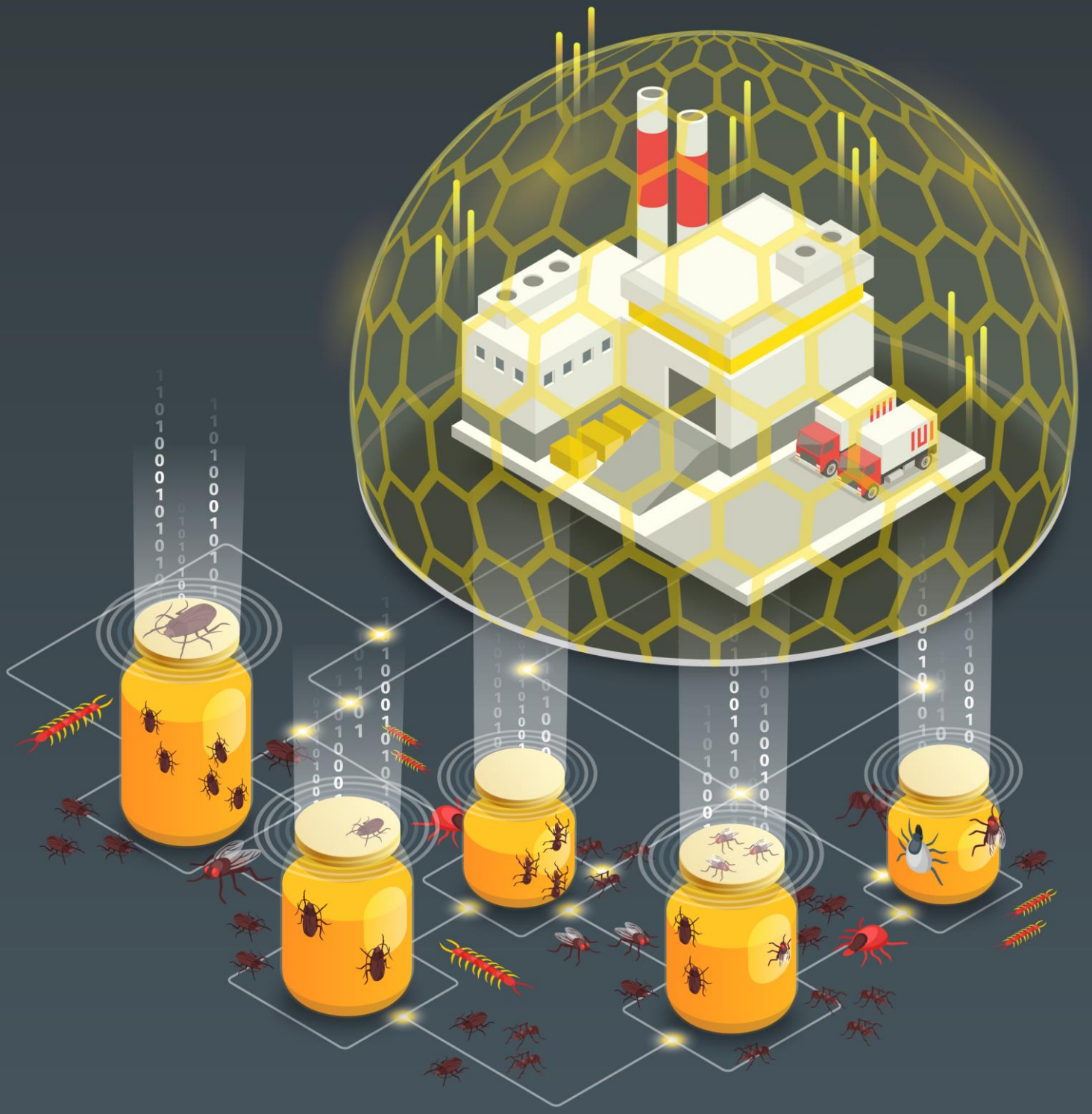


Guía de implantación de un honeypot industrial

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

incibe cert_

Octubre 2019

INCIBE-CERT_GUIA_IMPLANTACION_HONEYPOT_INDUSTRIAL_2019_v1

La presente publicación pertenece a INCIBE (Instituto Nacional de Ciberseguridad) y está bajo una licencia Reconocimiento-No comercial 3.0 España de Creative Commons. Por esta razón, está permitido copiar, distribuir y comunicar públicamente esta obra bajo las siguientes condiciones:

- Reconocimiento. El contenido de este informe se puede reproducir total o parcialmente por terceros, citando su procedencia y haciendo referencia expresa tanto a INCIBE o INCIBE-CERT como a su sitio web: https://www.incibe.es/. Dicho reconocimiento no podrá en ningún caso sugerir que INCIBE presta apoyo a dicho tercero o apoya el uso que hace de su obra.
- Uso No Comercial. El material original y los trabajos derivados pueden ser distribuidos, copiados y exhibidos mientras su uso no tenga fines comerciales.

Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra. Alguna de estas condiciones puede no aplicarse si se obtiene el permiso de INCIBE-CERT como titular de los derechos de autor. Texto completo de la licencia: https://creativecommons.org/licenses/by-nc-sa/3.0/es/.

Índice

1. Sobre esta guía 6
2. Organización del documento 7
3. Introducción 8
4. Honeynet 9
4.1. Descripción de una honeynet 9
4.2. Arquitectura general de una honeynet 9
4.3. Componentes 10
4.3.1. Honeywall 10
4.3.2. Honeypot 11
5. Clasificación de honeypot 12
5.1. Tipo de interacción 12
5.1.1. Alta interacción 12
5.1.2. Baja interacción 13
5.2. Tipo de equipamiento 14
5.2.1. Físico 14
5.2.2. Virtual 15
5.3. Tipo de comportamiento 15
5.3.1. Métodos de detección precisa 16
5.3.2. Protección contra intrusos 16
5.3.3. Desactivar acciones 16
5.3.4. Ralentización o defensa contra ataques automatizados 16
5.3.5. Ninguno 16
5.4. Tipo de rol 16
5.4.1. Cliente 16
5.4.2. Servidor 17
6. Herramientas 18
6.1. HoneyMonkey 18
6.2. BeEF 19
6.3. HoneyBadger 20
6.4. Spidertrap 21
6.5. Weblabyrinth 21
6.6. Network Obfuscation and Virtualized Anti-Reconnaissence 21
7. Proyectos 22
7.1. The Honeynet Project 22
7.2. Honeynet SCADA de DigitalBond 22
7.3. Conpot y GasPot 23
7.4. Honeyd 24

8. Despliegue de un honeypot industrial 25
8.1. Herramientas empleadas 25
8.2. Diseño del honeypot 25
8.3. Bastionado del sistema operativo anfitrión 26
8.4. Instalación Honeyd 26
8.4.1. Descarga de Honeyd 26
8.4.2. Instalación de dependencias 26
8.4.3. Compilación e instalación de Honeyd 27
8.5. Configuración de Honeyd 27
8.6. Mantenimiento del honeypot 31
8.7. Resultados 32
8.7.1. Herramientas empleadas 32
8.7.2. Comprobación de servicios 32
8.7.3. Análisis de Logs 39
9. Conclusiones 42
ANEXO: Cheat sheet 43
10. Referencias 44

ÍNDICE DE FIGURAS

Figura 1 Ejemplo de una arquitectura de honeynet 10
Figura 2 Clasificación de los honeypot 12
Figura 3 Honeypot de alta interacción 13
Figura 4 Honeypot de baja interacción 13
Figura 5 Honeypot con rol cliente 17
Figura 6 Honeypot con rol servidor 17
Figura 7 Fases de ejecución de HoneyMonkey 19
Figura 8 Logotipo de BeEF 19
Figura 9 Logotipo de HoneyBadger 20
Figura 10 Logotipo de NOVA 21
Figura 11 Logotipo de The Honeynet Project 22
Figura 12 Propuesta de arquitectura para una honeynet industrial 23
Figura 13 Logotipos de Conpot y GasPot 24
Figura 14 Comandos de instalación de Git y descarga de Honeyd 26
Figura 15 Comando para la instalación de las dependencias 27
Figura 16 Comandos para la instalación de Honeyd 27
Figura 17 Creación de la carpeta 27
Figura 18 Estado de la carpeta en este punto 28
Figura 19 Línea editada en el fichero “honeyd-http-siemens.py” 29
Figura 20 Línea editada en el fichero “honeyd-telnet-siemens.py” 29
Figura 21 Nombre completo del SO obtenido de nmap-os-db 29
Figura 22 Nombre del SO añadido en la última línea de nmap.assoc 29
Figura 23 Carpeta previamente creada, con el archivo de configuración 30
Figura 24 Contenido del archivo de configuración honeyd.conf 30
Figura 25 Comando y parámetros para ejecutar la máquina virtual de Honeyd 31
Figura 26 Resultado de ejecutar el comando de honeyd 31
Figura 27 Resultado del escaneo de puertos en el honeypot 33

Figura 28 Resultado escaneo del SO del honeypot 34
Figura 29 Página web emulada en el honeypot 35
Figura 30 Resultado de la conexión FTP al honeypot 35
Figura 31 Estado ModbusTool en Idle 36
Figura 32 Estado ModbusTool en Connected 36
Figura 33 Desplegable de funciones ModbusTool 37
Figura 34 Intercambio de paquetes de lectura de registros entre ModbusTool y el honeypot 37
Figura 35 Intercambio de paquetes de escritura de un registro entre ModbusTool y el honeypot 37
Figura 36 Captura de pantalla del cliente snap7 38
Figura 37 Intercambio de paquetes de conexión entre el cliente Snap7 y el honeypot 38
Figura 38 Resultado de un intento de acceso Telnet al honeypot 39
Figura 39 Resultado de ejecución del snmpwalk contra el honeypot 39
Figura 40 Formato log de peticiones ICMP 40
Figura 41 Formato log de peticiones FTP 40
Figura 42 Conexiones Telnet 40
Figura 43 Formato log de peticiones SNMP 40
Figura 44 Peticiones HTTP al honeypot 40
Figura 45 Conexiones Modbus 40
Figura 46 Conexión S7comm 41

ÍNDICE DE TABLAS

Tabla 1 Comparativa de honeypots de alta y baja interacción 14
Tabla 2 Comparativa de honeypots físicos y virtuales 15
Tabla 3 Listado de las herramientas empleadas 25
Tabla 4 Listado de herramientas utilizadas en la fase de pruebas 32

1. Sobre esta guía

En esta guía se define el concepto de honeypot, los requisitos recomendados para su correcta implementación, su clasificación en base a distintos criterios (iteración, equipamiento, comportamiento y rol) y su evolución hasta nuestros días, prestando especial atención a las honeynet, la forma en la que suelen implementarse.

En ella, se incluyen los pasos necesarios para la construcción de un honeypot industrial desde cero, con ilustraciones y ejemplos, indicando la utilización que se le podrá dar y sus características más importantes.

2. Organización del documento

Este documento consta de una 3.Introducción sobre las honeynets, detallando aún más los honeypot, a través de 5 apartados principales, divididos en diferentes categorías, donde se recogen puntos como, qué son, la forma de clasificarlos, los cambios que han tenido desde sus inicios, los proyectos más conocidos y la creación de un honeypot industrial paso a paso.

El primer apartado de contenido, 4.-Honeynet, se centra en la definición de las partes que componen esta red, la cual consta de un honeywall, que hace de cortafuegos, y honeypots, que actúan como señuelos para recabar información de posibles atacantes.

Explicados los conceptos relacionados con el estudio, llega el momento de la 5.-Clasificación de honeypot, que recoge los diferentes tipos posibles en función del criterio elegido. Según el tipo de interacción (alta o baja), el tipo de equipamiento que usaremos (máquinas físicas, virtuales o ambas), el uso o comportamiento que le queremos dar a nuestro honeypot (ya sea para detectar, protegernos contra intrusos, desactivar las acciones del atacante o perjudicar al atacante mediante una defensa que lo ralentice) y el tipo de rol que tendrá el honeypot, servidor (preparado para recibir ataques) o cliente (para enviar peticiones contra redes o dispositivos maliciosos).

El apartado 6.-Herramientas, se centra en cómo han ido cambiando y mejorando la seguridad y la forma de recolectar información sobre el atacante a lo largo de los años, para mejorar la defensa de los sistemas de control industrial.

A continuación, se hace un repaso por los diferentes 7.-Proyectos más destacados sobre honeypots industriales, indicando cuándo se han creado, sus principales objetivos y para qué se crearon.

El último apartado, 8.-Despliegue de un honeypot industrial, describe, a modo de manual de instalación, los pasos para crear desde cero un honeypot industrial, teniendo en cuenta las herramientas que se van a emplear, el diseño, un buen bastionado del sistema operativo base, además de una buena configuración y su posterior mantenimiento.

Para cerrar la guía, el capítulo 9.-Conclusiones recoge los puntos importantes aprendidos a lo largo de toda la guía.

3. Introducción

La búsqueda de cómo mejorar la ciberseguridad de nuestros sistemas es un proceso continuo cuyo principal problema reside en determinar cómo puede actuar un atacante o los métodos que utilizará para conseguir que sus acciones tengan éxito.

Una de las herramientas utilizadas con este propósito es la denominada honeynet, una red diseñada con diferentes herramientas y máquinas (Windows, Linux, Solaris, etc.) que se dedica exclusivamente a ser el objetivo de los atacantes para posteriormente, ser capaces de detectarlos, obtener información sobre estos y saber de qué manera han llevado a cabo los diferentes ataques contra él, con la finalidad de poder desarrollar medidas de protección contra esos ciberataques.

Como diseñar y desplegar una honeynet a veces no es una tarea tan trivial como nos gustaría, y dado que esta consiste en una red de honeypots conectados, nos centraremos en definir uno de esos nodos, que es un término más extendido y conocido. Estos honeypot son herramientas o sistemas que están diseñados para engañar al atacante, de ahí su nombre, “tarro de miel”, en relación a su finalidad de atraer a los atacantes. El empleo de este tipo de herramientas puede suponer, tanto para empresas como investigadores, la obtención de información valiosa acerca de los atacantes, al igual que proporcionar protección ante posibles intentos de intrusión, a los sistemas de control industrial, ya que su uso sirve como prevención, detección y respuesta.

4. Honeynet

Día a día, los ciberatacantes van aprendiendo y empleando nuevas técnicas para intentar comprometer los sistemas, lo cual implica el aumento de ciberataques de forma global.

