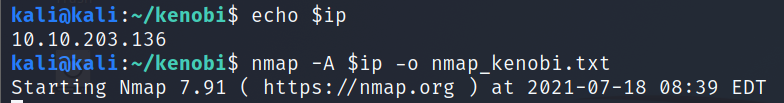
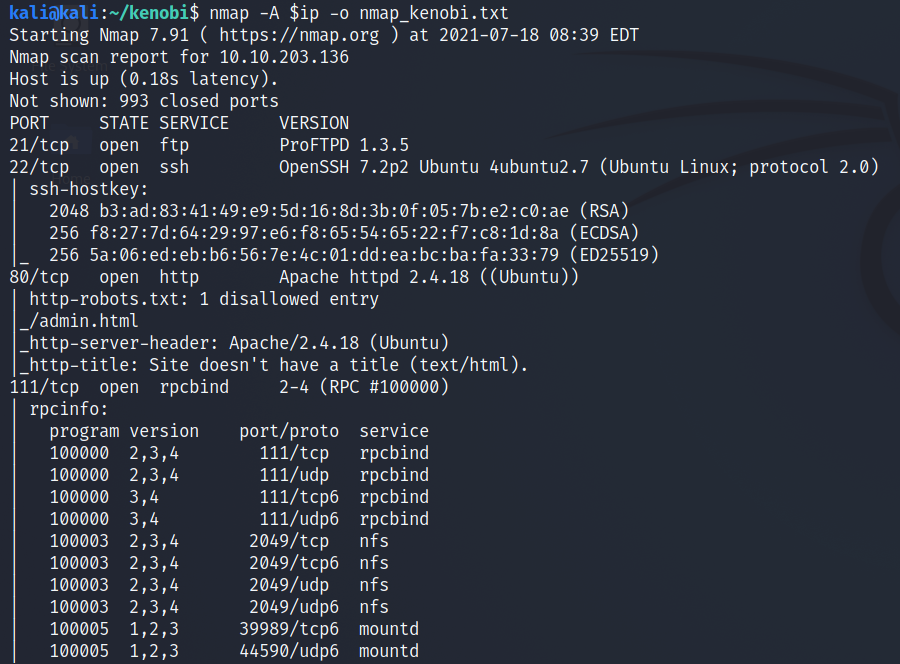
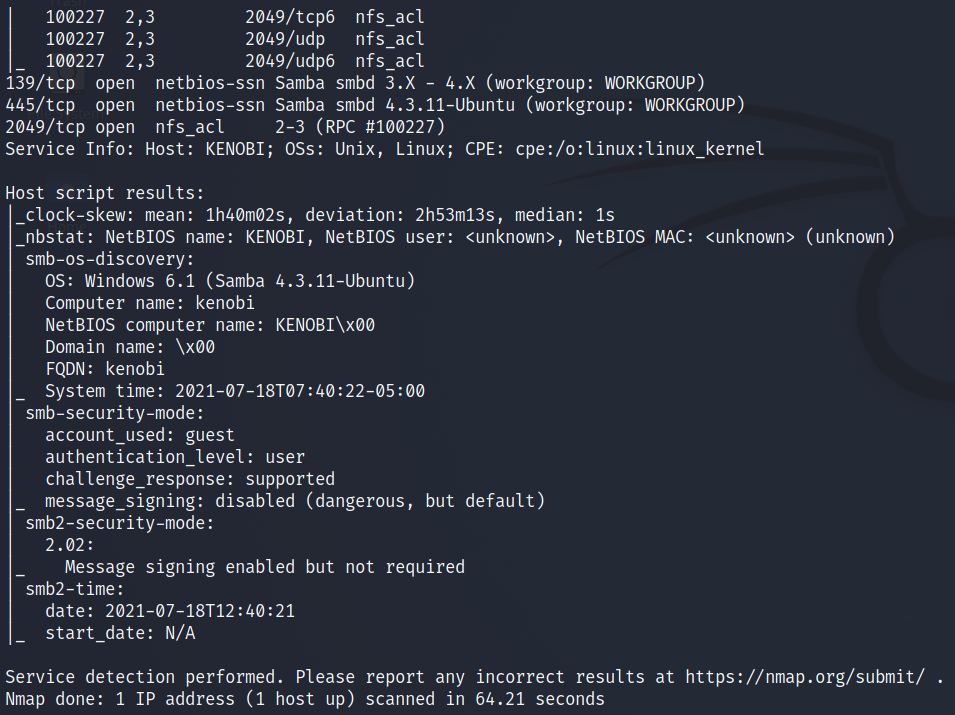
**Kenobi**

