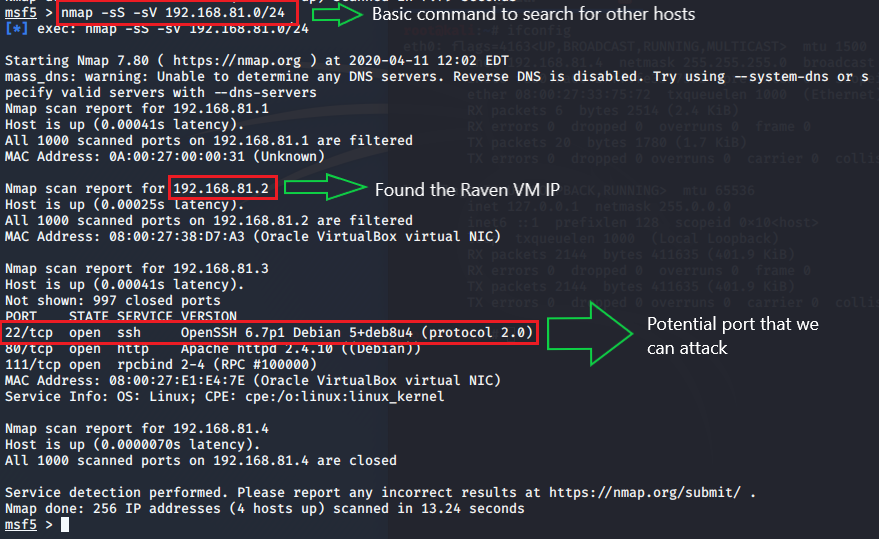
*Team AntiTrust CTF Project*

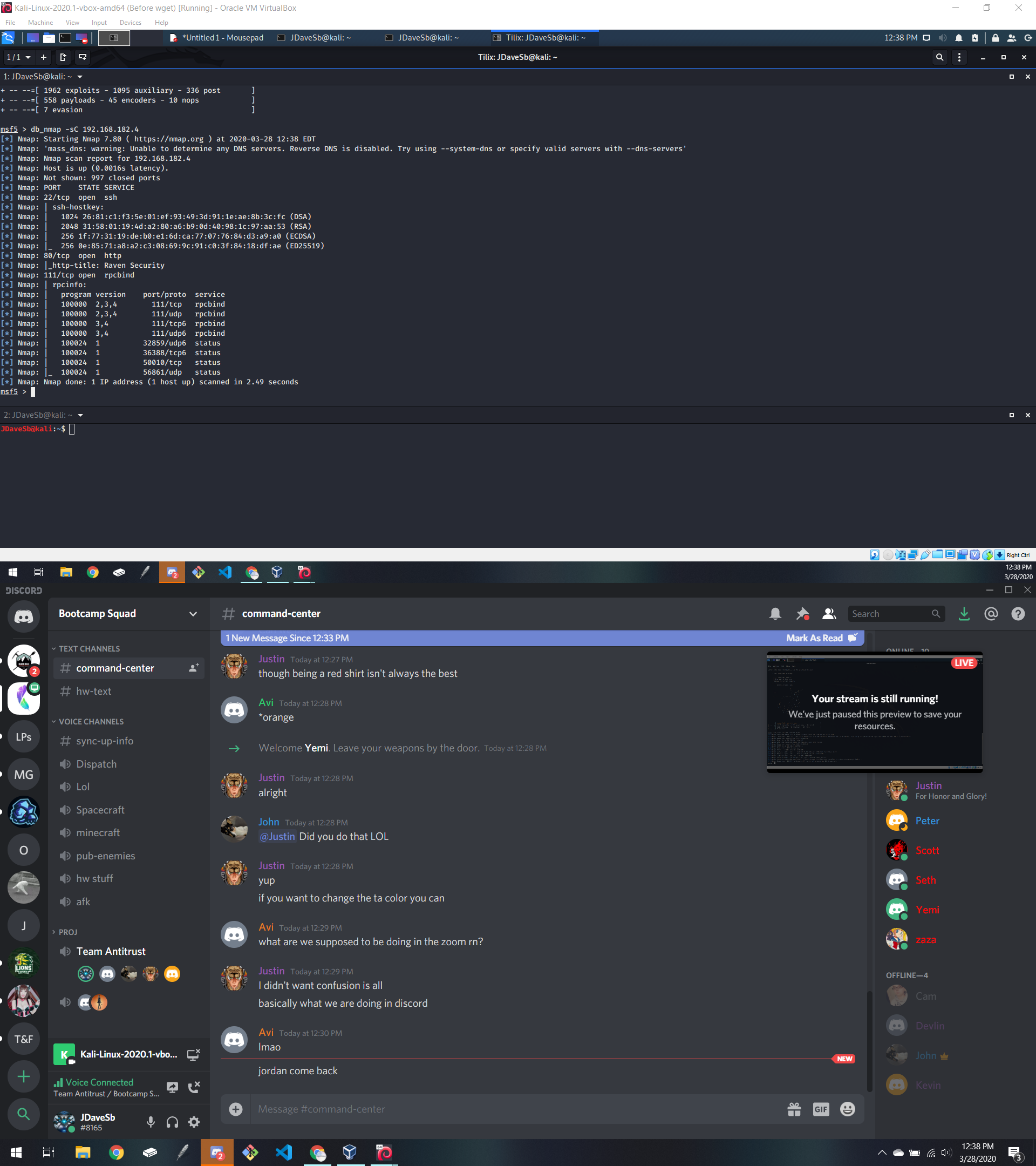
**Executive Summary- Steps for attack**

* **IP Discovery and nmap- vulnerabilities and open ports**
  + **Finding the host server**
* **WPScan to find users**
  + **Steven**
  + **Michael**
* **HTML analysis** 
  + **Locating flag 1**
* **Bruteforce attempt using hydra**
  + **Cracking michaels password**
* **Scanning the files (Mail, /var/www)**
  + **Locating flag 2**
* **Accessing the wordpress database**
  + **Gain steven’s password hash**
