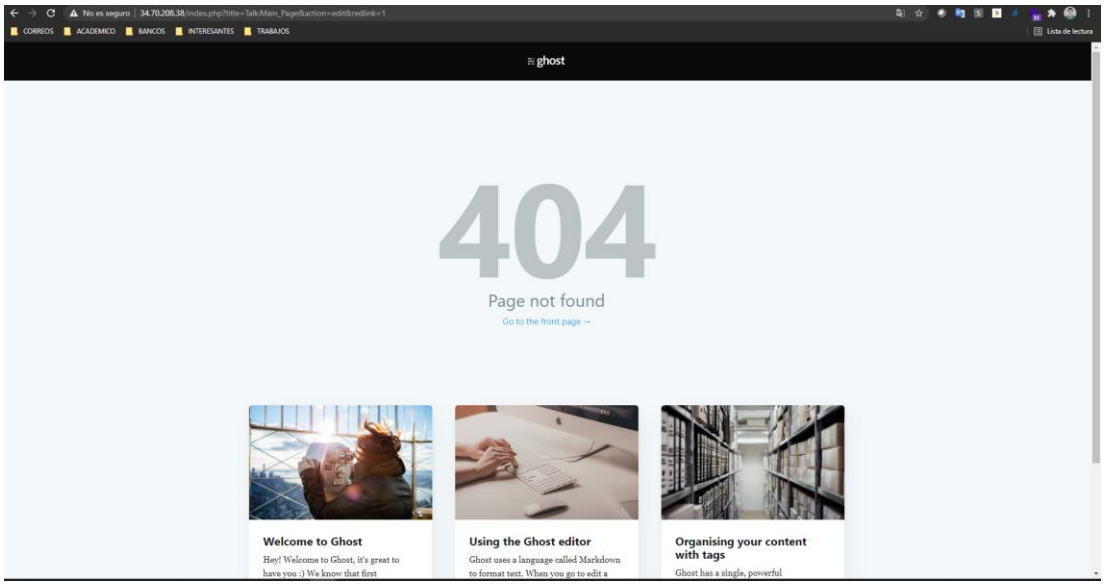
Las pruebas se las realiza accediendo desde cualquier navegador web a la dirección IP de T-Pot. Para verificar que publica diferentes sitios predefinidos, se procede a reiniciar el contenedor, obteniendo las siguientes páginas.

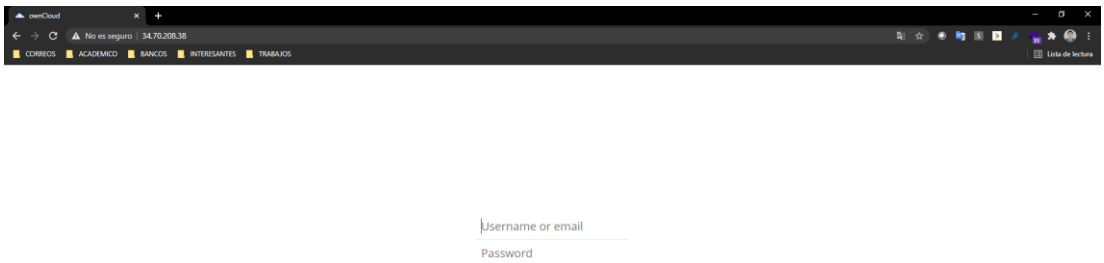









No es seguro | 34.70.208.38/index.php?title=Talk:Main_Page&action=edit&redlink=1
CORREOS
ACADEMICO
BANCOS
INTERESANTES
TRABAJOS
Lista de lectura
ghost
404
Page not found
Go to the front page →
Welcome to Ghost
Hey! Welcome to Ghost, it's great to have you :) We know that first
Using the Ghost editor
Ghost uses a language called Markdown to format text. When you go to edit a
Organising your content with tags
Ghost has a single, powerful


ownCloud
No es seguro | 34.70.208.38
CORREOS
ACADEMICO
BANCOS
INTERESANTES
TRABAJOS
Lista de lectura
Username or email
Password

BIBLIOGRAFÍA

[1] N. Poggi, «30 Estadísticas de Seguridad Informática que Importan (Actualizadas al 2021),» 19 5 2021. [En línea]. Available: https://preyproject.com/blog/es/30-estadisticas-seguridad-informatica/. [Último acceso: 19 01 2022].