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





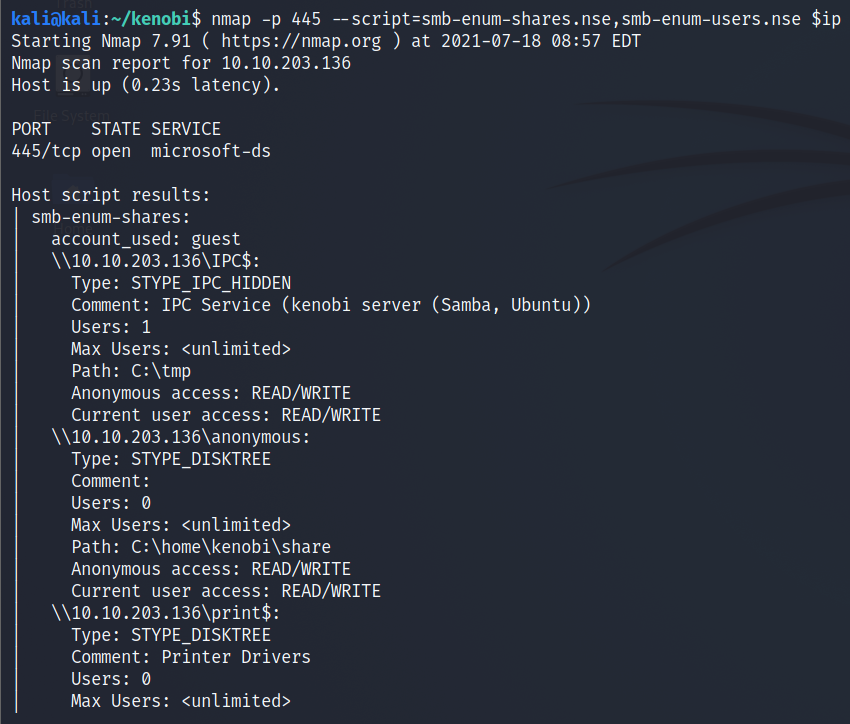


Answer: 7

Enumerating Samba for shares

nmap -p 445 --script=smb-enum-shares.nse,smb-enum-users.nse 10.10.203.136

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

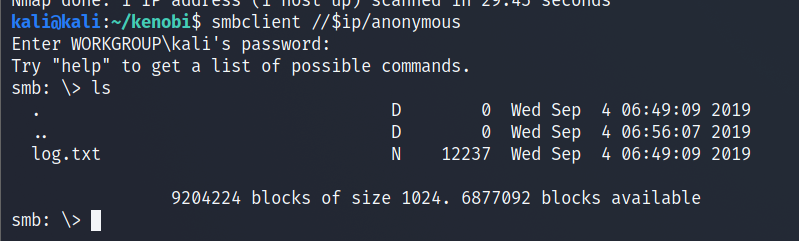


Answer: 3

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?