  + **Locating flag 3**
* **Cracking steven’s password** 
  + **Finding the hash**
  + **Using John the Ripper to crack it**
* **Switching to root from steven using python script**
* **Scanning the /root directory** 
  + **Locating flag 4**

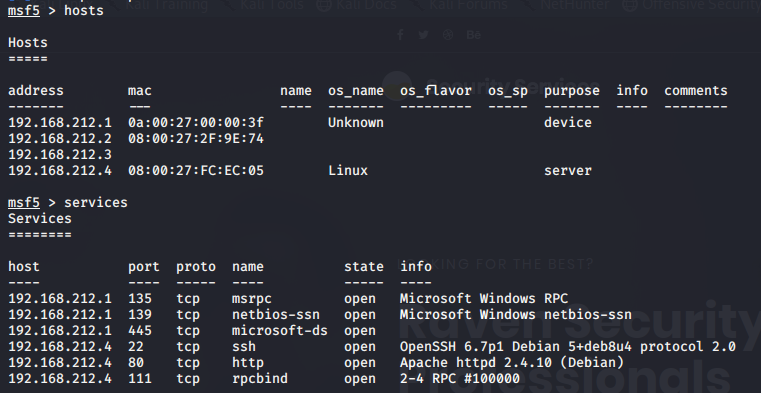
**Reconnaissance**

When getting started, we needed to find the active devices that were sharing the network. There are multiple ways of doing this (nmap, netdiscover, etc) so using the information that we learned during class, we used nmap.