[2] L. Spitzner, Honeypots: Tracking Hackers, Addison-Wesley Professional, 2002.

[3] N. Provos y T. Holz, Virtual Honeypots: From Botnet Tracking to Intrusion Detection, Addison-Wesley Professional, 2007.

[4] A. Rodríguez, «Go Daddy, ¿Qué es un honeypot y cómo usarlo en beneficio de tu negocio?,» 15 10 2019. [En línea]. Available: https://es.godaddy.com/blog/que-es-un-honeypot-y-como-usarlo-en-beneficio-de-tu-negocio/. [Último acceso: 18 01 2021].

[5] INCIBE, «INCIBE, Honeypot, una herramienta para conocer al enemigo,» 14 06 2018. [En línea]. Available: https://www.incibe-cert.es/blog/honeypot-herramienta-conocer-al-enemigo. [Último acceso: 18 01 2021].

[6] SeroBOT, «Wikipedia, Honeypot,» 11 11 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Honeypot. [Último acceso: 18 01 2021].

[7] A. Warburton, «Qué es un honeypot y cómo implementarlo en nuestra red,» 31 07 2020. [En línea]. Available: https://www.welivesecurity.com/la-es/2020/07/31/que-es-honeypot-como-implementarlo-nuestra-red/. [Último acceso: 18 01 2021].

[8] Kaspersky, «¿Qué es un honeypot?,» [En línea]. Available: https://latam.kaspersky.com/resource-center/threats/what-is-a-honeypot. [Último acceso: 18 01 2021].

[9] O. Espinosa, «Qué es y para qué sirve un Honeypot,» 08 06 2020. [En línea]. Available: https://www.redeszone.net/tutoriales/seguridad/que-es-honeypot/. [Último acceso: 18 01 2021].

[10] Telekom Profile, «Telekom Profile,» 2021. [En línea]. Available: https://www.telekom.com/en/company/company-profile. [Último acceso: 05 02 2021].

[11] Telekom Security, «Telekom Security,» 2021. [En línea]. Available: https://www.telekom.com/en/careers/our-focus-topics/it-security. [Último acceso: 05 02 2021].

[12] Telekom Security Alemania, «Telekom Security Alemania,» 2021. [En línea]. Available: https://geschaeftskunden.telekom.de/security. [Último acceso: 07 02 2021].

[13] D. Telekom, Escritor, Deutsche Telekom Company Presentation. [Performance]. Deutsche Telekom, 2020.

[14] T. S. Dashboard, «Telekom Security Dashboard,» [En línea]. Available: https://www.sicherheitstacho.eu/start/main. [Último acceso: 08 02 2021].

[15] TELEKOM T-POT, «Introducción a T-Pot: una plataforma multi-Honeypot,» 17 03 2015. [En línea]. Available: http://github.security.telekom.com/2015/03/honeypot-tpot-concept.html. [Último acceso: 05 02 2021].

[16] G. T-Pot, «Github T-Pot telekom-securiry/tpotce,» [En línea]. Available: https://github.com/telekom-security/tpotce. [Último acceso: 2 11 2020].

[17] «Android Debug Bridge (adb),» [En línea]. Available: https://developer.android.com/studio/command-line/adb?hl=es#issuingcommands. [Último acceso: 25 02 2021].

[18] G. Cirlig, «Gabriel Cirlig - ADBHoney - Analyzing the ADB Malware Ecosystem,» 15 1 2020. [En línea]. Available: https://www.youtube.com/watch?v=0YAJugdyqgw. [Último acceso: 25 2 2021].

[19] G. C. (Huuck), «ADBHoney GitHub,» [En línea]. Available: https://github.com/huuck/ADBHoney. [Último acceso: 25 02 2021].