Una honeynet es un tipo especial de red que ha sido preparada y diseñada, como medio de defensa, para poder ser atacada y poder recabar gran cantidad de información acerca de los métodos y técnicas utilizadas por los ciberatacantes. El empleo de este tipo de redes comenzó a utilizarse en 1999 con el proyecto "*The Honeynet Project*"[1], desde entonces su fundador Lance Spitzner, dio a conocer el concepto de honeynet.

4.1. Descripción de una honeynet

Con una honeynet se quiere conseguir una simulación lo más realista posible de lo que sería una red real, incluyendo sistemas de producción, servidores, servicios, etc. Están diseñadas para que puedan ser comprometidas por los atacantes y estructuradas para extraer, de la mejor forma posible, la mayor cantidad de información, de cara a aprender sobre las técnicas y herramientas utilizadas por los ciberatacantes.

El grado de éxito de una honeynet recae en conseguir monitorizar todos los movimientos y acciones que realizan los atacantes dentro de la red. Todos los rastros que van dejando los ciberatacantes, debido a sus acciones y la utilización de herramientas, son analizados y monitorizados para poder saber cuáles son las tácticas utilizadas y cuál es el objetivo final que se quiere alcanzar. Además, un factor importante, es la pericia de los atacantes pues, si el atacante tiene experiencia, puede ser capaz de detectar si es una honeynet y detener su ataque. Además, existen herramientas en Internet que pueden detectar redes que utilizan honeypots, como puede ser *Honeypot Or Not* de *Shodan*[2].

El problema más significativo de la honeynet recae en identificar y localizar las actividades que realizan los ciberatacantes dentro de la red, lo cual se realiza mediante una herramienta de captura de tráfico (Wireshark, tcpdump, etc.), un análisis y estudio de los paquetes que circulan por la red, tratando de identificar el tráfico del atacante, con el fin de monitorizarlo e intentar averiguar sus técnicas, herramientas y objetivos.

4.2. Arquitectura general de una honeynet

En general, no existen unos modelos de arquitectura concretos a la hora de montar una honeynet, es decir, no existe un estándar, por lo que la gran mayoría de ellas presentan una arquitectura diferente, variando las herramientas para el control, los elementos para una simulación industrial, como PLC, HMI, SCADA, o el registro de actividades y los tipos de análisis empleados para las acciones de los intrusos. Al fin y al cabo, una honeynet no es un producto, no se instala un software para luego ponerse en funcionamiento, es una red altamente controlada para contener y analizar atacantes en tiempo real.

Los componentes principales que forman una honeynet son:

[1] https://www.honeynet.org/

[2] https://honeyscore.shodan.io/

- un *gateway* denominado honeywall por el que pasa todo el tráfico, tanto entrante como saliente, y por el que ha de pasar obligatoriamente el atacante;
- los honeypots son equipos destinados a simular ser equipos finales y, por lo tanto, candidatos a ser atacados. Si se trata de simular un entorno industrial, con componentes de automatización, se incluyen componentes, como PLC, SCADA, HMI, RTU, etc., elementos para que el atacante crea que está dentro de la red de la industria y no dentro de un entorno replicado. Estos siempre están preparados para simular un entorno lo más realista posible, intentando engañar al atacante para hacerle pensar que está accediendo a un equipo real en producción, ya que, si en algún momento tuviera alguna sospecha, podría finalizar el ataque sin dejar ninguna información.

4.3. Componentes

A continuación, se describen los dos componentes principales de una honeynet.

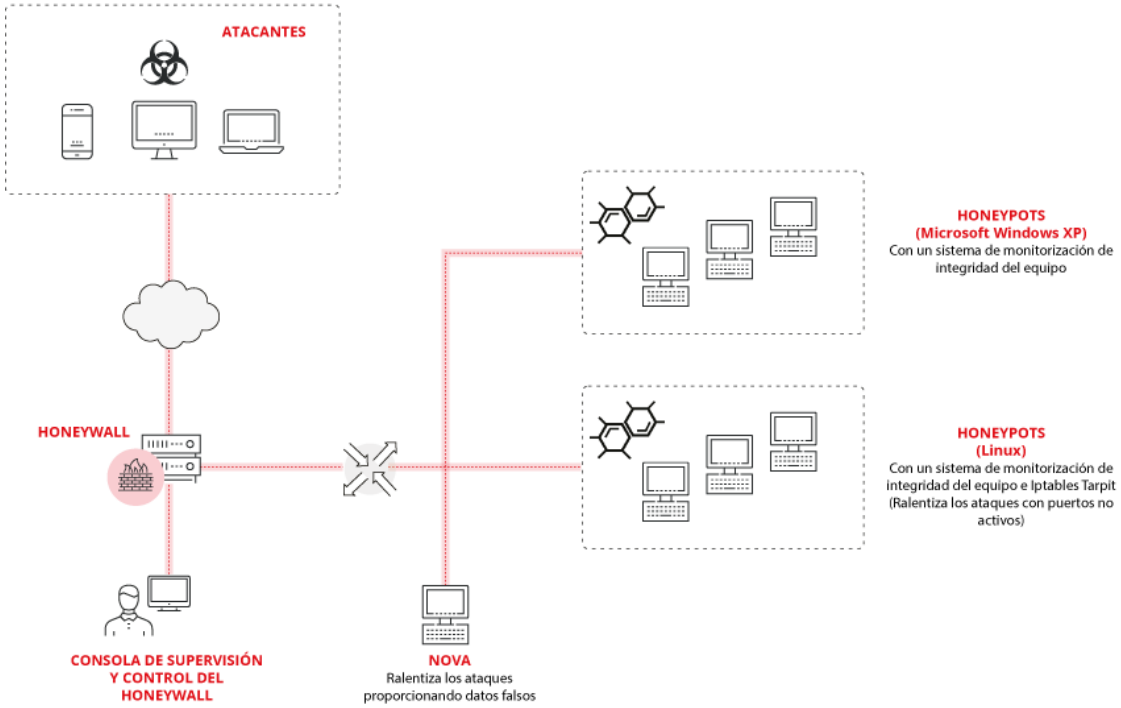


Figura 1 Ejemplo de una arquitectura de honeynet

4.3.1. Honeywall

Un honeywall es una máquina que está exclusivamente preparada para actuar como un cortafuegos, es decir, para filtrar y monitorizar el tráfico que se genera en una honeynet, por lo que, lo ideal, es que en él se utilicen o combinen herramientas de auditoría, analizadores de red e IDS.

De la honeynet se deben de capturar la mayor cantidad de datos e información útiles para que, posteriormente, se puedan analizar y extraer nuevos tipos de ataques, estrategias y herramientas utilizadas por los intrusos. Todo lo anterior, debe realizarse sin que el ciberatacante se dé cuenta de que está siendo monitorizado, por lo que todo el proceso de captura y análisis de datos tiene que realizarse de la forma más transparente y cuidadosa posible.

Respecto al diseño del honeywall, lo ideal es tener varios elementos, denominados sensores, que vayan recolectando datos en varios lugares, tanto dentro como fuera de la honeynet, es decir, realizar una organización por capas. Uno de los componentes principales es el cortafuegos, que se coloca en la entrada para poder analizar y capturar todo el tráfico de datos entrante y saliente. Además, servirá para alertar y avisar en el momento que se esté realizando un ataque. Otro elemento que se puede implementar es un sistema de detección de intrusos (IDS), utilizado para analizar el tráfico de la red, comparándolo con firmas de ataques ya conocidos o según los patrones de comportamiento sospechosos que tenga el atacante, como por ejemplo el escaneo de puertos o el envío y recepción de paquetes malformados, entre otros.

Lo ideal es implementar estas dos herramientas, convirtiendo el honeywall en una herramienta completa que combina la inteligencia y la capacidad de bloqueo, siendo este el punto por donde deben pasar todos los paquetes para entrar en la red interna de la honeynet.

4.3.2. Honeypot

Un honeypot es un sistema simulado que sirve de herramienta de seguridad, se sitúa en una red y está diseñada para recibir ataques con el principal objetivo de obtener datos valiosos sobre los ciberatacantes y sus métodos de ataque, herramientas utilizadas, técnicas de intrusión y modus operandi.

A la hora de montar un honeypot, hay que decidir entre las dos alternativas que existen, un equipamiento físico o un equipo virtual. Las diferencias entre ambos son que un equipo físico es totalmente funcional y al atacante no le será tan sencillo descubrir que está dentro de un honeypot; en cambio, el honeypot simulado no es funcional de forma completa, posee un servicio que permite imitar miles de sistemas operativos y sus características a la vez. Las herramientas existentes para crear un honeypot virtualizado tienen la capacidad de simular el sistema operativo a nivel de pila TCP/IP, por lo que permiten engañar a herramientas de escaneo de red y puertos como nmap y xprobe; al ser un subsistema de virtualización, le permite tener servicios reales como http, ftp o telnet, con el objetivo de hacer creer al atacante que está en un entorno real.

Para entornos de SCI, habrá que tener en cuenta diferentes aspectos de cara a montar una red industrial lo más completa posible, sobre todo de cara a dispositivos PLC o sistemas SCADA, ya que habrá que decidir cuales se emplean y definir si se deben utilizar reales o virtuales. El hecho de simular la red industrial por completo implica incluir cada uno de los elementos correspondientes en la red, disponer de elementos de control y automatización para poder desplegar una honeynet en condiciones y obtener la mayor y mejor información que sea posible.

5. Clasificación de honeypot

Los honeypots pueden clasificarse según diversas características o funcionalidades, por lo que antes de montar el honeypot hay que fijar cuales son nuestros objetivos para, de esta forma, elegir el tipo de honeypot que mejor se adapte a dichas necesidades según su funcionalidad. A continuación, veremos una clasificación de honeypots según los tipos de interacción, de equipamiento, de comportamiento y de rol.

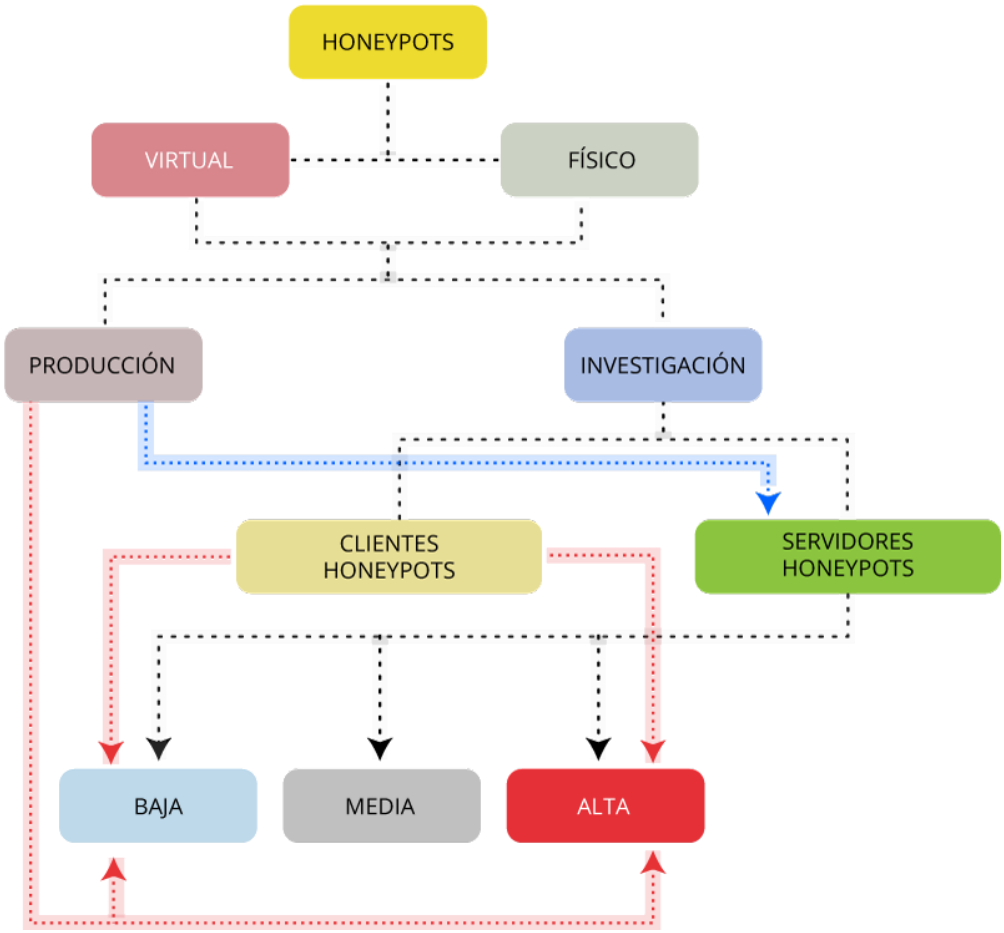


Figura 2 Clasificación de los honeypot

5.1. Tipo de interacción

Según el grado de complejidad que tienen los honeypots de cara al montaje y la interacción que le permite tener al atacante, pueden clasificarse como de alta o baja interacción.

5.1.1. Alta interacción

Los honeypots de alta interacción suelen ser sistemas con aplicaciones instaladas y completamente funcionales. De cara a la extracción de información, este tipo de sistemas va a aportar más información que un sistema de baja interacción sobre las acciones, técnicas y herramientas que utiliza el usuario dentro del honeypot.

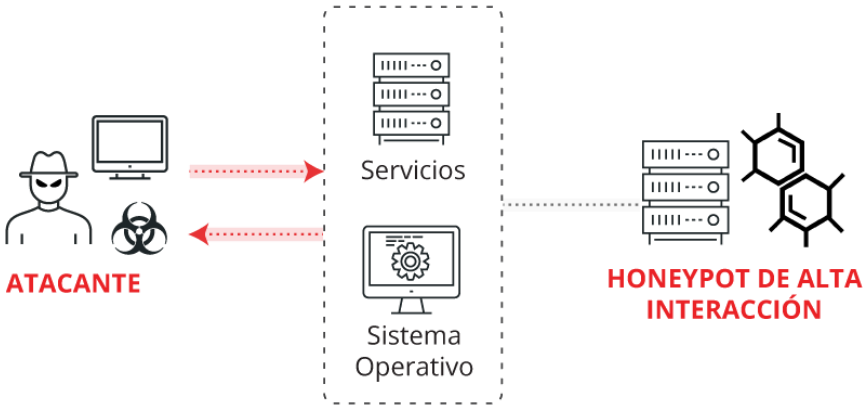


Figura 3 Honeypot de alta interacción

Hay que tener cuidado con la configuración de este tipo de honeypots, puesto que los atacantes, en el caso de no tenerlo bien configurado y controlado, podrían utilizarlo como un punto de acceso al resto de sistemas de la red, por lo que siempre se debe tener aislado, para que ningún sistema se vea comprometido.