Answer: log.txt

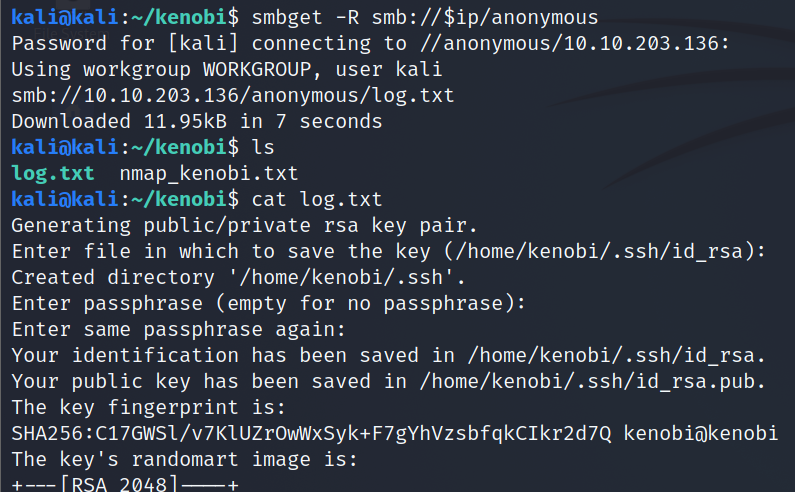
You can recursively download the SMB share too. Submit the username and password as nothing.

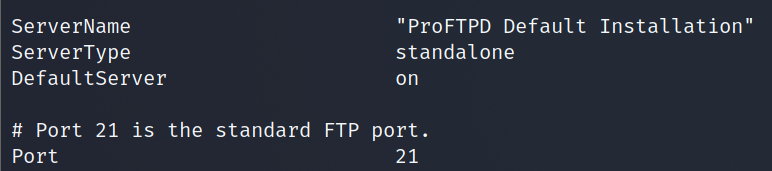
smbget -R smb://<ip>/anonymous

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?





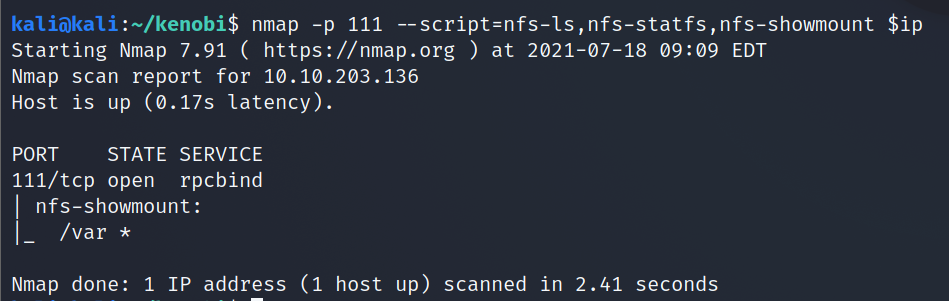
Answer: 21

Your earlier nmap port scan will have shown 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.

nmap -p 111 --script=nfs-ls,nfs-statfs,nfs-showmount 10.10.203.136

What mount can we see?

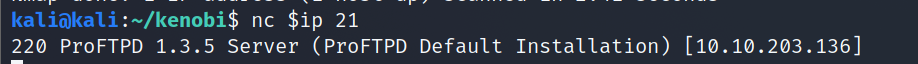


Answer: /var

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

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

What is the version?

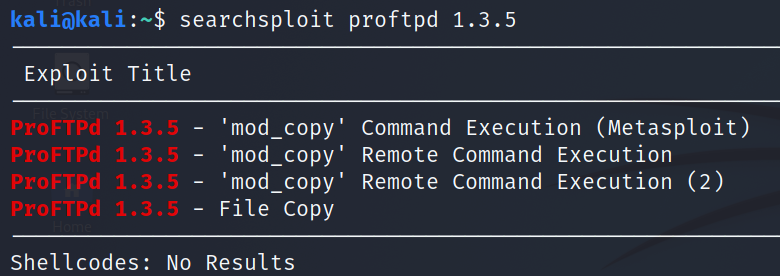


Answer: 1.3.5

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?



Answer: 3

You should have found an exploit from ProFtpd's [mod\_copy module](http://www.proftpd.org/docs/contrib/mod_copy.html).

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.