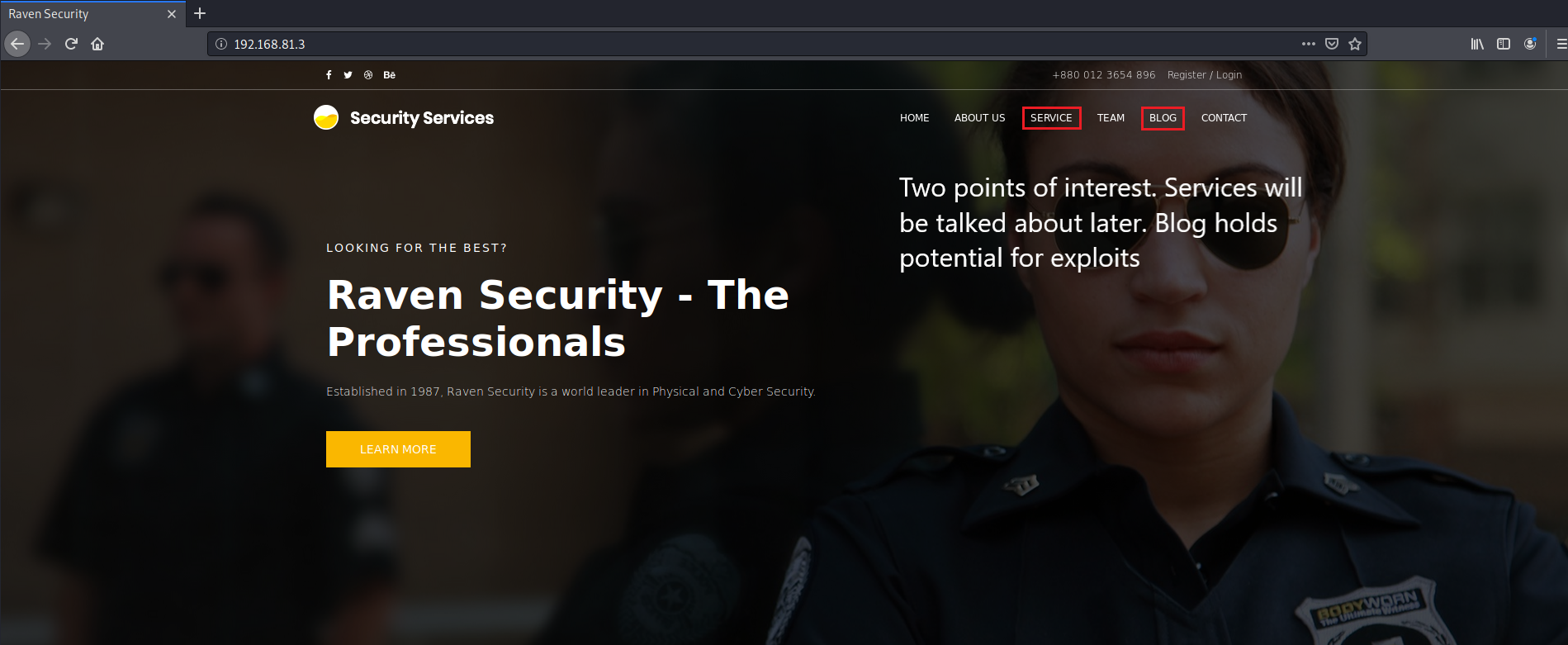
****



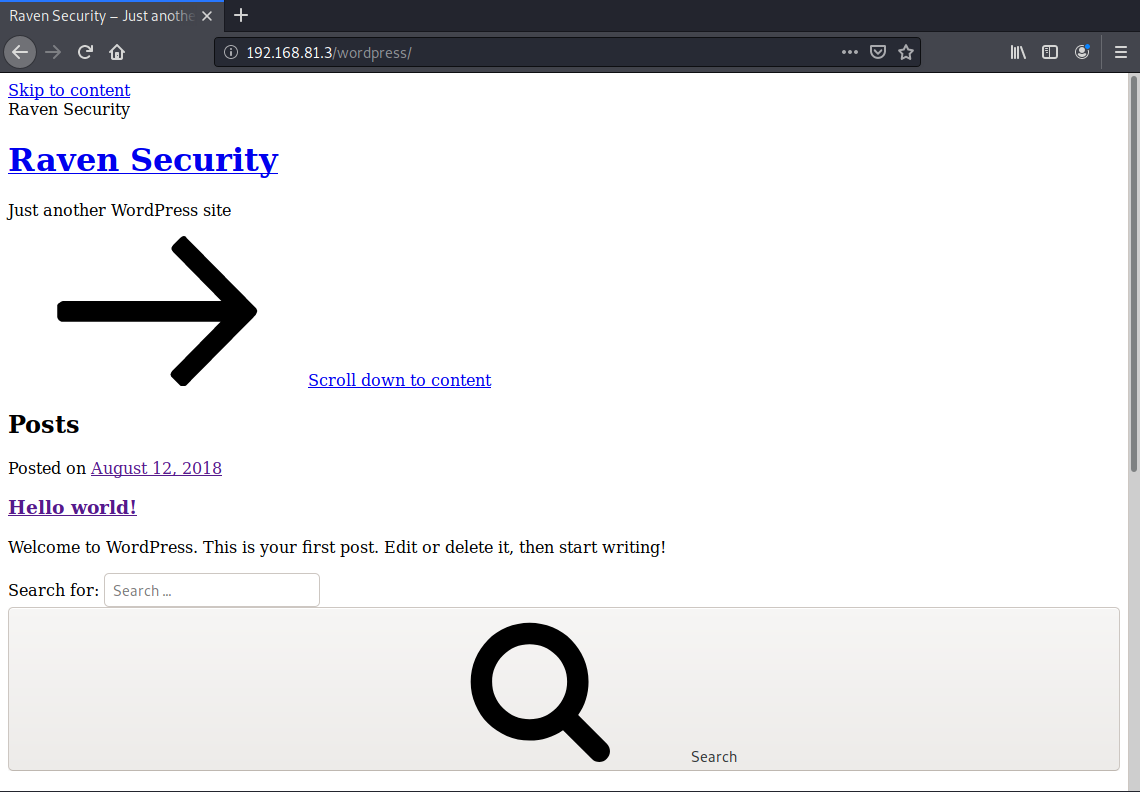
With the basic search completed, we can gather the IP of the Raven server and which port we could exploit (ie. start a remote session through ssh)



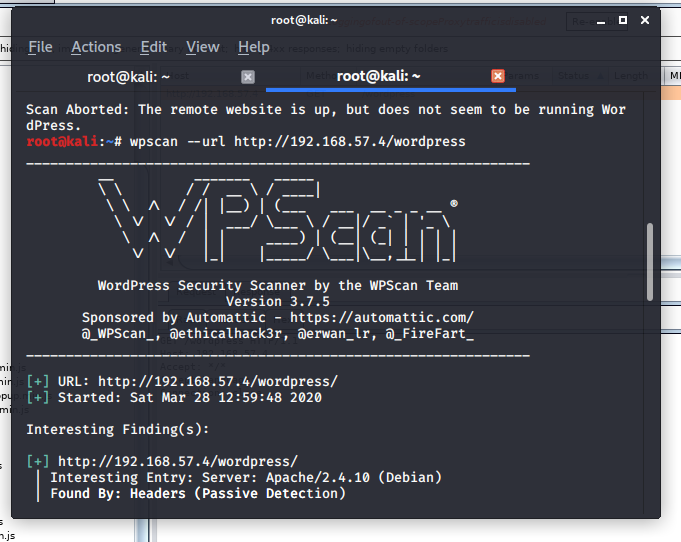
Then using the provided IP address that we found we ran a WPScan to search for valuable users that we can exploit.