5.1.2. Baja interacción

Los sistemas de baja interacción suelen ser sistemas de fácil manejo. Tienen una rápida puesta en funcionamiento, ya que una vez se tiene instalado y configurado, únicamente habría que ponerlo en marcha. Sin embargo, las aplicaciones y servicios que simula no son completamente funcionales, ya que disponen de implementaciones parciales, a menudo repetitivas, lo que hace que sean fácilmente identificables por atacantes expertos, por lo que suelen ser más limitados a la hora de recopilar información.

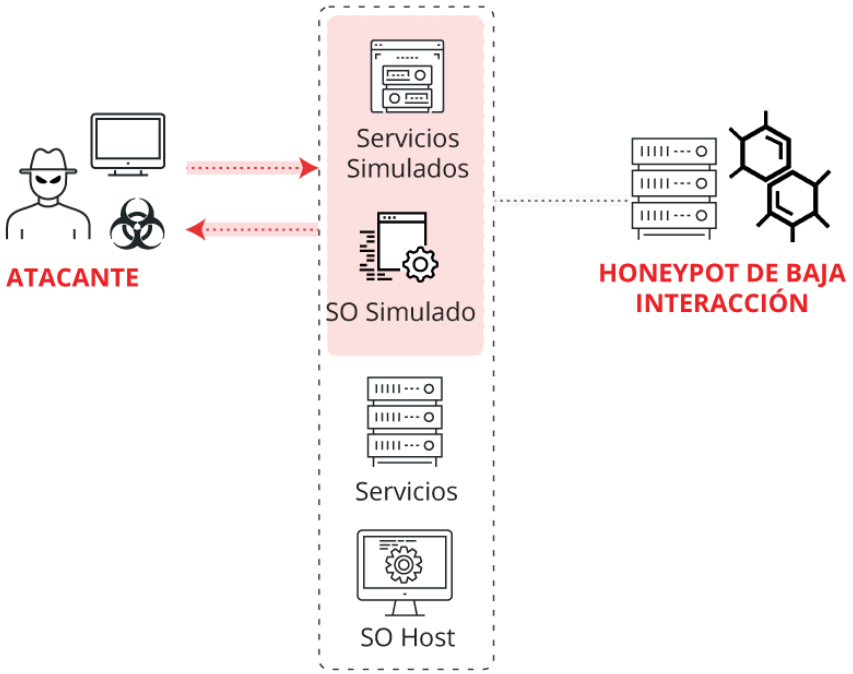


Figura 4 Honeypot de baja interacción

	Alta Interacción	Baja Interacción
Simulación	Emplean servicios reales, aplicaciones o dispositivos. Su identificación suele ser compleja.	Simulan servicios o sistemas operativos. Tiene muchas posibilidades de detectarse como trampa fácilmente.
Objetivos	Descubrir nuevos ataques o comportamientos anómalos anteriormente no detectados.	Descubrir herramientas automatizadas o de vulnerabilidades ya conocidas en servicios concretos.
Información	Capturan una gran cantidad de información de gran valor por contener en ocasiones registros de ataques no conocidos. Su implementación es perfecta para investigaciones y análisis en profundidad.	La cantidad de recursos recopilados es limitada. No es la mejor opción si se quiere realizar un análisis en profundidad del sistema.
Riesgos	El riesgo al que se ven expuestos frente a los atacantes es mayor ya que puede tener toda la red a su alcance si no se aísla adecuadamente el honeypot.	El riesgo al que se ven expuestos frente a los atacantes es menor debido a que todo suele estar virtualizado.

Tabla 1 Comparativa de honeypots de alta y baja interacción

5.2. Tipo de equipamiento

En función de los dispositivos que se quieran emplear de cara a montar un honeypot, la red podrá estar formado por sistemas virtuales o físicos. Por lo que, de cara a montar un honeypot, ya sea de alta o baja interacción, se puede realizar empleando o combinando distintos tipos de equipamiento.

5.2.1. Físico

Un honeypot físico se trata de un equipo real dispuesto para ser atacado desde el exterior. Al ser un equipo físico, su funcionalidad es la que ofrezca de serie el equipo, sin ningún tipo de restricción. Esto favorece que el atacante no lo identifique como un honeypot, pero limita el número de ataques al no poder seleccionar los servicios que se desea que estén disponibles.

El precio de un honeypot físico ha de tenerse en cuenta, debido a la necesidad de disponer de hardware y no solo de software. Además, también conlleva un mantenimiento mayor.

Los sistemas de alta interacción, la gran mayoría de veces, son sistemas físicos.

5.2.2. Virtual

Un honeypot virtual es básicamente un sistema real pero ejecutado sobre una máquina virtual. La virtualización ofrece la ventaja de que es posible simular más tipos diferentes de dispositivos en un mismo equipo físico. El número y tipo de servicios ofrecidos dependerá de la implementación que haya hecho el desarrollador del honeypot virtual, así como su tipo de interacción.

El mantenimiento de un honeypot virtual dependerá del número de servicios ofrecidos y de la interacción mostrada. Si se dispone de servicios de funcionamiento completo, disponiendo de un honeypot de alta interacción, su mantenimiento será similar a un equipo físico.

Por norma general, los honeypot virtualizados suelen ser de baja interacción.

	Honeypot Virtual	Honeypot Físico
Ventajas	Desplegar gran número de honeypots de forma sencilla. Escalabilidad y fácil mantenimiento. Baratos. Puesta en marcha rápida y sencilla. Ideales para honeypots de baja interacción	Maquina real en la red. Mayor realismo Mayor dificultad para identificar como honeypot Ideales para honeypots de alta interacción
Desventajas	Es más fácilmente detectable por los atacantes. Dificultad para simular sistemas complejos o amplios. Recopilan menos información.	Precio elevado No son prácticos para lugares con muchas direcciones IP.

Tabla 2 Comparativa de honeypots físicos y virtuales

5.3. Tipo de comportamiento

Aunque un honeypot podría presentar diferentes comportamientos, eso podría facilitar su detección, por lo que en la práctica solamente suelen implementar un único tipo de comportamiento.

En función de los mecanismos implementados en el honeypot para comportarse frente a distintas amenazas, pueden clasificarse y ayudar a prevenir ataques de distintas maneras.

5.3.1. Métodos de detección precisa

La ventaja de un honeypot para la detección precisa es que se reducen los falsos positivos, haciendo posible capturar información esencial sobre las nuevas técnicas o herramientas de explotación de vulnerabilidades y siendo capaz de trabajar con comunicaciones cifradas y bajo redes IPv6.

5.3.2. Protección contra intrusos

La idea de esta contramedida consiste en intentar confundir a los atacantes, con el objetivo de hacerles perder tiempo y aprovecharlo para detectar las actividades que está realizando el atacante para tomar las contramedidas necesarias para bloquearlo, consiguiendo que el atacante nunca llegue al objetivo.

5.3.3. Desactivar acciones

Durante este proceso, las acciones del atacante son permitidas, pero finalmente son desactivadas para que no pueda aprovechar dichas vulnerabilidades. El atacante consigue llegar a su objetivo, pero se manipulan las peticiones para que fallen.

5.3.4. Ralentización o defensa contra ataques automatizados

Con este tipo de honeypots, el atacante es ralentizado en todas sus acciones maliciosas. Si un atacante encuentra algún aspecto vulnerable, sus técnicas de ataque serán ralentizadas por la solución. Esto se logra mediante modificaciones o alteraciones en los paquetes de red, modificando parámetros como el “Tamaño de ventana” o “Windows Size” fijándolo a cero o dejando al atacante en espera.

Este tipo de técnica es perfecta para prevenir la velocidad de propagación de gusanos que hayan afectado al honeypot o a la red.

5.3.5. Ninguno

En este caso, el honeypot no realiza ninguna acción o contramedida respecto a las acciones de los atacantes, por lo que no existe limitación del alcance o daños que pueda generar.

5.4. Tipo de rol

En la actualidad, también se pueden diferenciar los honeypot según el tipo de rol que desempeñan. Los roles que puede ejercer son de servidor, es decir, el honeypot recibe los ataques, o con el rol de cliente, el cual realiza peticiones contra servidores o aplicaciones maliciosas.

5.4.1. Cliente

Un honeypot con el rol de cliente sirve para imitar a un software que utiliza los servicios en el servidor. Uno de los ejemplos más clásicos consiste en tener un navegador que va

visitando diferentes páginas web con el objetivo de que estas le ataquen aprovechando alguna vulnerabilidad. Se basa en ir visitando webs e ir recopilando información acerca de los ataques y riesgos de seguridad.

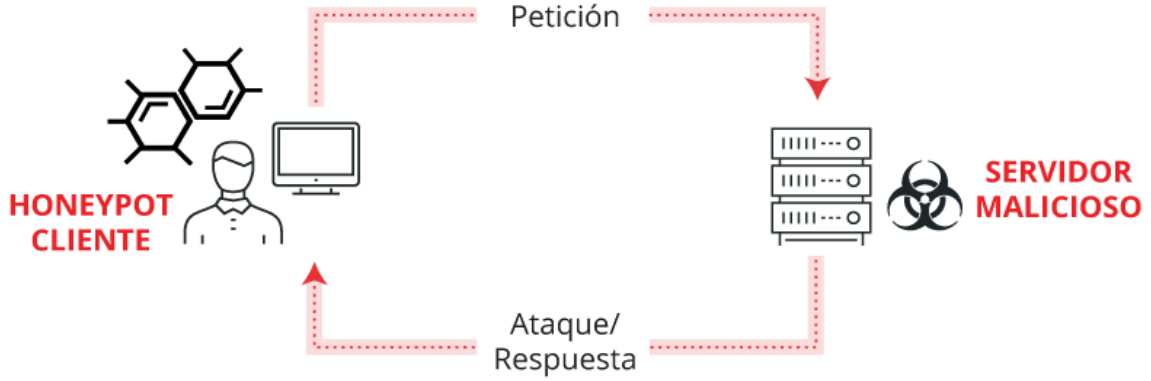


Figura 5 Honeypot con rol cliente

5.4.2. Servidor

El funcionamiento de un honeypot con el rol de servidor consiste en atraer a los atacantes hacia un entorno seguro o aislado para poder realizar estudios de investigación o alejar a posibles atacantes de la red real.

Se basa en la simulación de un entorno lo más realista y creíble posible, así a los atacantes no les será tan sencillo distinguir este tipo de red de una red real. Para atraer su atención, se suelen simular aplicaciones, servicios o dispositivos de automatización industrial, intentando captar la atención de un ciberatacante.

Cuando algún atacante cae en la trampa se van registrando todas las acciones, herramientas y técnicas que va empleando para conseguir sus objetivos, permitiendo a los administradores o investigadores obtener nueva información que permite una mejor protección y conocimiento frente a futuros atacantes.

Figura 6 Honeypot con rol servidor

6. Herramientas

Los honeypot han ido evolucionando desde su creación hasta hoy, y lo seguirán haciendo en el futuro, utilizando nuevas herramientas y técnicas. Si bien su misión principal no ha cambiado, si lo ha hecho el tipo de información que se busca y la forma de recopilarla

Actualmente, es posible utilizar un gran abanico de herramientas, algunas de ellas son HoneyMonkey, BeEF y HoneyBadger, orientadas a la extracción de información de los atacantes, o Spider trap, Weblabyrinth y Nova para ralentizar las acciones del atacante mediante engaños y trampas.

6.1. HoneyMonkey

Un HoneyMonkey es un tipo de herramienta especial creada por Microsoft Research que, usando una red de ordenadores o máquinas virtuales, es capaz de recibir y analizar ataques visitando páginas web sospechosas. Su misión principal es detectar nuevos tipos o patrones de ataque y fórmulas de infección que aprovechen vulnerabilidades de navegadores.

Algunas de las principales características de este proyecto son las siguientes:

- Está formado por un módulo de “exploración” y otro de recogida de datos.
  - La exploración activa, consiste en ir explorando de forma automática, por medio de agentes, un listado de páginas web, recopilando información acerca de posibles ataques.
- Funciona como un sistema automático de navegación.
  - Visita toda clase de páginas web con el fin de que alguna de ellas intente aprovechar vulnerabilidades en el navegador.
  - Es un tipo de honeypot con rol de cliente.
- Los navegadores pueden ser configurados.
  - Estar al día con las últimas actualizaciones o ejecutarse sin alguna actualización específica, con el fin de que algún sitio web explote o aproveche alguna vulnerabilidad.
- El funcionamiento de HoneyMonkey se basa en snapshots.
  - Se toma un snapshot del registro, ejecutables y memoria antes de visitar las páginas.
  - Después de visitar las páginas con el malware se toma otro snapshot y se comparan ambos para ver los efectos que ha generado y qué vulnerabilidad ha explotado.

HoneyMonkey se ejecuta sobre Windows XP con varios niveles de actualizaciones o parches, algunos de ellos con todo parcheado, otros con algunas vulnerabilidades y finalmente sin ningún tipo de parche o actualización.

Una vez están configuradas las máquinas y los navegadores, hay que llevar a cabo la recolección y el análisis, lo cual se compone de las siguientes fases:

- Fase de *crawling*. Se elabora una lista previa con las páginas web clasificadas como potencialmente peligrosas, es decir, aquellas webs de las que van a intentar

aprovechar vulnerabilidades en nuestro navegador para explotarlas. El sistema aumenta el tamaño de la lista añadiendo los enlaces externos que va encontrando en cada web, debido a que es probable que dichas webs también sean maliciosas.

- Análisis de exploits. HoneyMonkey utiliza el sistema de caja negra para la detección de vulnerabilidades. Cada máquina virtual ejecuta Internet Explorer para ir visitando cada página de la lista. Durante el proceso se van anotando todos los datos del registro, operaciones de lectura y escritura.
- Detección de malware. Todos los cambios que se realizan en los archivos que estén fuera de la carpeta temporal del navegador habrán aprovechado alguna vulnerabilidad del navegador, debido a que no se aceptan, ni permiten *pop-ups,* ni instalación de software. Los ficheros se analizan mediante un programa de detección de malware y se revisan manualmente. Finalmente, se reinicia la máquina para volver a su estado inicial y continuar con el resto de las páginas.