[20] «What is the Cisco ASA?,» 21 10 2011. [En línea]. Available: https://www.cxtec.com/resources/blog/what-is-cisco-asa-security-appliance/. [Último acceso: 25 2 2021].

[21] «CiscoASA honeypot - GitHub,» [En línea]. Available: https://github.com/Cymmetria/ciscoasa_honeypot. [Último acceso: 25 2 2021].

[22] Citrix, «Entrega y seguridad de aplicaciones de Citrix,» Citrix, [En línea]. Available: https://www.citrix.com/es-mx/solutions/app-delivery-and-security/what-is-application-delivery-controller.html. [Último acceso: 27 2 2021].

[23] «Citrix Honeypot - GitHub,» [En línea]. Available: https://github.com/MalwareTech/CitrixHoneypot. [Último acceso: 27 2 2021].

[24] «Vulnerabilidad en Citrix CVE-2019-19781,» 08 01 2020. [En línea]. Available: https://www.incibe-cert.es/alerta-temprana/vulnerabilidades/cve-2019-19781. [Último acceso: 27 02 2021].

[25] «Sistema de Control Industrial,» [En línea]. Available: https://www.trendmicro.com/vinfo/us/security/definition/industrial-control-system. [Último acceso: 28 2 2021].

[26] «Qué es un sistema SCADA, para qué sirve y cómo funciona,» [En línea]. Available: https://www.cursosaula21.com/que-es-un-sistema-scada/. [Último acceso: 28 2 2021].

[27] «CONPOT ICS / SCADA Honeypot,» [En línea]. Available: http://conpot.org/. [Último acceso: 28 2 2021].

[28] S. A. T. Balderas, «Contpot: Honeypot de Sistemas de Control Industrial,» [En línea]. Available: https://revista.seguridad.unam.mx/numero29/conpot-honeypot-de-sistemas-de-control-industrial. [Último acceso: 28 2 2021].

[29] «Protocolo de red de teletipo (Telnet),» [En línea]. Available: https://www.extrahop.com/resources/protocols/telnet/. [Último acceso: 2 3 2021].

[30] D. C, «SSH - Hostinger,» 2020, 31 03. [En línea]. Available: https://www.hostinger.es/tutoriales/que-es-ssh. [Último acceso: 02 03 2021].

[31] «SSH (Secure Shell),» [En línea]. Available: https://en.wikipedia.org/wiki/SSH_(Secure_Shell). [Último acceso: 02 03 2021].

[32] M. Oosterhof, «Cowrie,» [En línea]. Available: https://github.com/cowrie/cowrie. [Último acceso: 02 03 2021].

[33] Kaspresky, «¿Qué son los ataques DDoS?,» Kaspresky, [En línea]. Available: https://latam.kaspersky.com/resource-center/threats/ddos-attacks. [Último acceso: 18 01 2022].

[34] «DDoSPot - GitHub,» GitHub, 27 12 2020. [En línea]. Available: https://github.com/aelth/ddospot. [Último acceso: 18 01 2022].

[35] «Digital Imaging and Communications in Medicine,» [En línea]. Available: https://www.sciencedirect.com/topics/biochemistry-

genetics-and-molecular-biology/digital-imaging-and-communications-in-medicine. [Último acceso: 2 3 2021].

[36] «About DICOM: Overview,» [En línea]. Available: https://www.dicomstandard.org/about. [Último acceso: 2 03 2021].

[37] «DICOM - Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/DICOM. [Último acceso: 02 03 2021].

[38] «DICOMPOT - GitHub,» [En línea]. Available: https://github.com/nsmfoo/dicompot. [Último acceso: 2 3 2021].

[39] «Documentación Dionaea,» [En línea]. Available: https://dionaea.readthedocs.io/en/latest/introduction.html. [Último acceso: 5 3 2021].

[40] The Honeynet Project, «Dionaea - Atrapando Insectos,» [En línea]. Available: https://www.honeynet.org/projects/active/dionaea/. [Último acceso: 5 3 2021].

