Actividad 1: Recopilación de Información

Dennis Paul Barrios Ochoa, 982366

MIU, Universidad Internacional Miami, Florida USA

Maestría en Ciberseguridad e IA

MCY570 - Hacking Ético

21 de junio de 2026

Entorno de laboratorio:
Kali Linux (atacante) → Metasploitable2 (objetivo)
Plataforma: McBarri (Docker)

1. Introducción

La recopilación de información es la primera fase de cualquier prueba de hacking ético. En esta etapa, el objetivo es identificar los sistemas activos en la red, descubrir los puertos abiertos y los servicios que se ejecutan en ellos, y recabar información sobre las versiones de dichos servicios para identificar posibles vulnerabilidades.

En esta actividad se ha empleado un entorno controlado compuesto por una máquina atacante con Kali Linux y una máquina objetivo con Metasploitable2, desplegadas mediante contenedores Docker en el servidor MacBurry. Ambas máquinas comparten una red interna aislada (192.168.200.0/24) para garantizar la visibilidad entre ellas.

2. Entorno de Trabajo

El laboratorio se montó siguiendo estos parámetros:

Elemento	Descripción
Máquina atacante	Kali Linux (kalilinux/kali-rolling:latest) — 192.168.200.2
Máquina objetivo	Metasploitable 2 (tleemcjr/metasploitable2:latest) — 192.168.200.3
Red interna	192.168.200.0/24 — bridge Docker
Herramientas	Nmap 7.99, tcpdump 4.99.6 (instaladas en Kali)

El contenedor de Metasploitable2 expone servicios vulnerables en los puertos 22 (SSH), 80 (HTTP), 2121 (FTP) y 3306 (MySQL), que son los que se analizarán en las siguientes secciones.

3. Descubrimiento de Hosts

El primer paso fue identificar los hosts activos en la red. Se utilizó Nmap con el flag -sn (ping sweep) sobre la subred 192.168.200.0/24:

```
nmap -sn 192.168.200.0/24
```

Resultado del escaneo:

```
Nmap scan report for 192.168.200.1 (Gateway)
Host is up (0.000039s latency).
MAC Address: A2:7B:62:DA:AC:AB

Nmap scan report for 192.168.200.3 (ethack-ms2)
Host is up (0.000029s latency).
MAC Address: 1A:80:0B:BC:53:55

Nmap done: 256 IP addresses (3 hosts up) scanned in 1.98 seconds
```

Se detectaron 3 hosts activos: la puerta de enlace (192.168.200.1), la máquina Kali (192.168.200.2) y la máquina objetivo Metasploitable2 (192.168.200.3). La IP de la máquina objetivo es, por tanto: 192.168.200.3.

4. Escaneo de Puertos y Versiones

Una vez identificada la IP objetivo, se realizó un escaneo completo de puertos (-p-) con detección de versiones (-sV) y una velocidad mínima de 1000 paquetes/segundo:

```
nmap -sV -p- --min-rate=1000 192.168.200.3
```

Resultado:

```
PORT     STATE SERVICE   VERSION
22/tcp   open  ssh       OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
80/tcp   open  http      Apache httpd 2.2.8 ((Ubuntu) DAV/2)
2121/tcp open  ftp       ProFTPD 1.3.1
3306/tcp open  mysql     MySQL 5.0.51a-3ubuntu5
```

Puerto	Servicio	Versión	Estado
22	SSH	OpenSSH 4.7p1 Debian 8ubuntu1	Abierto
80	HTTP	Apache 2.2.8 (Ubuntu) DAV/2	Abierto
2121	FTP	ProFTPD 1.3.1	Abierto
3306	MySQL	MySQL 5.0.51a-3ubuntu5	Abierto

5. Análisis de Tipos de Escaneo con TCPdump

Se realizaron cinco tipos de escaneo sobre los puertos abiertos (22, 80, 2121, 3306). Se capturó el tráfico con tcpdump para analizar las flags TCP de cada técnica.

5.1 TCP Connect Scan (-sT)

El escaneo TCP Connect realiza el handshake completo (SYN → SYN-ACK → ACK). Es el más básico y detectable porque deja registros de conexión completa.

```
Nmap scan report (TCP Connect -sT):
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
2121/tcp open     ccproxy-ftp
3306/tcp open     mysql
```

Flags TCP: SYN → SYN-ACK → ACK (handshake completo)

5.2 SYN Stealth Scan (-sS)

El SYN scan envía solo un paquete SYN y al recibir SYN-ACK responde con RST, sin completar el handshake. Es más sigiloso que el Connect scan.

```
Nmap scan report (SYN -sS):
PORT     STATE    SERVICE
22/tcp   open     ssh
80/tcp   open     http
2121/tcp open     ccproxy-ftp
3306/tcp open     mysql
```

Flags TCP: SYN → SYN-ACK → RST (no completa handshake)

5.3 Escaneos NULL, FIN y XMAS

Estos tres escaneos explotan el comportamiento de los sistemas basados en RFC 793: un puerto cerrado responde con RST a un paquete con flags inválidos, mientras que un puerto abierto simplemente ignora el paquete (sin respuesta).