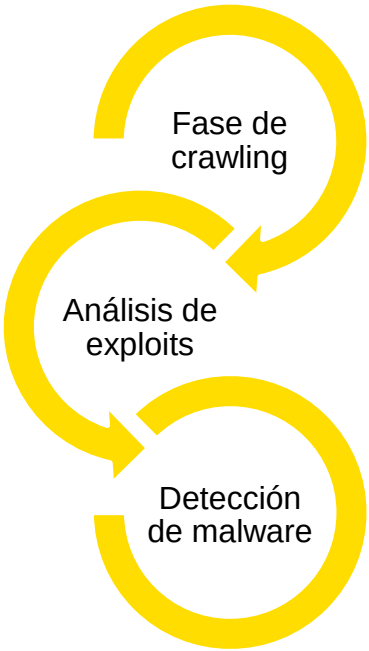


Figura 7 Fases de ejecución de HoneyMonkey

6.2. BeEF

BeEF (*Browser Exploitation Framework*) es una herramienta para pruebas de penetración que se centra en el navegador web, su funcionamiento y recogida de datos y se basa en la ejecución de scripts en el equipo del atacante. Durante el ataque, se encuentra embebido dentro de una página web segura y va recogiendo información útil de seguridad para los analistas. Algunas de los problemas encontrados se deben a la dificultad en el despliegue y desarrollo, y en la experiencia de los atacantes de cara a descubrir honeypots.

Figura 8 Logotipo de BeEF

Cuando BeEF está activo y funcionando va extrayendo información del navegador. La información que puede llegar a obtener es:

- El nombre y la versión del navegador,
- Los plugins utilizados (Java, ActiveX, VBS, Flash, etc),
- El tamaño de la ventana,
- El User Agent (UA).

Además de estos datos, gracias a la utilización de una serie de plugins, se pueden extraer más detalles acerca de la máquina del atacante. Entre esta información obtenida, se encuentra:

- Detalles de la máquina virtual Java (JVM),
- Cantidad de memoria,
- Detalles del sistema operativo,
- Número de procesadores,
- Tipo de tarjeta de red (NIC),
- Direcciones IP,
- Uso de TOR,
- Uso de redes sociales,
- Geolocalización.

6.3. HoneyBadger

HoneyBadger es un framework utilizado para la ayuda en la detección y geolocalización de los atacantes. Su uso combinado con la herramienta Molehunt, permite al usuario crear documentos que al ser abiertos informan sobre ello, ayudando a identificar y geolocalizar al atacante.

Figura 9 Logotipo de HoneyBadger

Su funcionamiento se basa en ofrecer a los atacantes las funciones administrativas que quieran controlar. Según los creadores, puede funcionar en forma de ActiveX o Java applets, haciendo creer al atacante, una vez ejecutados, que ha conseguido vulnerar la página web.

HoneyBadger emplea el análisis del flujo TCP para detectar e identificar ataques, combinando una variedad de ataques de inyección TCP para asegurar que es un ataque real y evitar así los falsos positivos.

El resultado se concluye en la geolocalización del atacante con un margen de error de unos 20 metros. Su funcionamiento es similar a la tecnología que utilizan los smartphones para la geolocalización, mediante posicionamiento basado en triangulación.

6.4. Spidertrap

Un spidertrap consiste en un honeypot destinado a ralentizar e impedir la tarea de los crawler utilizados para indexar páginas web.

Su funcionamiento se basa en la creación de bucles o páginas web anidadas de manera que el servicio crawler se vea atrapado y no pueda proseguir con su tarea, pudiendo llegar incluso a fallar si no está correctamente construido.

6.5. Weblabyrinth

Al igual que la herramienta anterior, esta también está destinada al entorno web y no a la creación de un honeypot físico, completo y funcional.

Weblabyrinth está pensada para crear, como su nombre indica, un laberinto de páginas web para tratar de confundir a los crawler. La gran mayoría de las páginas serán falsas, de manera que retrasan las tareas de posibles crawlers maliciosos en su tarea de búsqueda de información sobre la web.

6.6. Network Obfuscation and Virtualized Anti-Reconnaissence

Esta herramienta, denominada habitualmente por sus siglas, NOVA, permite crear una honeynet utilizando una versión mejorada de honeyd para crear el honeypot. Su operación se basa su en reglas básicas de IP origen, destino, puerto, tamaño paquete etc.

Figura 10 Logotipo de NOVA

Dispone de un aprendizaje automatizado que permite la activación de alertas cuando detecta actividades sospechosas o intentos de acceso fuero de lo habitual. Para ralentizar los ataques proporciona datos falsos a los atacantes, protegiendo los sistemas internos.

Mediante un panel web muy amigable permite la configuración y la revisión de información de los honeypot generados.

7. Proyectos

A lo largo de los años, han ido surgiendo proyectos relacionados con honeypots industriales para la protección y análisis de los atacantes. Algunos de los proyectos más destacados y populares son The Honeynet Project, Conpot/GasPot y Honeyd, que veremos a continuación.

7.1. The Honeynet Project

El proyecto, fundado en 1999, corresponde a una organización que se dedica a investigar los ataques más recientes, diseñando herramientas de código abierto para mejorar la seguridad en Internet y aprender cómo actúan los atacantes o ciberdelincuentes.

Figura 11 Logotipo de The Honeynet Project

Sus principales objetivos y metas son:

- Proporcionar herramientas y técnicas utilizadas por “The Honeynet Project” para que el resto de las organizaciones puedan beneficiarse de su uso. Algunas de las herramientas son Cuckoo, Capture-HPC, Glastopf, HoneyC, Honeyd y Honeywall.
- Crear conciencia acerca de todos los peligros y amenazas existentes en Internet.
- Llevar a cabo investigaciones de análisis de datos, desarrollo de herramientas únicas de seguridad y recopilar y analizar información acerca de atacantes y software malicioso que utilizan.

7.2. Honeynet SCADA de DigitalBond

Honeynet SCADA es un proyecto cuyo objetivo principal es construir un software factible para la simulación de una variedad de dispositivos industriales, como arquitecturas PLC, SCADA o DCS.

Mediante un sistema Linux se puede simular múltiples dispositivos y redes industriales, para su implementación también se hace uso del proyecto Honeyd, el cual permite crear diferentes máquinas virtuales y simular sus servicios como sistema operativo.

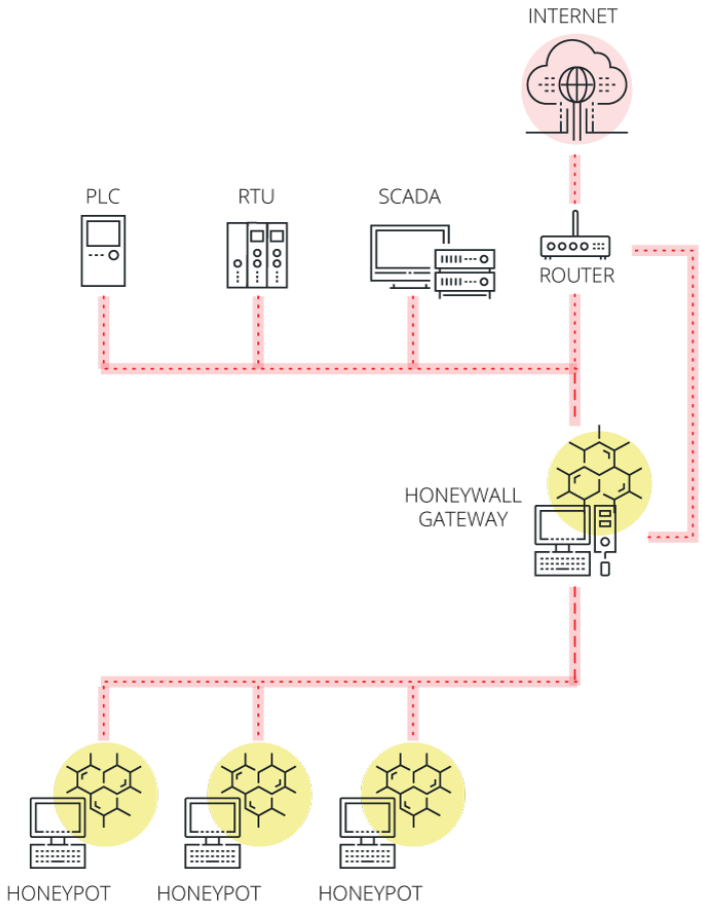


Figura 12 Propuesta de arquitectura para una honeynet industrial

El proyecto Honeynet SCADA está formado por algunos de los siguientes componentes:

- Monitorización
  - Uso de un Honeywall de 3ª generación
  - Firmas Quickdraw IDS
- Objetivo
  - Creación de PLC virtual o utilización de uno físico
- Servicios disponibles
  - HTTP: Porción de una web de administración de un PLC Schneider
  - FTP: Porción de servicio de FTP de uso administrativo
  - Telnet: Servicio Telnet de poca interacción
  - Modbus TCP: Puntos realistas programados en la honeynet. Sus valores cambian pseudo-aleatoriamente
  - SNMP: Servicio SNMP implementado, usa información del PLC
  - VxWorks Debugger: Simula el servicio escuchando en el puerto UDP/17185

7.3. Conpot y GasPot

Conpot es un honeypot que ha sido diseñado en base a sistemas de control industrial de baja interacción. Algunas de sus ventajas son que es fácil de desplegar, modificar y ampliar. Su facilidad de uso y la capacidad de simular una amplia gama de protocolos y sistemas

hace que pueda implementar casi cualquier sistema. Actualmente el proyecto se encuentra integrado dentro de "*The Honeynet Project*". Simula por defecto un PLC Siemens SIMATIC S7-200 y los protocolos Modbus, SNMP y HTTP.

Por otra parte, GasPot ha sido diseñado para simular un medidor de un tanque de líquidos, correspondiente al modelo *Guardian AST* de Veeder Root, el cual es un dispositivo diseñado para el control y cumplimiento de inventario en tanques de almacenaje. Sus funciones principales son el monitoreo de los niveles de las bombas, los sistemas de bombeo y el inventario de los tanques.

Los sectores que más utilizan este tipo de medidores son las empresas pertenecientes a la industria petrolera, ya que sirven para medir el nivel de combustible de los tanques. Es un proyecto abierto (*open source*) que puede ser implementado e instalado por cualquier usuario, descargándolo desde su repositorio en GitHub.[3]

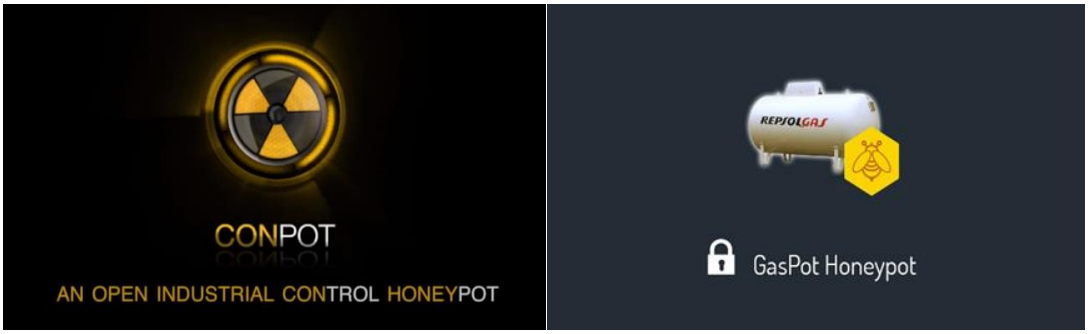


Figura 13 Logotipos de Conpot y GasPot

7.4. Honeyd

Honeyd consiste en un software creado por Niels Provos que permite la creación de múltiples honeypots y su ejecución de forma virtual en la red. Estas máquinas se pueden configurar para simular diferentes tipos de componentes, sistemas operativos o servicios, con el objetivo de extraer información útil de los atacantes.

Los dos objetivos principales de este software son la distracción de los atacantes y su función como honeypot. El objetivo de la distracción es mantener a los atacantes enfocados en él, retrasando y ralentizando sus acciones. El uso del honeypot está más orientado al estudio e investigación de las formas de ataque empleadas.

En el apartado 8 Despliegue de un honeypot industrial, se muestra cómo construir un honeypot utilizando este software.

[3] https://github.com/sjhilt/GasPot

8. Despliegue de un honeypot industrial

En este apartado se recogen instrucciones precisas paso a paso de cómo desplegar un honeypot industrial. Para la elaboración de este honeypot se va a utilizar la herramienta Honeyd, que permite la creación de una máquina virtual, con los servicios y características de red que más convengan, y los scripts desarrollados por SCADA HoneyNet Project, puesto que incluyen servicios para emular un PLC de Siemens o de Schneider Electric.

Para esta guía se ha optado por el dispositivo Siemens Simatic S7-300, dado que es uno de los más utilizados en el mundo industrial.

8.1. Herramientas empleadas

A continuación, se listan y describen las herramientas utilizadas para configurar el honeypot.

Herramienta software	Especificaciones
Sistema operativo base	Ubuntu Desktop 19.04 - 64 bits[4]
Honeyd	1.6d[5]
Python	2.7.16
Bash	5.0.3
Perl	5
Scripts SCADA HoneyNet Project[6]	honeyd-ftp-siemens.py honeyd-http-siemens.py honeyd-telnet-siemens.py honeyd-s7.py honeyd-snmp-siemens.py honeyd-modbus.py

Tabla 3 Listado de las herramientas empleadas

8.2. Diseño del honeypot

En esta guía se ha optado por la instalación de un sistema operativo base Ubuntu Desktop 19.04 de 64 bits, sobre el que se instalará la herramienta honeyd para poder desplegar posteriormente los scripts de SCADA HoneyNet Project.

Los scripts que se utilizarán para simular los servicios en el honeypot son los siguientes:

- **honeyd-ftp-siemens.py:** FTP en el puerto TCP 21. Se simulará el servidor FTP del modelo CP 343-1 IT. No es posible hacer *login* en la simulación.

[4] https://ubuntu.com/download/desktop

[5] https://github.com/DataSoft/Honeyd

[6] https://sourceforge.net/projects/scadahoneynet/

- **honeyd-telnet-siemens.py:** Telnet en el puerto TCP 23. Tampoco es posible hacer *login* en la simulación.
- **honeyd-s7.py:** S7 en el puerto 102. Dispondrá de una simulación básica del servidor S7 interno (protocolo de control propietario de Siemens) del CP 343-1 IT.
- **honeyd-http-siemens.py:** HTTP en el puerto TCP 80. Dispondrá de una simulación básica del *frontend* propio de un CP 343-1 IT.
- **honeyd-snmp-siemens.py:** SNMP en el puerto UDP 161.
- **honeyd-modbus.py:** MODBUS en el puerto 502. Contará con una simulación básica de un servidor Modbus, con respuesta para varios códigos de paquetes y peticiones.