[41] J. B. Vázquez, «POC: Captura de malware con el honeypot Dionaea - parte I,» [En línea]. Available: https://revista.seguridad.unam.mx/numero23/poc-captura-de-malware-con-el-honeypot-dionaea-parte-i. [Último acceso: 5 3 2021].

[42] «ElasticSearch,» [En línea]. Available: https://www.elastic.co/es/what-is/elasticsearch. [Último acceso: 5 3 2021].

[43] «ElasticPot - GitLab,» [En línea]. Available: https://gitlab.com/bontchev/elasticpot. [Último acceso: 5 3 2021].

[44] C. Wellons, «Endelessh,» Null Program, [En línea]. Available: https://nullprogram.com/blog/2019/03/22/. [Último acceso: 18 01 2022].

[45] «Endlessh: un tarpit SSH,» GitHub, 30 04 2021. [En línea]. Available: https://github.com/skeeto/endlessh. [Último acceso: 18 01 2022].

[46] «Glutton - GitHub,» [En línea]. Available: https://github.com/mushorg/glutton. [Último acceso: 5 3 2021].

[47] M. T. Sheikh, «An analysis of Glutton,» 13 6 2018. [En línea]. Available: https://cstayyab.medium.com/an-analysis-of-glutton-all-eating-honeypot-625adf70a33b. [Último acceso: 5 3 2021].

[48] «Using NFQUEUE and libnetfilter_queue,» [En línea]. Available: https://home.regit.org/netfilter-en/using-nfqueue-and-libnetfilter_queue/. [Último acceso: 5 3 2021].

[49] «Heralding - GitHub,» [En línea]. Available: https://github.com/johnnykv/heralding. [Último acceso: 3 5 2021].

[50] «HellPot - GitHub,» GitHub, 22 09 2021. [En línea]. Available: https://github.com/yunginnanet/HellPot. [Último acceso: 18 01 2022].

[51] «Honeypots - GitHub,» GitHub, 16 01 2022. [En línea]. Available: https://github.com/qeeqbox/honeypots. [Último acceso: 18 01 2022].

[52] «HoneyPy Docs,» [En línea]. Available: https://honeypy.readthedocs.io/en/latest/plugins/. [Último acceso: 5 3 2021].

[53] P. Galiana, «¿Qué es SAP y para qué sirve?,» 21 1 2020. [En línea]. Available: https://www.iebschool.com/blog/que-es-para-que-sirve-sap-management/. [Último acceso: 6 3 2021].

[54] «What is SAP? Meaning & Definition of SAP ERP Software,» [En línea]. Available: https://www.guru99.com/what-is-sap-definition-of-sap-erp-software.html. [Último acceso: 6 3 2021].

[55] «HoneySAP,» [En línea]. Available: https://honeysap.readthedocs.io/en/latest/user/intro.html. [Último acceso: 6 3 2021].

[56] «Honeytrap - The Honeynet Project,» [En línea]. Available: https://www.honeynet.org/projects/active/honeytrap/. [Último acceso: 16 3 2021].

[57] «Honeytrap - GitHub,» [En línea]. Available: https://github.com/armedpot/honeytrap/. [Último acceso: 16 3 2021].

[58] «Internet Printing Protocol,» 29 1 2021. [En línea]. Available: https://en.wikipedia.org/wiki/Internet_Printing_Protocol. [Último acceso: 16 3 2021].

[59] «IPPHoney - GitLab,» [En línea]. Available: https://gitlab.com/bontchev/ipphoney. [Último acceso: 16 3 2021].

[60] «¿Cómo me afecta y cómo actuar ante la vulnerabilidad CVE-2021-44228 Log4Shell?,» Kippeo, [En línea]. Available: https://kippeo.com/como-me-afecta-y-como-actuar-ante-la-vulnerabilidad-cve-2021-44228-log4shell/. [Último acceso: 18 01 2022].