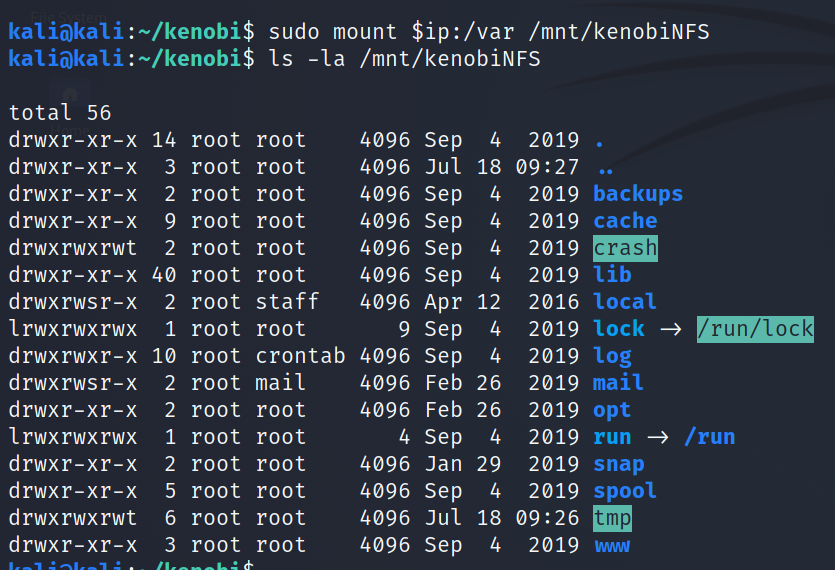
We know that the FTP service is running as the Kenobi user (from the file on the share) and an ssh key is generated for that user.



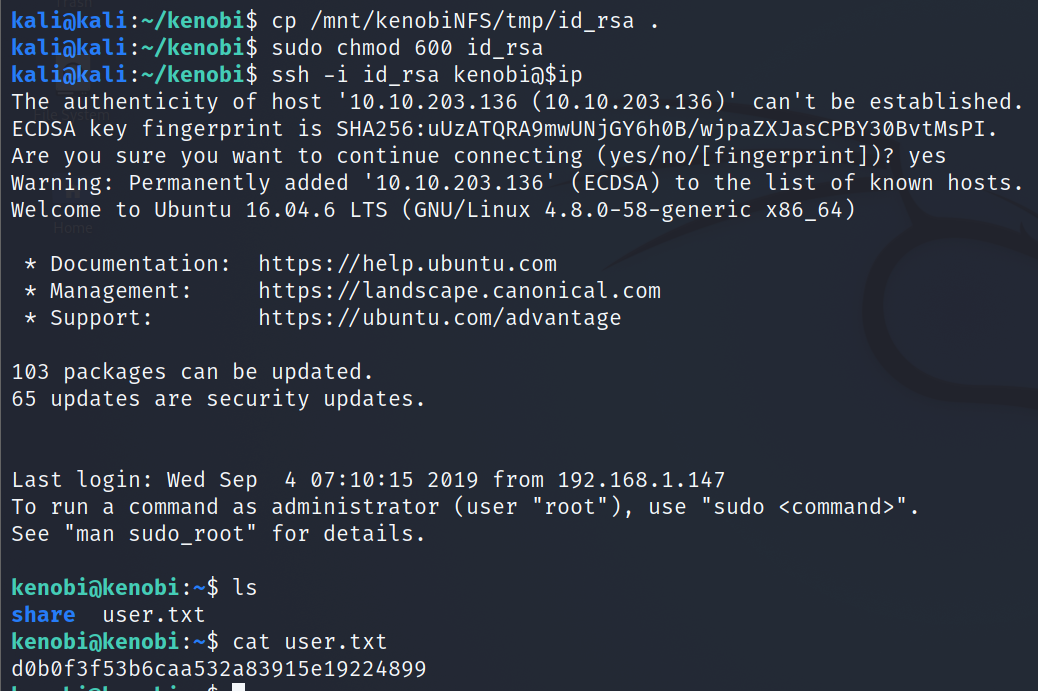
We're now going to copy Kenobi's private key using SITE CPFR and SITE CPTO commands.