8.3. Bastionado del sistema operativo anfitrión

Debido a que esta máquina estará expuesta a potenciales ataques, es conveniente bastionar el sistema operativo base para que no sea vulnerable y toda la atención se fije en los servicios simulados por Honeyd y no en servicios del propio Linux. Por ello, se parte de un sistema ya bastionado con las siguientes medidas de seguridad adoptadas:

- Configuración del firewall mediante iptables.
- Cambiar y reforzar la contraseña de usuario.
- Desactivar el usuario root.
- Desactivar servicios innecesarios.
- Actualizaciones del sistema.
- Reforzar el acceso mediante SSH.

8.4. Instalación Honeyd

La instalación de Honeyd se realizará mediante la descarga de las fuentes, instalando las dependencias y compilando el programa.

8.4.1. Descarga de Honeyd

La última versión disponible de Honeyd se encuentra en el repositorio de GitHub[7]. La manera más sencilla de descargarla es la siguiente:

```
sudo apt-get install git
git clone https://github.com/DataSoft/Honeyd
```

Figura 14 Comandos de instalación de Git y descarga de Honeyd

8.4.2. Instalación de dependencias

Honeyd depende de las siguientes librerías:

- **libevent:** API que proporciona un mecanismo para ejecutar una función de devolución de llamada cuando ocurre un evento específico en un descriptor de archivo o después de que se haya alcanzado un tiempo de espera.
- **libdumbnet:** provee una interfaz simplificada y portable para varias rutinas de red de bajo nivel.
- **libpcap:** se trata de una librería de captura de paquetes por línea de comandos.

[7] https://github.com/DataSoft/Honeyd

- **libpcre:** librería de C de expresiones regulares inspirada por Perl.
- **libedit:** programa que permite la edición de ficheros en línea de comandos.
- **bison y flex:** generadores de analizadores sintácticos de propósito general.
- **zlib:** librería de software utilizada para compresión de datos.
- **python:** lenguaje de programación necesario para la ejecución de los scripts del proyecto SCADA HoneyNet.

El comando a ejecutar para llevar a cabo la instalación de las mismas será el siguiente:

```
sudo apt-get install libevent-dev libdumbnet-dev libpcap-dev libpcre3-dev
libedit-dev bison flex libtool automake zlib1g-dev python
```

Figura 15 Comando para la instalación de las dependencias

8.4.3. Compilación e instalación de Honeyd

Una vez finalizados los pasos anteriores, se pasará a la compilación e instalación del programa. Dentro del directorio **Honeyd** descargado anteriormente:

```
./autogen.sh
./configure
make
sudo make install
```

Figura 16 Comandos para la instalación de Honeyd

Con el fin de tener los resultados y configuraciones en un mismo lugar, es conveniente crear una carpeta donde se incluirán los scripts de SCADA Honeynet Project que se descargarán posteriormente, y el archivo de configuración y de log de honeyd *honeyd.conf* y *honeyd.log.*

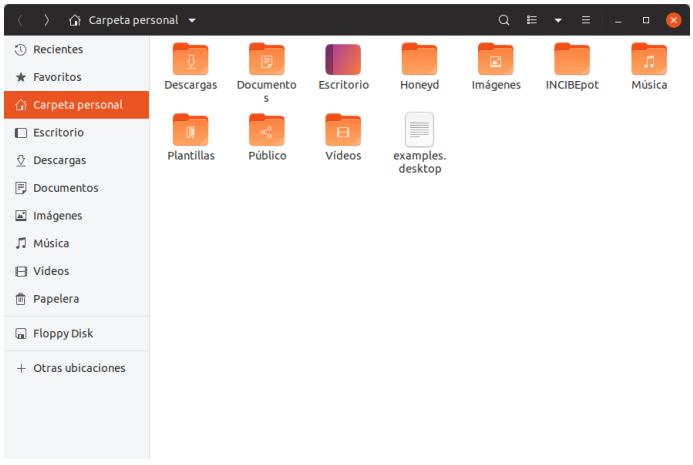


Figura 17 Creación de la carpeta

8.5. Configuración de Honeyd

Para una buena configuración del Honeyd, será necesario comprender los archivos localizados en **/usr/share/honeyd**, entre ellos:

- **config.sample:** fichero que contiene un ejemplo de la configuración de Honeyd.
- **nmap.assoc:** lista de *fingerprints* que pueden ser detectados por **nmap**.

- **nmap-os-db:** base de datos de *fingerprints* de **nmap**.
- **nmap-mac-prefixes:** lista de identificadores MAC de cada fabricante. Se utilizarán para darle más autenticidad a los escaneos por MAC. En este caso, se han utilizado los de la división de automatización de Siemens. Para emular otro dispositivo se podrá elegir otro prefijo.

El objetivo de dichos ficheros es dar una respuesta lo más realista posible del SO simulado cuando le están realizando un escaneo mediante nmap u otra herramienta similar.

Por otro lado, la simulación de los servicios se realiza mediante la ejecución de scripts como respuesta a la recepción de una petición a un puerto abierto de Honeyd. Esta herramienta dispone de una amplia gama de scripts que se encuentran en el directorio **/usr/share/honeyd/scripts.** Algunos de estos se encuentran en formato **bash** o **perl**, por lo que sería necesario tener instalados dichos intérpretes para que puedan ser ejecutados. Con esto se podrían simular una gran variedad de máquinas TI, como Linux, Windows o alguna versión embebida.

Sin embargo, Honeyd no dispone de scripts que simulen servicios propios de SCI, por lo que se han utilizado los scripts programados en Python del proyecto SCADA HoneyNet Project[8]. Existen scripts para servicios de dos PLC de diferentes marcas como, por ejemplo, Siemens y Schneider Electric.

Con el fin de facilitar el trabajo, se recomienda mover toda la carpeta **/cernscadahoneynet/files/scripts,** situada en la raíz del proyecto SCADA HoneyNet Project, a la carpeta que se indicó crear con anterioridad.

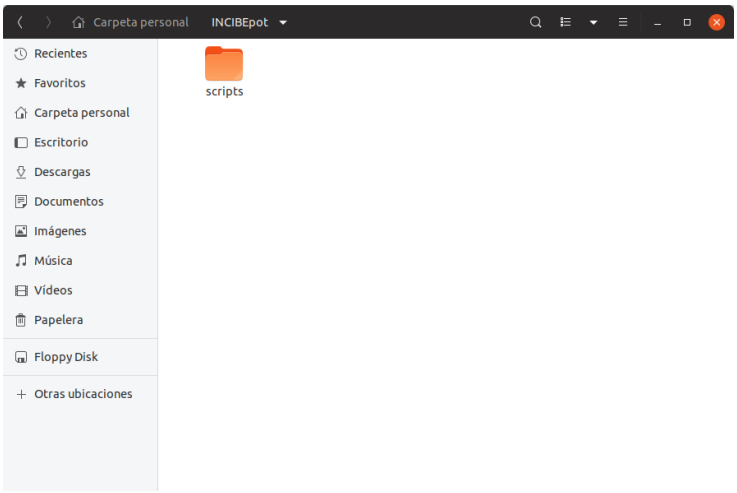


Figura 18 Estado de la carpeta en este punto

Antes de arrancar el honeypot, es necesario hacer dos modificaciones:

- Abrir el archivo **honeyd-http-siemens.py** y editar la variable **webroot**, incluyendo el *path* absoluto donde se encuentra la carpeta **web-siemens**, contigua al archivo editado.

[8] http://scadahoneynet.sourceforge.net/

```
webroot = "/home/incibepot/INCIBEpot/scripts/web-siemens"
```

Figura 19 Línea editada en el fichero "honeyd-http-siemens.py"

En el caso de que se quiera simular la web de Schneider, bastaría con modificar el directorio "web-siemens" por "web-schneider". Para mostrar cualquier otra web, se podría utilizar el script web.sh, localizado en la carpeta **/scripts/backdoor**, dentro de la instalación de honeyd.

- Puesto que no se dispone de un script que simule Telnet en un PLC Siemens, duplicar el archivo **honeyd-telnet-schneider.py** y renombrar la copia a **honeyd-telnet-siemens.py.** Es necesario modificar el fichero para que indique que es un equipo Siemens y no Schneider Electric.

```
logintext = "\n\rSiemens Login: "
```

Figura 20 Línea editada en el fichero "honeyd-telnet-siemens.py"

- Si no está ya incluido, extraer el nombre completo del SO a simular del archivo **nmap-os-db** (el nombre completo inmediatamente después de **Fingerprint** de producto deseado, en nuestro caso SIMATIC 300 PLC), que está situado en la ruta **/usr/share/honeyd/**, y añadirlo, editándolo como *root*, al final de la lista de **nmap.assoc**, seguido por un punto y coma (se encuentra en el mismo directorio que el anterior archivo). En el caso de que ya se encuentre en la lista, simplemente asegurarse de que dicha línea está sin comentar. En este caso, el SO a simular será **Siemens Simatic 300 programmable logic controller.**

```
# SIMATIC 300 PLC
Fingerprint Siemens Simatic 300 programmable logic controller
Class Siemens | embedded || specialized
```

Figura 21 Nombre completo del SO obtenido de nmap-os-db

```
#ZyXEL ZyWALL 2 Plus firewall;
#ZyXEL ZyWALL 5 firewall;
Siemens Simatic 300 programmable logic controller;
```

Figura 22 Nombre del SO añadido en la última línea de nmap.assoc

Posteriormente, se debe crear el archivo de configuración del honeypot, donde se incluirán todos los parámetros de la máquina a simular, como puertos abiertos, servicios que se emulan en ellos, información de red, como la dirección MAC o la IP, nombre del sistema operativo, entre otros. En este caso, se llamará **honeyd.conf** y se situará dentro de la carpeta previamente creada.

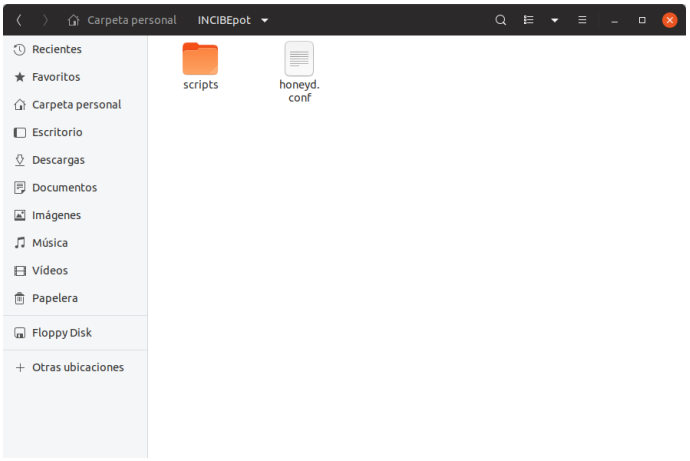


Figura 23 Carpeta previamente creada, con el archivo de configuración

En el fichero se debe introducir el siguiente contenido:

```
create siemens
set siemens ethernet "00:1f:f8:cc:d0:23" # Siemens Automation MAC ID
set siemens default tcp action closed
set siemens default udp action reset
set siemens personality "Siemens Simatic 300 programmable logic
controller"
add siemens tcp port 21 "python /home/incibepot/INCIBEpot/scripts/honeyd-
ftp-siemens.py"
add siemens tcp port 23 "python /home/incibepot/INCIBEpot/scripts/honeyd-
telnet-siemens.py"
add siemens tcp port 80 "python /home/incibepot/INCIBEpot/scripts/honeyd-
http-siemens.py"
add siemens tcp port 102 "python
/home/incibepot/INCIBEpot/scripts/honeyd-s7.py"
add siemens udp port 161 "python
/home/incibepot/INCIBEpot/scripts/honeyd-snmp-siemens.py"
add siemens tcp port 502 "python
/home/incibepot/INCIBEpot/scripts/honeyd-modbus.py"
set siemens uptime 4532786 # 52 días encendido.
bind <IP_honeypot> siemens # En nuestro caso de uso, la IP es
192.168.252.130
```

Figura 24 Contenido del archivo de configuración honeyd.conf

Donde cada comando y parámetro introducido significan:

- **create <name>:** define una plantilla para un SO simulado.
- **set <template> ethernet "MAC":** se define la MAC del *honeypot.* El listado de MACs correspondiente a cada fabricante se encuentra en el fichero **nmap-mac-prefixes** previamente mencionado.
- **set <template> personality "<name>":** define el nombre del SO a emular, el cual será cotejado con la base de datos de **nmap** y **nmap.assoc** para dar respuesta al escaneo de SO de **nmap** o **xprobe2**. Debe ser idéntico al incluido en el archivo **nmap.assoc** en pasos previos.
- **set default <portType> action <actionName>:** se indica a Honeyd cómo tratar las conexiones no definidas en los puertos específicos.

- **add <template> <portType> port <#> "<script>":** se realiza la apertura y asignación de puertos específicos, especificando la ruta a un script que se ejecutará escuchando en dicho puerto, para emular el servicio en cuestión.
- **set <template> uptime <timestamp>:** establece artificialmente el tiempo que ha pasado desde el último inicio del dispositivo. En algunos sistemas operativos se puede extraer del **timestamp** de TCP, por lo que se podría averiguar el tiempo que lleva iniciado mediante este método.
- **bind <ipaddress> <template>:** se asigna una dirección IP al SO simulado. Dicha IP debe estar en la misma red que la interfaz del host en el que escuchará el *honeyd.* Para la elaboración de esta guía, se ha empleado la IP 192.168.252.130.

Una vez que ya hemos movido los scripts y completado el fichero de configuración, se pasará a ejecutar el *honeypot*. Para ello, se ejecutará el siguiente comando en la raíz de la carpeta creada al principio, por ejemplo:

```
sudo honeyd -d -p nmap-os-db -i ens38 -l honeyd.log -f honeyd.conf IP -u
0 -g 0 --disable-webserver
```

Figura 25 Comando y parámetros para ejecutar la máquina virtual de Honeyd

Donde:

- **-d:** ejecuta Honeyd sin demonizar y con mensajes de depuración.
- **-p file:** permite a Honeyd leer *fingerprints* de **nmap** contenidos en dicho archivo.
- **-i interface**: especifica la interfaz de red del host que ocupará el honeypot.
- **-l logfile:** especifica el archivo de registro. De esta manera, habrá dos logs: uno de **iptables** y otro de Honeyd.
- **-f config:** especifica el archivo de configuración específico que Honeyd ejecutará.
- **IP:** dirección IP de la máquina virtual emulada o subred en la que están contenidas varias máquinas emuladas (en el caso de haber más de una). Esta será la red que Honeyd emulará. Debe ser la misma red que la del interfaz del *host* especificado en el comando con el argumento -i.
- **-u:** establece el UID con el que Honeyd está funcionando.
- **-g:** establece el GID con el que Honeyd está funcionando.
- **--disable-webserver:** desactiva el servidor web que honeyd tiene configurado por defecto.