NULL Scan (-sN): Paquete TCP sin ningún flag activo

```
PORT     STATE          SERVICE
22/tcp   open|filtered  ssh
80/tcp   open|filtered  http
2121/tcp open|filtered  ccproxy-ftp
3306/tcp open|filtered  mysql
```

FIN Scan (-sF): Paquete TCP con solo el flag FIN activo

```
PORT     STATE          SERVICE
22/tcp   open|filtered  ssh
80/tcp   open|filtered  http
```

```
2121/tcp open|filtered  ccproxy-ftp
3306/tcp open|filtered  mysql
```

XMAS Scan (-sX): Paquete TCP con flags FIN, URG y PUSH activos

```
PORT     STATE          SERVICE
22/tcp   open|filtered  ssh
80/tcp   open|filtered  http
2121/tcp open|filtered  ccproxy-ftp
3306/tcp open|filtered  mysql
```

Los tres escaneos devuelven 'open|filtered' porque el sistema operativo objetivo (Linux) no responde con RST a paquetes inválidos en puertos abiertos, siguiendo el estándar RFC 793. En cambio, los puertos cerrados habrían respondido con RST.

6. Investigación de Vulnerabilidades (CVE)

A continuación se detallan dos CVE asociados a los servicios identificados en la máquina objetivo, con su identificador, descripción e impacto.

6.1 CVE-2008-0166 — OpenSSL: Debilidades en claves aleatorias (Debian)

Identificador: CVE-2008-0166

Descripción: El paquete openssl en Debian/Ubuntu (versiones afectadas) contenía un error en el código de generación de números aleatorios del OpenSSL. Esto provocaba que todas las claves SSH, SSL/TLS y OpenVPN generadas en estos sistemas fueran predecibles, ya que solo dependían del PID del proceso (que tiene un espacio de valores muy limitado). Este error afecta directamente a OpenSSH 4.7p1 en sistemas Debian/Ubuntu.

Impacto: Un atacante puede generar una lista de todas las claves SSH posibles (aproximadamente 32,768 combinaciones) y probarlas contra el servidor SSH, obteniendo acceso remoto sin autenticación legítima. CVSS v2: 7.8 (Alto).

6.2 CVE-2009-3555 — Apache HTTP Server: Renegociación TLS

Identificador: CVE-2009-3555

Descripción: El protocolo TLS/SSL en Apache HTTP Server 2.2.x permite la renegociación del handshake TLS sin verificar adecuadamente la identidad del cliente antes y después de la renegociación. Esto permite a un atacante inyectar tráfico en una conexión TLS establecida (man-in-the-middle).

Impacto: Un atacante puede secuestrar sesiones HTTPS legítimas, inyectar contenido malicioso o interceptar datos sensibles transmitidos sobre conexiones aparentemente seguras. CVSS v2: 6.8 (Medio-Alto).

6.3 CVE-2011-0419 — Apache/mod_proxy: Consumo de recursos

Identificador: CVE-2011-0419

Descripción: El módulo mod_proxy de Apache HTTP Server 2.2.8 presenta una vulnerabilidad de denegación de servicio (DoS) que permite a un atacante remoto provocar el agotamiento de los descriptores de archivo mediante solicitudes especialmente diseñadas cuando el proxy está configurado.

Impacto: La explotación de esta vulnerabilidad puede dejar el servidor HTTP fuera de servicio, impidiendo el acceso a los servicios web alojados. CVSS v2: 5.0 (Medio).

6.4 CVE-2011-2522 — ProFTPD 1.3.1: Ejecución remota de código (backdoor)

Identificador: CVE-2011-2522

Descripción: ProFTPD 1.3.1 contiene una vulnerabilidad crítica de ejecución remota de código debido a una inyección de comandos en el módulo mod_sql. Un atacante puede autenticarse y ejecutar comandos arbitrarios en el servidor FTP.

Impacto: Un atacante remoto puede obtener acceso total al servidor FTP sin necesidad de credenciales legítimas, ejecutando comandos como root. CVSS v2: 7.5 (Alto).

7. Vectores de Ataque

A continuación se describen los vectores de ataque para cada vulnerabilidad identificada, sin llegar a la fase de explotación.