[61] «Simple Mail Transfer Protocol,» 15 3 2021. [En línea]. Available: https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol#:~:text=The%20Simple%20Mail%20Transfer%20Protocol,send%20and%20receive%20mail%20messages.. [Último acceso: 16 3 2021].

[62] W. Duff, «What Is an SMTP Server?,» 17 7 2019. [En línea]. Available: https://sendgrid.com/blog/what-is-an-smtp-server/. [Último acceso: 16 3 2021].

[63] «Mailoney - GitHub,» [En línea]. Available: https://github.com/awhitehatter/mailoney. [Último acceso: 16 3 2021].

[64] «Fast Healthcare Interoperability Resources,» 26 2 2021. [En línea]. Available: https://en.wikipedia.org/wiki/Fast_Healthcare_Interoperability_Resources. [Último acceso: 16 3 2021].

[65] M. Klar, «FHIR vs HL7: EHR-wise, it’s an acronym you can connect with,» 4 12 2019. [En línea]. Available: https://www.adsc.com/blog/fhir-vs-hl7. [Último acceso: 16 3 2021].

[66] «Medpot - GitHub,» [En línea]. Available: https://github.com/schmalle/medpot. [Último acceso: 16 3 2021].

[67] «Remote Desktop Protocol,» 17 3 2021. [En línea]. Available: https://en.wikipedia.org/wiki/Remote_Desktop_Protocol. [Último acceso: 17 3 2021].

[68] B. Posey, «Remote Desktop Protocol (RDP),» 6 2020. [En línea]. Available: https://searchenterprisedesktop.techtarget.com/definition/Remote-Desktop-Protocol-RDP. [Último acceso: 17 3 2021].

[69] «RDPY - GitHub,» [En línea]. Available: https://github.com/citronneur/rdpy. [Último acceso: 17 3 2021].

[70] «Redis,» Wikipedia, 05 11 2021. [En línea]. Available: https://es.wikipedia.org/wiki/Redis. [Último acceso: 18 01 2022].

[71] «Redis Protocol specification,» Redis, [En línea]. Available: https://redis.io/topics/protocol. [Último acceso: 18 01 2022].

[72] «RedisHoneyPot,» 23 04 2021. [En línea]. Available: https://github.com/cypwnpwnsocute/RedisHoneyPot. [Último acceso: 18 01 2022].

[73] «Snare,» [En línea]. Available: https://snare.readthedocs.io/en/latest/quick-start.html#basic-concepts. [Último acceso: 17 3 2021].

[74] C. K. R. K. Wai Hoe Chong, «Learning cyberattack patterns with active honeypots,» 9 2018. [En línea]. Available: https://faculty.nps.edu/ncrowe/oldstudents/Chong_Koh_thesis.htm. [Último acceso: 17 3 2021].

[75] «Tanner,» [En línea]. Available: https://tanner.readthedocs.io/en/latest/quick-start.html. [Último acceso: 17 3 2021].

[76] «¿Qué es el ELK Stack?,» [En línea]. Available: https://www.elastic.co/es/what-is/elk-stack. [Último acceso: 18 3 2021].

[77] «¿QUÉ ES ELK? ElasticSearch, Logstash y Kibana,» Open Webinars, 30 7 2018. [En línea]. Available: https://openwebinars.net/blog/que-es-elk-elasticsearch-logstash-y-kibana/. [Último acceso: 18 3 2021].

[78] J. Mo, «What is ELK Stack?,» [En línea]. Available: https://www.missioncloud.com/blog/what-is-elk-stack. [Último acceso: 18 3 2021].

[79] «ELK Stack Tutorial: What is Kibana, Logstash & Elasticsearch?,» [En línea]. Available: https://www.guru99.com/elk-stack-tutorial.html. [Último acceso: 18 3 2021].

[80] «Elastic is a Search Company,» [En línea]. Available: https://www.elastic.co/es/videos/elastic-is-a-search-company. [Último acceso: 18 3 2021].

[81] «Logstash,» [En línea]. Available: https://www.elastic.co/es/logstash. [Último acceso: 18 3 2021].