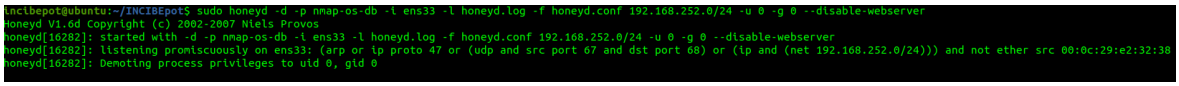


Figura 26 Resultado de ejecutar el comando de honeyd

Se ha elaborado un cheatsheet con todos los comandos resumidos en la página 43.

8.6. Mantenimiento del honeypot

Un honeypot desplegado mediante honeyd consiste en la virtualización de un host funcionando dentro de una máquina. Por lo tanto, sus tareas de mantenimiento serán similares a las que se podrían aplicar a cualquier dispositivo que tenga servicios levantados disponibles. Entre estas tareas se encuentran:

- Verificar el funcionamiento de los servicios del honeypot.

- Comprobar que la máquina host funciona correctamente y está activa.
- Verificar la conectividad del honeypot con el resto de la red.
- Asegurarse de que las medidas de seguridad aplicadas al host son correctas.
- Revisar periódicamente los logs generados tanto por honeyd como por el firewall del host.
- Se podría arrancar Honeyd de forma automática mediante un script, útil en el caso de que ocurra un fallo en la red eléctrica o la máquina anfitriona se apague o reinicie.

8.7. Resultados

En este apartado se comprobará el correcto funcionamiento del honeypot mediante el uso de un conjunto de herramientas con finalidades de análisis de red y dispositivos, así como la interpretación de los logs generados por el honeypot.

8.7.1. Herramientas empleadas

Para la realización de las pruebas sobre la máquina emulada con Honeyd, se utilizarán varias herramientas que permitirán analizar los servicios simulados y la fidelidad de sus respuestas ante diferentes peticiones en su tarea de aparentar ser, en este caso, un PLC de Siemens. A continuación, se listan las herramientas empleadas:

Herramienta software	Especificaciones
Zenmap - Linux[9]	7.70
ModbusTool - Windows[10]	1.0
Wireshark - Windows[11]	3.0.5
Snap7[12]	1.4.0

Tabla 4 Listado de herramientas utilizadas en la fase de pruebas

8.7.2. Comprobación de servicios

- **Escaneo con nmap**

El programa utilizado para el escaneo de puertos y servicios es Zenmap, versión de nmap con interfaz gráfica. El resultado del escaneo muestra los puertos abiertos especificados en el archivo de configuración de honeyd, así como la información relativa al sistema operativo y el *fingerprint* obtenido.

[9] https://nmap.org/zenmap/

[10] http://www.modbusmaster.com/

[11] https://www.wireshark.org/#download

[12] http://snap7.sourceforge.net/

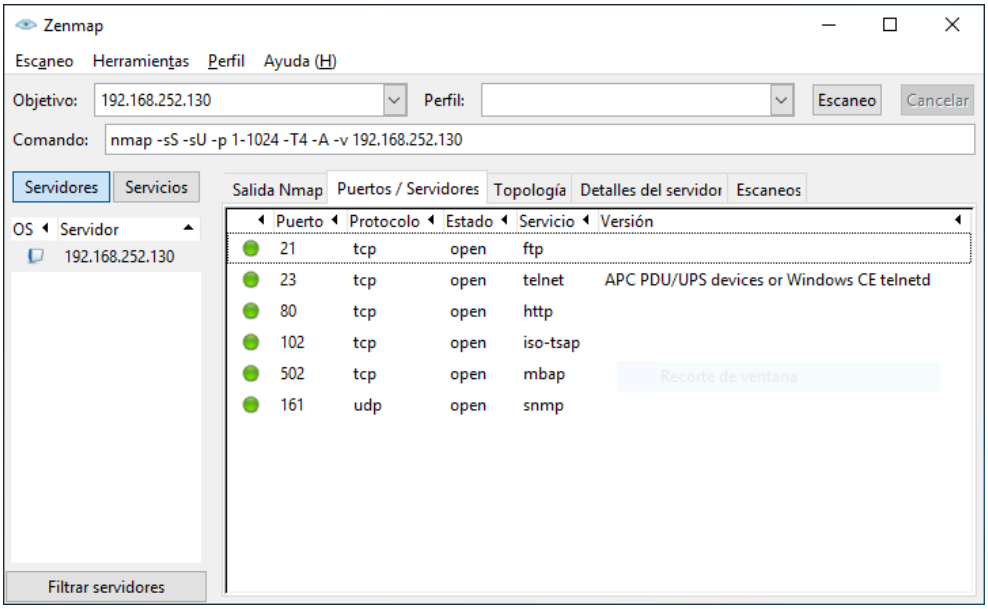


Figura 27 Resultado del escaneo de puertos en el honeypot

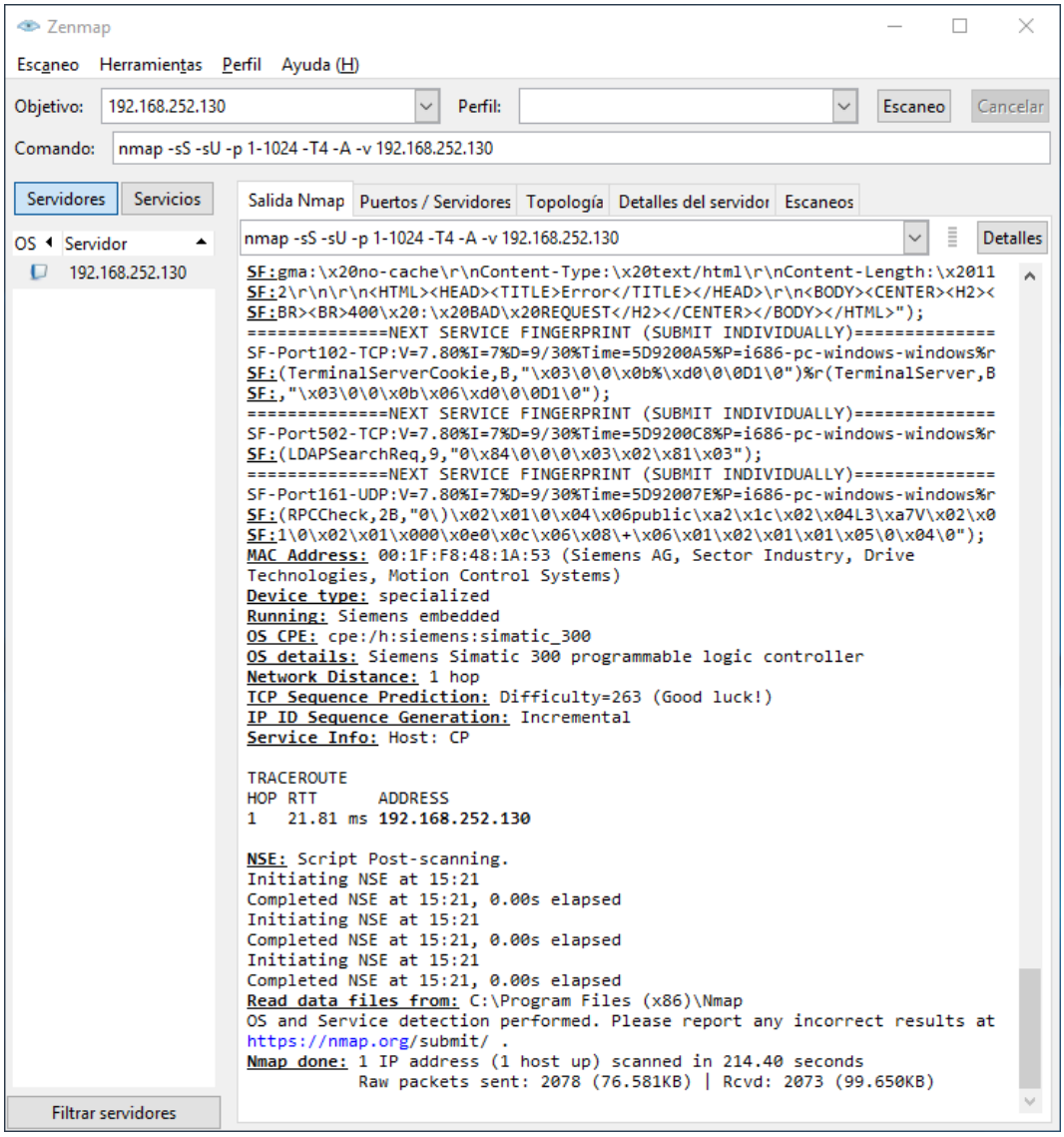


Figura 28 Resultado escaneo del SO del honeypot

- **HTTP**

Se ha realizado una conexión HTTP desde un navegador de una máquina dentro de la red de este con el fin de comprobar el funcionamiento del servicio web. Se muestra una página web similar a la que mostraría un PLC Siemens Simatic con el que se puede interactuar hasta cierto nivel, brindando algunos datos del dispositivo simulado.

Figura 29 Página web emulada en el honeypot

- **FTP**

Al hacer una conexión FTP contra el honeypot se espera que el servicio muestre el mensaje de login junto con sus diferentes respuestas, dependiendo de la entrada del usuario. En este caso, se obtienen los correspondientes códigos de error, tal y como haría un servicio FTP real. Cabe destacar que no es posible iniciar sesión en el servicio simulado.

```
incibepot@ubuntu:~$ ftp 192.168.252.130
Connected to 192.168.252.130.
220 CP 343-1 IT FTP-Server V1.36 ready for new user
Name (192.168.252.130:incibepot): admin
530 Not logged in.
Login failed.
Remote system type is CP.
ftp> exit
221 Closing control connection; Thank you for using our FTP server.
incibepot@ubuntu:~$
```

Figura 30 Resultado de la conexión FTP al honeypot

- **Modbus**

Para realizar esta prueba se ha utilizado el software ModbusTool para Windows, con el que se pueden enviar paquetes Modbus de escritura y lectura de registros. El honeypot responde a los mensajes Modbus TCP correctamente, tal y como lo haría un PLC con comunicación Modbus TCP.

Para la generación de mensajes Modbus se seguirán los siguientes pasos:

- Introducir la dirección IP del Honeypot en el apartado Connection de ModbusTool.

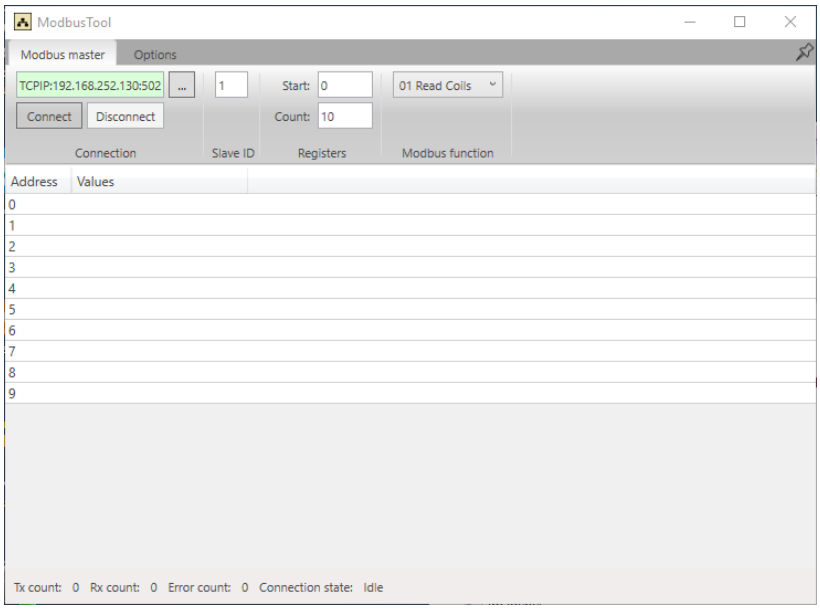


Figura 31 Estado ModbusTool en Idle

- Hacer click en Connect. El estado debería cambiar a Connected y se deberían poder ver el valor de los registros.

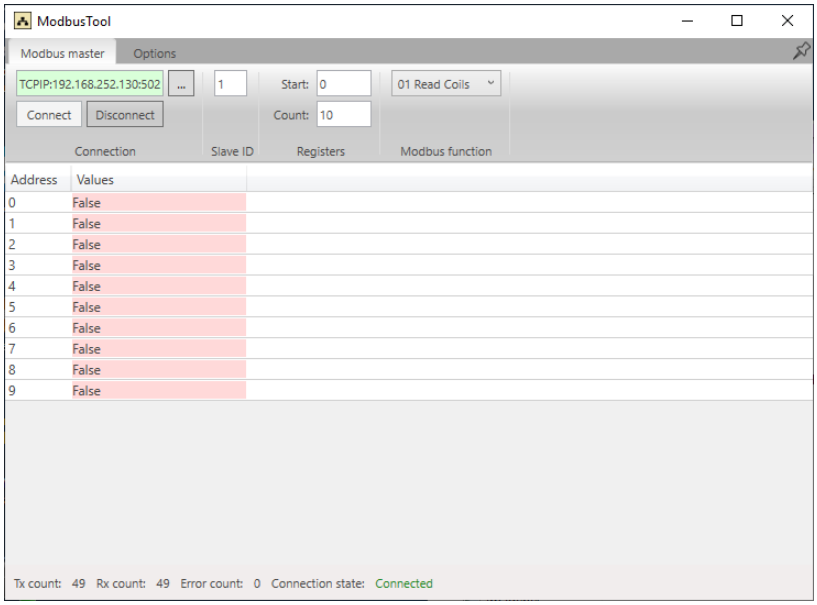


Figura 32 Estado ModbusTool en Connected

- La función de Modbus que realizará el programa por defecto es “01 Read Coils”. Si se quiere cambiar a alguna otra, se debe seleccionar la que se desee en el desplegable de la pestaña Modbus Function.

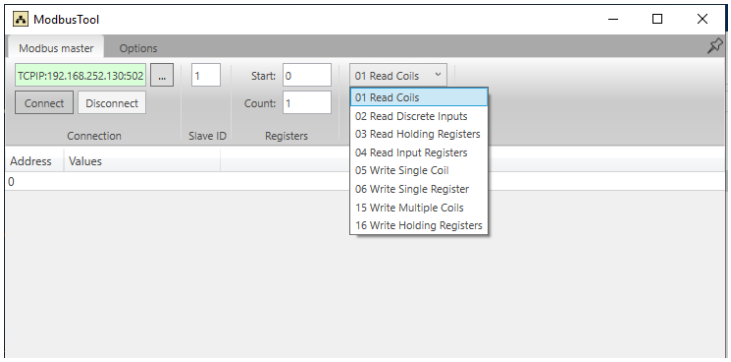


Figura 33 Desplegable de funciones ModbusTool

A continuación, se muestran las capturas de los paquetes enviados por el programa, las cuales se pueden localizar con el siguiente filtro de Wireshark: *tcp.port==502*.