Lets mount the /var/tmp directory to our machine

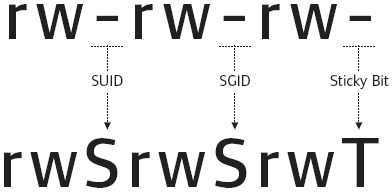
mkdir /mnt/kenobiNFS  
mount machine\_ip:/var /mnt/kenobiNFS  
ls -la /mnt/kenobiNFS



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.



Privilege Escalation with Path Variable Manipulation



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

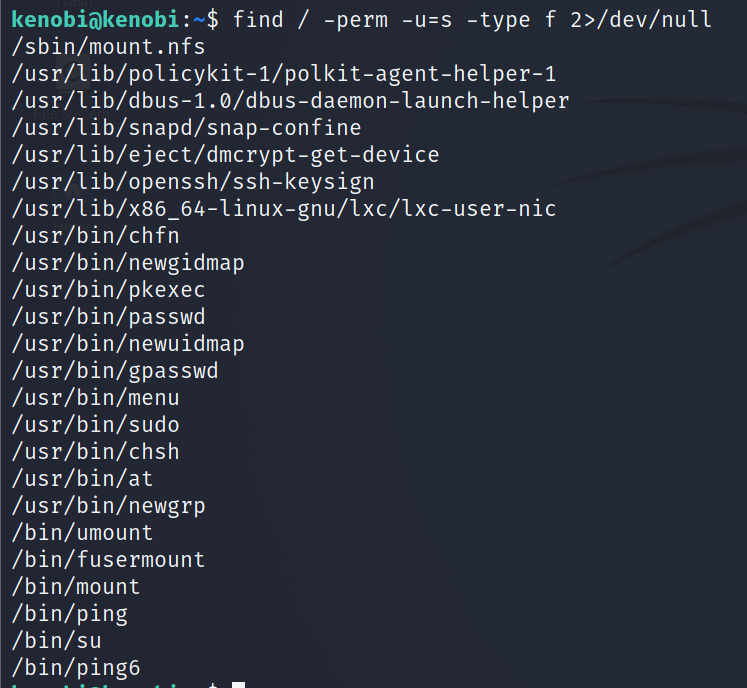
|  |  |  |
| --- | --- | --- |
| **Permission** | **On Files** | **On Directories** |
| SUID Bit | User executes the file with permissions of the *file* owner | - |
| SGID Bit | User executes the file with the permission of the *group* owner. | File created in directory gets the same group owner. |
|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

***Answer the questions below***

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.

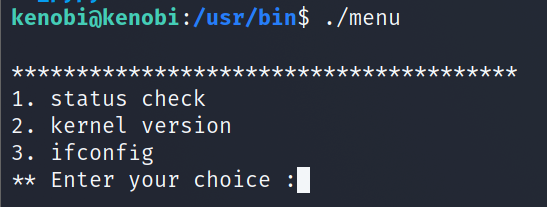
To search the a system for these type of files run the following: find / -perm -u=s -type f 2>/dev/null

What file looks particularly out of the ordinary?



Answer: /usr/bin/menu

Run the binary, how many options appear?



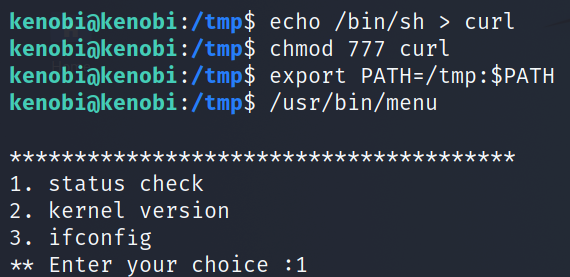
Answer: 3

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!

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