[82] «Kibana,» [En línea]. Available: https://www.elastic.co/es/logstash. [Último acceso: 18 3 2021].

[83] «FATT - GitHub,» [En línea]. Available: https://github.com/0x4D31/fatt. [Último acceso: 21 3 2021].

[84] «Fatt: Network Metadata & Fingerprint Extractor,» [En línea]. Available: https://www.cyberpunk.rs/fatt-network-metadata-and-fingerprint-extractor. [Último acceso: 21 3 2021].

[85] «Operating System (OS) Fingerprinting with p0F,» 10 6 2016. [En línea]. Available: https://www.hackers-

arise.com/post/2016/06/10/operating-system-os-fingerprinting-with-p0f. [Último acceso: 21 3 2021].

[86] «P0f,» [En línea]. Available: https://lcamtuf.coredump.cx/p0f3/. [Último acceso: 21 3 2021].

[87] «Suricata,» [En línea]. Available: https://suricata-ids.org/. [Último acceso: 21 3 2021].

[88] M. Zamot, «An introduction to Cockpit, a browser-based administration tool for Linux,» 14 4 2020. [En línea]. Available: https://www.redhat.com/sysadmin/intro-cockpit. [Último acceso: 21 3 2021].

[89] A. F. Duval, «How I use Cockpit for my home's Linux server management,» 11 11 2020. [En línea]. Available: https://opensource.com/article/20/11/cockpit-server-management. [Último acceso: 21 3 2021].

[90] S. Henry, «Sitting in the Linux cockpit,» 13 2 2019. [En línea]. Available: https://www.networkworld.com/article/3340038/sitting-in-the-linux-cockpit.html. [Último acceso: 21 3 2021].

[91] «Cyberchef - GitHub,» [En línea]. Available: https://github.com/gchq/CyberChef. [Último acceso: 21 3 2021].

[92] «SpiderFoot - Documentación,» [En línea]. Available: https://www.spiderfoot.net/documentation/. [Último acceso: 23 3 2021].

[93] «SpiderFoot - Github,» [En línea]. Available: https://github.com/smicallef/spiderfoot. [Último acceso: 21 3 2021].

[94] «EWS Poster Armedpot - GitHub,» [En línea]. Available: https://github.com/armedpot/ewsposter. [Último acceso: 23 3 2021].

[95] «EWS Poster Telekom Security - GitHub,» [En línea]. Available: https://github.com/telekom-security/ewsposter. [Último acceso: 23 3 2021].

[96] «Heimdall,» [En línea]. Available: https://heimdall.site/. [Último acceso: 23 3 2021].

[97] «Awsome Honeypots,» Open Source Libs, 27 01 2022. [En línea]. Available: https://opensourcelibs.com/lib/awesome-honeypots. [Último acceso: 02 02 2022].

[98] I. Ramirez, «Cómo conectar un móvil Android al ordenador con ADB,» 23 9 2019. [En línea]. Available:

https://www.xatakandroid.com/tutoriales/como-conectar-movil-android-al-ordenador-adb. [Último acceso: 25 2 2021].

[99] J. I. Campoverde Armijos, Honeypot como herramienta de prevención de, Buenos Aires, 2018.

[100] «Telekom Security,» 20 8 2020. [En línea]. Available: http://github.security.telekom.com/index.html. [Último acceso: 2 11 2020].

[101] «Papers The Honeynet Project,» [En línea]. Available: https://www.honeynet.org/papers/. [Último acceso: 2 11 2020].

[102] «The Honeynet Project,» 2021. [En línea]. Available: https://www.honeynet.org/. [Último acceso: 04 02 2021].

[103] «eMobility - GitHub,» GitHub, 23 10 2017. [En línea]. Available: https://github.com/msbeiti/eMobility-Honeypot. [Último acceso: 24 01 2022].

[104] «VNClowPot - GitHub,» GitHub, 10 08 2019. [En línea]. Available: https://github.com/magisterquis/vnclowpot. [Último acceso: 24 01 2022].