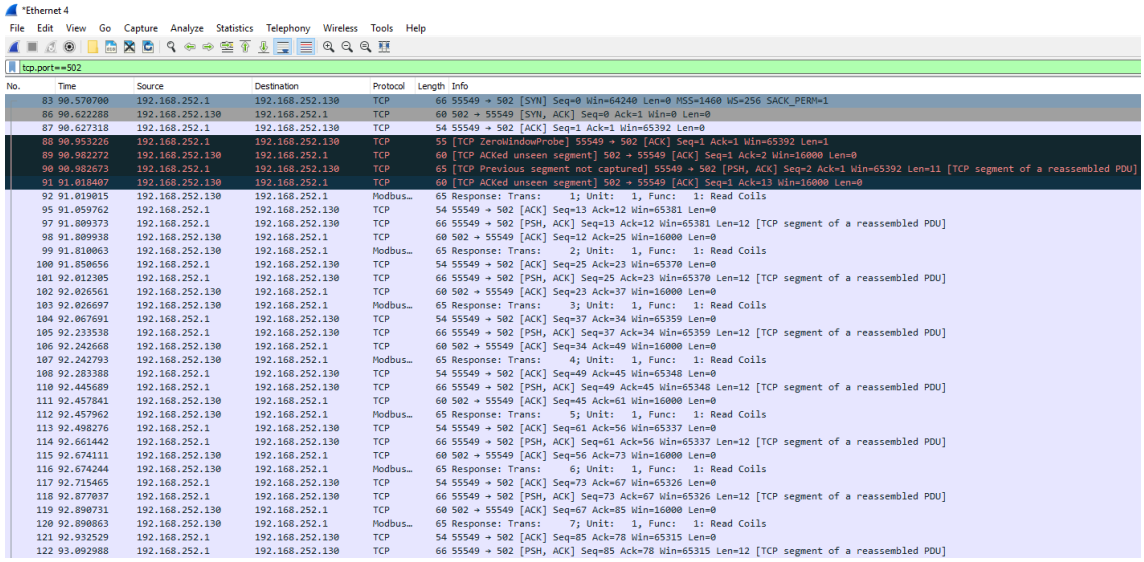


Figura 34 Intercambio de paquetes de lectura de registros entre ModbusTool y el honeypot

Figura 35 Intercambio de paquetes de escritura de un registro entre ModbusTool y el honeypot

- **S7**

También es posible enviar paquetes S7comm al honeypot. Para ello, se ha utilizado el cliente de ejemplo de la librería Snap7, **clientdemo.exe**, localizado en el directorio **\snap7-full-1.4.0\rich-demos\x86_64-win64\bin.** El honeypot responde a las peticiones tal y como lo haría un Siemens S7.

Para la generación de mensajes S7, se seguirán los siguientes pasos:

- Introducir la dirección IP del honeypot en el programa anteriormente mencionado.
- Hacer click en **Connect**.

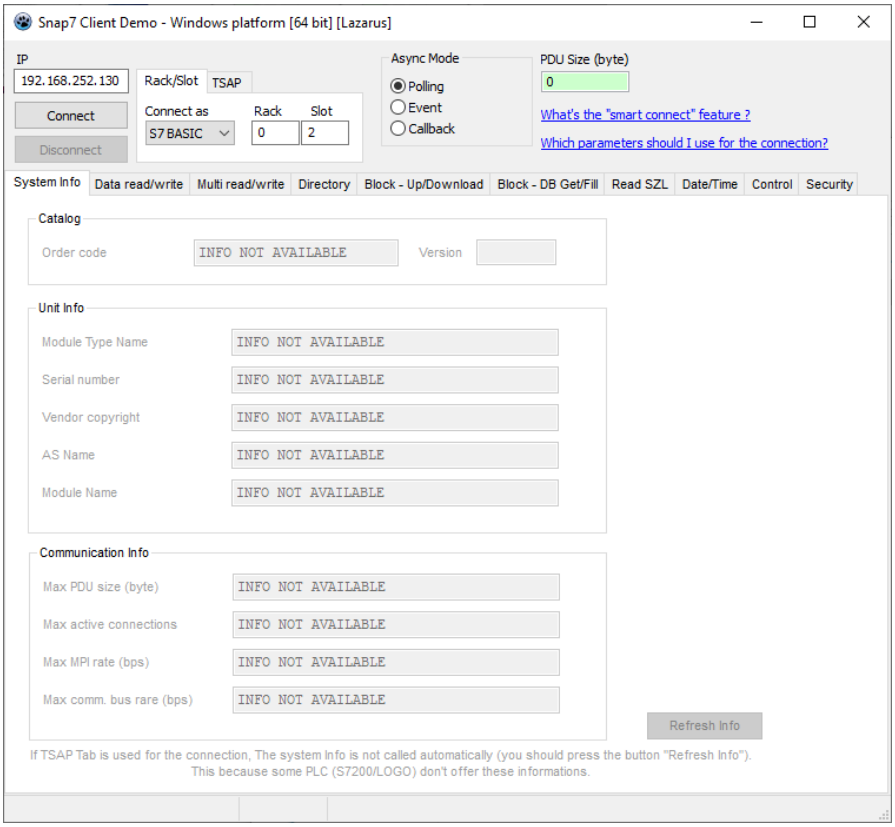


Figura 36 Captura de pantalla del cliente snap7

Este intercambio se puede comprobar utilizando el filtro *tcp.port == 102 && s7comm* en Wireshark:

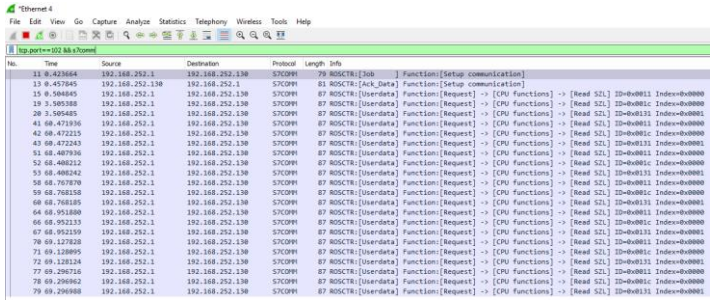


Figura 37 Intercambio de paquetes de conexión entre el cliente Snap7 y el honeypot

- **Telnet**

El resultado de realizar una conexión telnet con el honeypot es el esperado, obteniendo los mensajes de introducción de usuario y contraseña. Tal y como ocurre con el servicio FTP, no es posible iniciar sesión.

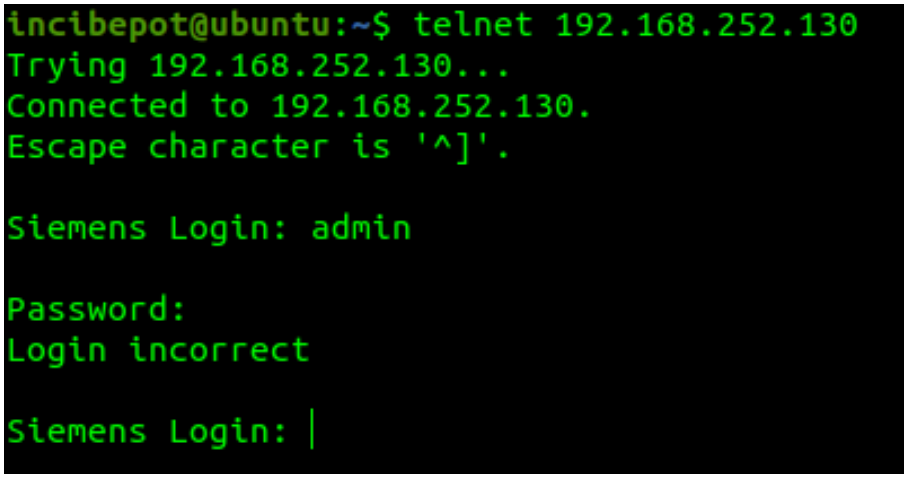


Figura 38 Resultado de un intento de acceso Telnet al honeypot

- **SNMP**

También es posible realizar un sondeo SNMP en el honeypot. La captura inferior muestra el resultado de realizar un snmpwalk contra su IP, confirmando el comportamiento esperado del servicio, pudiendo obtener información detallada sobre el hardware y software del dispositivo simulado.

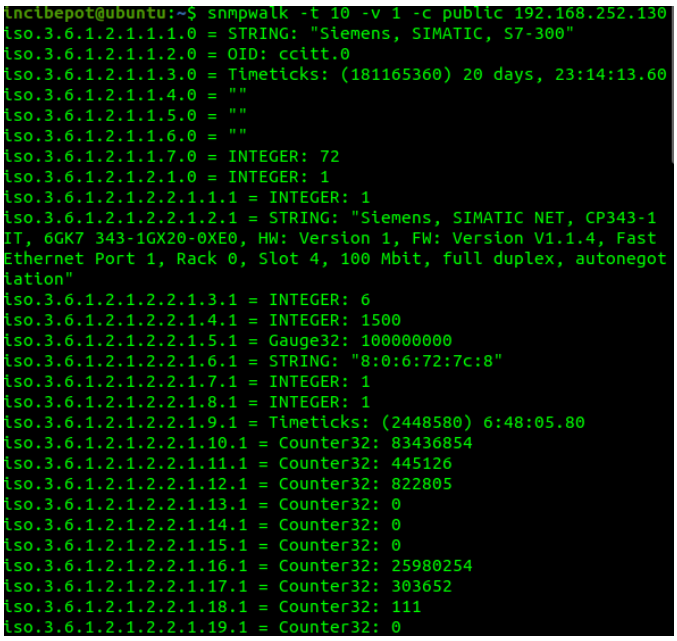


Figura 39 Resultado de ejecución del snmpwalk contra el honeypot

8.7.3. Análisis de Logs

A continuación, se mostrará el contenido generado por Honeyd en el log, a medida que se vayan produciendo intentos de conexiones hacia él. En cada entrada quedaría registrada la siguiente información:

- Fecha, hora, minuto y segundo en el que se ha producido la petición.
- Protocolo del paquete registrado.

- Tipo de conexión: “E” si es un cierre, “S” si es un inicio de conexión y “-“ si no pertenece a ninguna conexión.
- IP y puerto origen.
- IP y puerto destino.
- En algunos paquetes TCP que no forman parte de una conexión, Honeyd incluye el tamaño del paquete y los *flags* TCP.
- En ocasiones, existe una última columna con el sistema operativo de la máquina origen, obtenido mediante *fingerprinting* pasivo.

```
2019-09-30-04:38:16.5826 icmp(1) - 192.168.252.1 192.168.252.130: 8(0): 60
2019-09-30-04:38:17.5549 icmp(1) - 192.168.252.1 192.168.252.130: 8(0): 60
2019-09-30-04:38:18.5624 icmp(1) - 192.168.252.1 192.168.252.130: 8(0): 60
2019-09-30-04:38:19.5341 icmp(1) - 192.168.252.1 192.168.252.130: 8(0): 60
```

Figura 40 Formato log de peticiones ICMP

```
2019-09-30-06:37:56.6694 tcp(6) S 192.168.252.129 33610 192.168.252.130 21
2019-09-30-06:37:57.6413 tcp(6) E 192.168.252.129 33610 192.168.252.130 21: 0 0
2019-09-30-06:37:59.6580 tcp(6) S 192.168.252.129 33610 192.168.252.130 21
2019-09-30-06:38:35.3690 tcp(6) E 192.168.252.129 33610 192.168.252.130 21: 24 283
2019-09-30-06:39:03.3773 tcp(6) - 192.168.252.129 33610 192.168.252.130 21: 40 R
2019-09-30-06:39:31.8573 tcp(6) - 192.168.252.129 33610 192.168.252.130 21: 40 R
2019-09-30-06:39:59.8654 tcp(6) - 192.168.252.129 33610 192.168.252.130 21: 40 R
2019-09-30-06:40:19.2337 tcp(6) - 192.168.252.129 33610 192.168.252.130 21: 40 R
2019-09-30-06:40:27.8729 tcp(6) - 192.168.252.129 33610 192.168.252.130 21: 40 R
```

Figura 41 Formato log de peticiones FTP

```
2019-09-30-07:12:31.7646 tcp(6) S 192.168.252.129 57280 192.168.252.130 23
2019-09-30-07:14:10.8996 tcp(6) E 192.168.252.129 57280 192.168.252.130 23: 44 102
```

Figura 42 Conexiones Telnet

```
2019-09-30-07:14:53.6762 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.7000 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 40 67
2019-09-30-07:14:53.7119 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.7257 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 44
2019-09-30-07:14:53.7479 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.7589 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 47
2019-09-30-07:14:53.7846 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.7939 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 43
2019-09-30-07:14:53.8200 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.8297 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 43
2019-09-30-07:14:53.8560 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.8754 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 43
2019-09-30-07:14:53.8927 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.9364 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 44
2019-09-30-07:14:53.9648 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:53.9749 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 44
2019-09-30-07:14:54.0001 udp(17) S 192.168.252.129 58689 192.168.252.130 161
2019-09-30-07:14:54.0108 udp(17) E 192.168.252.129 58689 192.168.252.130 161: 43 46
```

Figura 43 Formato log de peticiones SNMP

```
2019-09-30-06:17:48.3266 tcp(6) S 192.168.252.1 55162 192.168.252.130 80 [Windows 2000 RFC1323]
2019-09-30-06:17:48.3626 tcp(6) E 192.168.252.1 55157 192.168.252.130 80: 12 0
2019-09-30-06:17:48.7947 tcp(6) E 192.168.252.1 55162 192.168.252.130 80: 53 226
2019-09-30-06:17:48.7948 tcp(6) S 192.168.252.1 55163 192.168.252.130 80 [Windows 2000 RFC1323]
2019-09-30-06:17:49.2626 tcp(6) S 192.168.252.1 55164 192.168.252.130 80 [Windows 2000 RFC1323]
```

Figura 44 Peticiones HTTP al honeypot

```
2019-09-30-06:19:17.5390 tcp(6) S 192.168.252.1 55233 192.168.252.130 502 [Windows 2000 RFC1323]
2019-09-30-06:19:17.5743 tcp(6) E 192.168.252.1 55230 192.168.252.130 502: 223 0
```

Figura 45 Conexiones Modbus

```
2019-09-30-06:19:57.5350 tcp(6) S 192.168.252.1 55250 192.168.252.130 102 [Windows 2000 RFC1323]
2019-09-30-06:19:57.5711 tcp(6) E 192.168.252.1 55249 192.168.252.130 102: 52 0
2019-09-30-06:20:02.5383 tcp(6) S 192.168.252.1 55251 192.168.252.130 102 [Windows 2000 RFC1323]
2019-09-30-06:20:02.5745 tcp(6) E 192.168.252.1 55250 192.168.252.130 102: 18 0
2019-09-30-06:20:07.5795 tcp(6) E 192.168.252.1 55251 192.168.252.130 102: 48 0
```

Figura 46 Conexión S7comm

Tal y como se hace con otros dispositivos dentro de una red, se podría configurar el envío de este registro por syslog para que pueda ser tratado por un sistema centralizado de logs.