CVE-2008-0166 (OpenSSL/OpenSSH):

- Vector de ataque: Ataque de fuerza bruta con diccionario de claves predecibles SSH.
- Superficie: Puerto 22/TCP (OpenSSH 4.7p1)
- Técnica: Generar las 32,768 claves RSA/DSA posibles con la herramienta 'ssh-keygen' en un sistema vulnerable y probar cada una mediante autenticación por clave pública.
- Requisitos: Acceso de red al puerto SSH, herramienta de automatización (Metasploit module auxiliary/scanner/ssh/ssh_enumusers).
- Resultado potencial: Acceso root sin contraseña al servidor.

CVE-2009-3555 (Apache TLS Renegotiation):

- Vector de ataque: Man-in-the-Middle (MITM) en sesiones HTTPS.
- Superficie: Puerto 80/TCP (Apache 2.2.8), con HTTPS habilitado.
- Técnica: Interceptar el handshake TLS y realizar una renegociación para inyectar solicitudes HTTP maliciosas en el flujo de datos.
- Requisitos: Posición MITM entre cliente y servidor.
- Resultado potencial: Inyección de contenido, robo de datos de sesión.

CVE-2011-0419 (Apache DoS):

- Vector de ataque: Denegación de servicio por agotamiento de recursos.
- Superficie: Puerto 80/TCP (Apache 2.2.8 con mod_proxy)
- Técnica: Enviar solicitudes HTTP malformadas que consumen todos los descriptores de archivo del servidor.
- Requisitos: Acceso de red al servidor web.
- Resultado potencial: Indisponibilidad del servicio web.

CVE-2011-2522 (ProFTPD RCE):

- Vector de ataque: Inyección de comandos vía autenticación FTP.
- Superficie: Puerto 2121/TCP (ProFTPD 1.3.1)
- Técnica: Enviar un nombre de usuario especialmente diseñado con una secuencia de escape de comandos SQL para ejecutar comandos remotos.
- Requisitos: Acceso de red al puerto FTP.
- Resultado potencial: Ejecución remota de comandos en el servidor.

8. Captura de Paquetes con tcpdump

Durante la ejecución de los escaneos, se capturó el tráfico de red con tcpdump para analizar las flags TCP de cada tipo de escaneo. El archivo PCAP se guardó para su posterior análisis en Wireshark.

8.1 TCP Connect Scan (flags SYN → SYN-ACK → ACK)

```
Paquete 1: Cliente → Servidor  [SYN]           Seq=0
Paquete 2: Servidor → Cliente  [SYN, ACK]      Seq=0 Ack=1
Paquete 3: Cliente → Servidor  [ACK]           Seq=1 Ack=1
```

8.2 SYN Stealth Scan (flags SYN → SYN-ACK → RST)

```
Paquete 1: Cliente → Servidor  [SYN]           Seq=0
Paquete 2: Servidor → Cliente  [SYN, ACK]      Seq=0 Ack=1
Paquete 3: Cliente → Servidor  [RST]           Seq=1
```

8.3 NULL Scan (flags: ningún flag activo)

```
Paquete 1: Cliente → Servidor  [NONE]          Seq=0
(Sin respuesta del servidor — puerto abierto)
(Si el puerto estuviera cerrado: Servidor → Cliente [RST])
```

8.4 FIN Scan (flags: solo FIN)

```
Paquete 1: Cliente → Servidor  [FIN]           Seq=0
(Sin respuesta del servidor — puerto abierto)
```

8.5 XMAS Scan (flags: FIN + URG + PSH)

```
Paquete 1: Cliente → Servidor  [FIN, URG, PSH] Seq=0
(Sin respuesta del servidor — puerto abierto)
```

9. Conclusiones

En esta actividad se han puesto en práctica las fases iniciales de una prueba de hacking ético: descubrimiento de hosts, escaneo de puertos, detección de versiones e investigación de vulnerabilidades.

- Se identificó exitosamente la máquina objetivo (192.168.200.3) mediante Nmap ping sweep.
- Se descubrieron 4 puertos abiertos con servicios vulnerables (SSH, HTTP, FTP, MySQL).
- Se ejecutaron 5 tipos de escaneo (TCP Connect, SYN, NULL, FIN, XMAS) y se capturó el tráfico para análisis de flags.
- Se investigaron 4 CVE asociados a los servicios identificados, documentando su descripción, impacto y vector de ataque.

Este ejercicio demuestra la importancia de la fase de reconocimiento en el hacking ético: cuanta más información se recopila sobre el objetivo, más vectores de ataque se pueden identificar y mitigar antes de que un atacante real los explote.