9. Conclusiones

Tal y como reflejan las estadísticas[13], el número de ciberataques contra los entornos industriales, al igual que en el resto de ellos, va en aumento día a día. Cada uno de estos ataques puede suponer un gran impacto a nivel global, lo cual pone a los entornos industriales en el punto de mira de los atacantes. Para poder preparar una respuesta más eficaz contra estos posibles ataques, numerosos organismos de referencia en ciberseguridad poseen iniciativas enfocadas al estudio de las últimas técnicas en ciberataques que se emplean en este tipo de redes.

Una de las formas para protegernos es aprender y conocer cómo actúan los atacantes, una tarea que gracias a la implementación y uso de honeypots, puede resultar más sencilla y que aporta múltiples beneficios, entre los que destacan el aprendizaje de nuevas técnicas de ataque y la distracción sobre su objetivo real. No obstante, también hay que tener en cuenta y controlar sus desventajas, como son exponer al exterior nuestros dispositivos de la red o facilitar a los atacantes el acceso a determinados tipos de servicios y dispositivos.

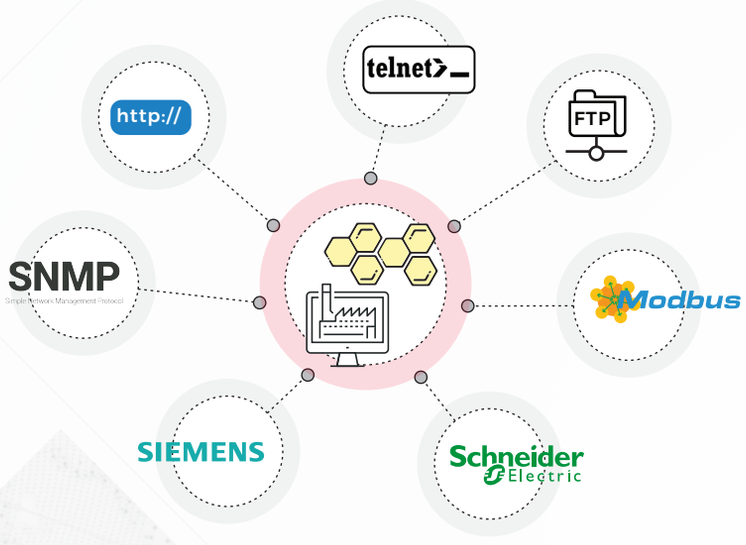
La herramienta utilizada en este estudio, Honeyd, permite al usuario desplegar honeypots para identificar las posibles amenazas a las que se enfrenta una red industrial, simulando diferentes elementos de su arquitectura, como PLC, diferentes sistemas operativos, servicios, etc., recopilando así información de los supuestos atacantes para mejorar su nivel de ciberseguridad.

[13] https://vestertraining.com/sectores-industriales-recibieron-mas-ciberataques-2018/

incibe_ INSTITUTO NACIONAL DE CIBERSEGURIDAD

incibe-cert_

Despliegue de un honeypot industrial



INSTALACIÓN DE HONEYD

INSTALACIÓN DE GIT

```
sudo apt-get install git
```

DESCARGA DE HONEYD

```
git clone https://github.com/DataSoft/Honeyd
```

INSTALACIÓN DE DEPENDENCIAS

```
sudo apt-get install libevent-dev libdumbnet-dev libpcap-dev libpcre3-dev
libedit-dev bison flex libtool automake zliblg-dev python net-tools
```

COMPILACIÓN E INSTALACIÓN DE HONEYD

```
cd Honeyd/
./autogen.sh
./configure
make
```

CREACIÓN DE DIRECTORIO PARA FICHEROS DE CONFIGURACIÓN

```
cd ..
mkdir <nombre_directorio>
```

CONFIGURACIÓN DEL HONEYPOT

DESCARGA DE SCADA HONEYNET PROJECT

```
http://www.sf.net/projects/scadahoneynet
```

MOVER DIRECTORIO SCRIPTS A LA RUTA DEL HONEYPOT

```
cd <ruta_descarga>
tar -xvzf <archivo_scadahoneynet.tar>
cp -a ./cernsacadahoneynet/files/scripts <nombre_directorio>/scripts
```

Esta última ruta se etiquetará como <ruta_scripts> para los siguientes pasos.

MODIFICACIÓN DEL SCRIPT WEB

Editar <ruta_scripts>/honeyd-http-siemens.py

```
webroot = "/var/cshoneynet/scripts/web-siemens" -> webroot = "<ruta_scripts>/web-siemens"
```

RENOMBRAR Y MODIFICAR ARCHIVO TELNET

Dentro de <ruta_scripts>

```
cp honeyd-telnet-schneider.py honeyd-telnet-siemens.py
```

Modificar fichero honeyd-telnet-siemens.py:

```
logintext = "\n\rVxWorks login: " -> logintext = "\n\rSiemens Login: "
```

MODIFICAR ARCHIVO NMAP.ASSOC

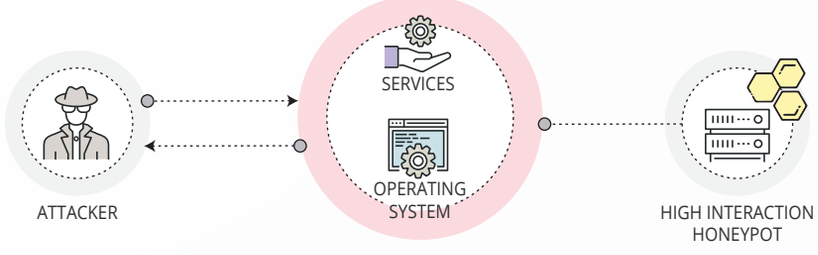
```
cat /usr/share/honeyd/nmap-os-db | grep "Siemens\ Simatic\ 300"
```

Añadir el resultado (sin Fingerprint) al final de la lista de /usr/share/honeyd/nmap.assoc. En caso de que ya esté, asegurarse de que no quede como comentario.

ARCHIVO DE CONFIGURACIÓN

Crear un archivo de configuración (<nombreFichero.conf>) dentro del directorio <nombre_directorio> e incluir las siguientes líneas de configuración:

```
create siemens
set siemens ethernet "00:1f:f8:cc:d0:23"
set siemens default tcp action closed
set siemens default udp action reset
set siemens personality "Siemens Simatic 300 programmable logic controller"
add siemens tcp port 21 "python <ruta_scripts>/honeyd-ftp-siemens.py"
add siemens tcp port 23 "python <ruta_scripts>/honeyd-telnet-siemens.py"
add siemens tcp port 80 "python <ruta_scripts>/honeyd-http-siemens.py"
add siemens tcp port 102 "python <ruta_scripts>/honeyd-s7.py"
add siemens udp port 161 " python <ruta_scripts>/honeyd-snmp-siemens.py"
add siemens tcp port 502 " python <ruta_scripts>/honeyd-modbus.py"
set siemens uptime <timestamp en segundos>
bind <dirección_IP> siemens
```

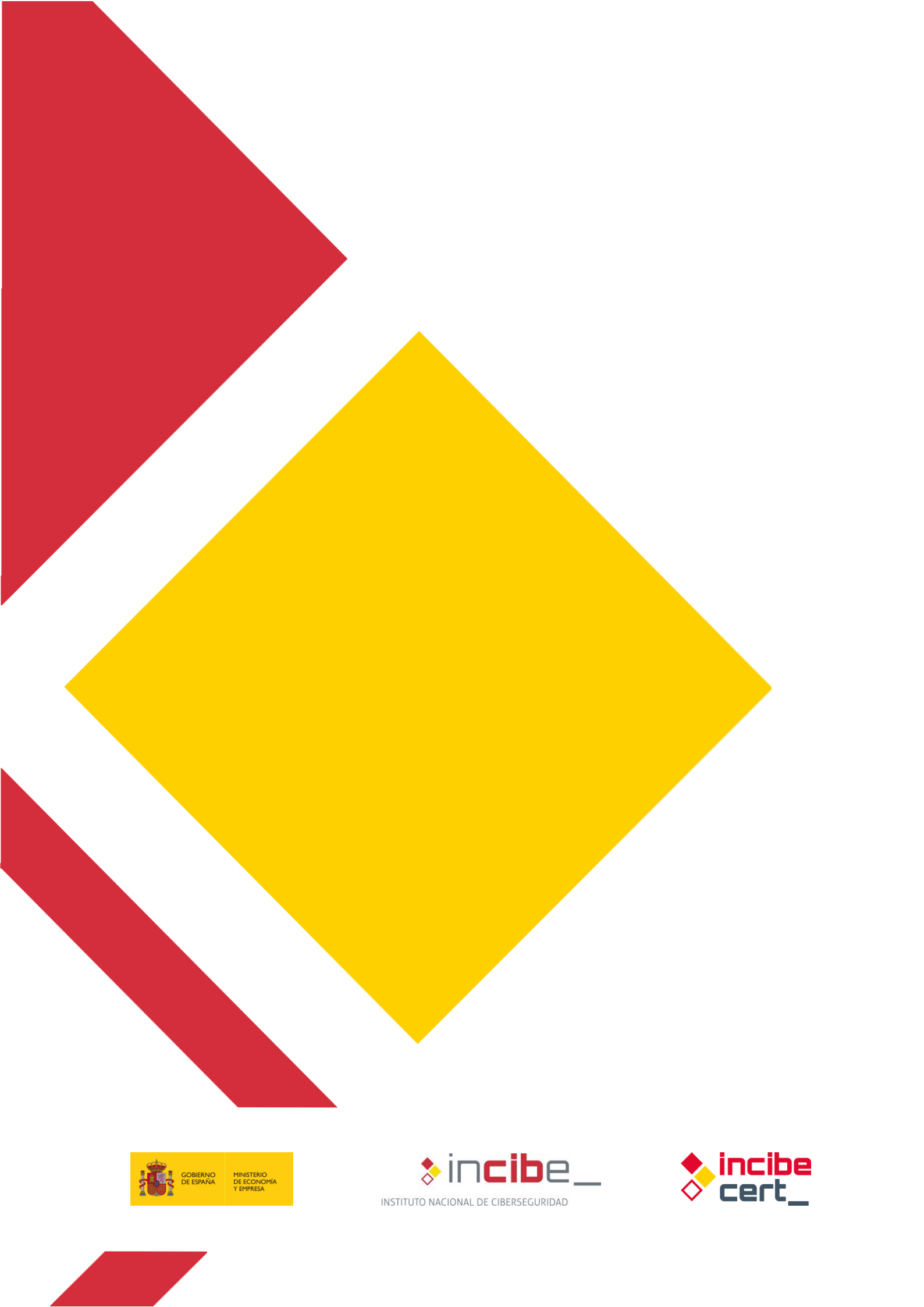


EJECUCIÓN DE HONEYD

```
sudo honeyd -d -p nmap-os-db -i <interfaz> -l <nombre_log.log> -f
<nombreFichero.conf> <dirección_IP_o_subred> -u 0 -g 0 --disable-webserver
```

10. Referencias

Referencia	Título, autor, fecha y enlace web
[Ref.- 1]	"Conoce a tu enemigo: Definiendo honeynets virtuales". Honeynet Project. 4 de septiembre de 2012. URL: http://his.sourceforge.net/honeynet/papers/virtual/
[Ref.- 2]	"Honeypots". 9 de marzo de 2012. URL: http://tecnoloxiaxa.blogspot.com/2012/03/honeypots.html
[Ref.- 3]	"Honeyd". Wikipedia. 10 de julio de 2019. URL: https://en.wikipedia.org/wiki/Honeyd
[Ref.- 4]	"Honeynet". Wikipedia. 19 de julio de 2019. URL: https://es.wikipedia.org/wiki/Honeynet
[Ref.- 5]	"Honeypot". Wikipedia. 19 de julio de 2019. URL: https://es.wikipedia.org/wiki/Honeypot
[Ref.- 6]	"BeEF Project". Brendan Coles. 11 de marzo de 2018. URL: https://github.com/beefproject/beef/wiki/
[Ref.- 7]	"Real vs. virtual honeypots". Brien Posey. 11 de marzo de 2005. URL: https://searchenterprisedesktop.techtarget.com/tip/Real-vs-virtual-honeypots
[Ref.- 8]	"Open Source Tools for Active Defense Security". Ed Moyle. 5 de junio de 2018. URL: https://securityintelligence.com/open-source-tools-for-active-defense-security/
[Ref.- 9]	"Honeynets, una desconocida en la seguridad informática". Enseñanza CCOO Andalucía. 5 de noviembre de 2009. URL: https://www.feandalucia.ccoo.es/docu/p5sd6337.pdf
[Ref.- 10]	"The Honeynet Project". The Honeynet Project. URL: https://www.honeynet.org/about
[Ref.- 11]	"Honeypot, una herramienta para conocer al enemigo". INCIBE-CERT. INCIBE (Instituto Nacional de Ciberseguridad de España). 14 de junio de 2018. URL: https://www.incibe-cert.es/blog/honeypot-herramienta-conocer-al-enemigo
[Ref.- 12]	"Honeypots industriales". INCIBE-CERT. INCIBE (Instituto Nacional de Ciberseguridad de España). 23 de marzo de 2017. URL: https://www.incibe-cert.es/blog/honeypots-industriales
[Ref.- 13]	"¿Qué es un honeypot?". IONOS Digital Guide. 8 de agosto de 2017. URL: https://www.ionos.es/digitalguide/servidores/seguridad/honeypot-seguridad-informatica-para-detectar-amenazas/
[Ref.- 14]	"How does a Honeypot work?". Manoj Murali. 31 de enero de 2015. URL: https://www.quora.com/How-does-a-Honeypot-work


GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD
incibe cert_