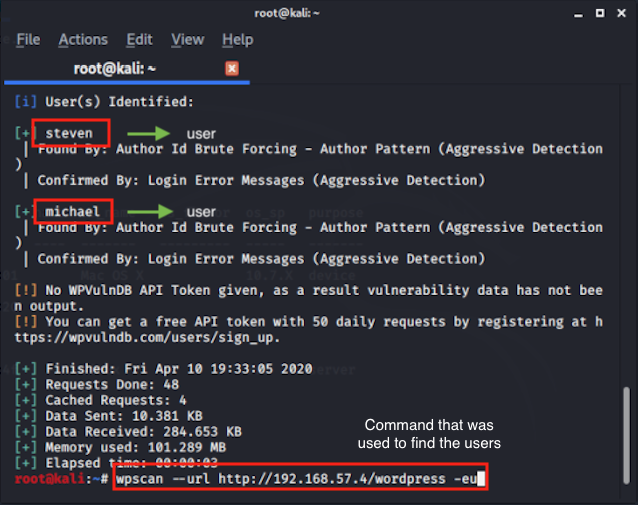


We then move onto the website that is being hosted by this IP, we find some interesting things. If not familiar with web development or some of its tools; the tldr is that this page looks like it is using some kind of management service. You could scan it to see if there are any vulnerabilities (which there are) or depending on your knowledge you could check: /index.html, /wordpress.html, wp-dir.html, etc



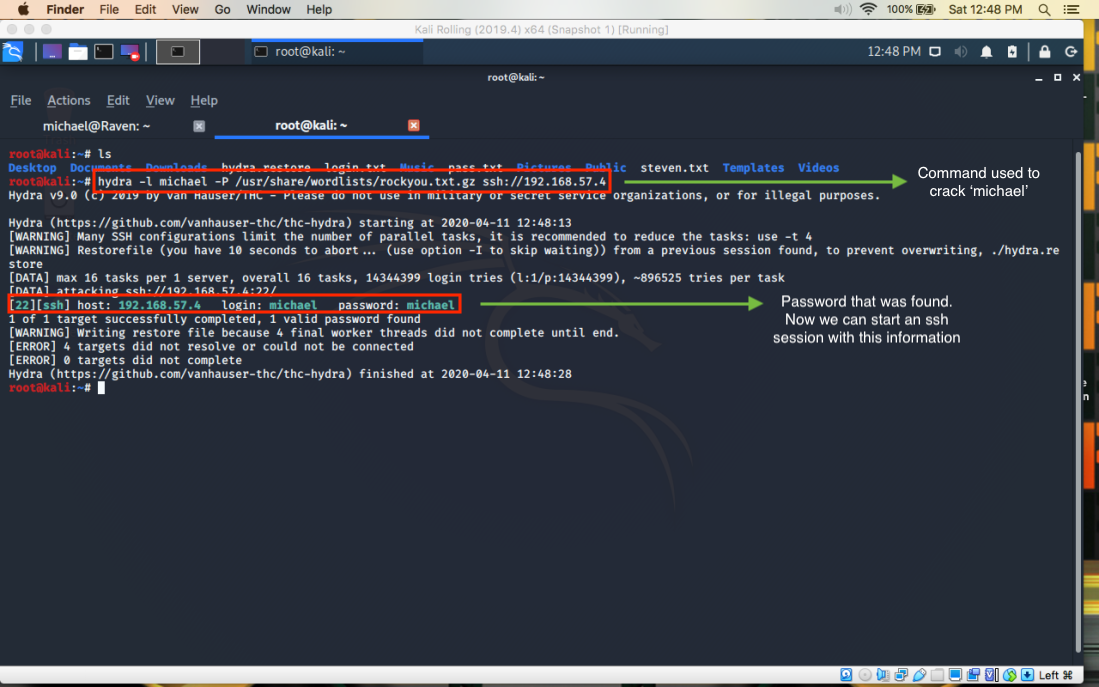
Success! We have loaded the [http://[1920.168.\*\*\*.\*\*\*]/wordpress.html](about:blank) (granted we are missing some formatting rules) and from here we could do a WPScan to see if there is any other information that we could find.





Our scan returned steven and michael. From here there are a few methods that we could use to exploit these users.

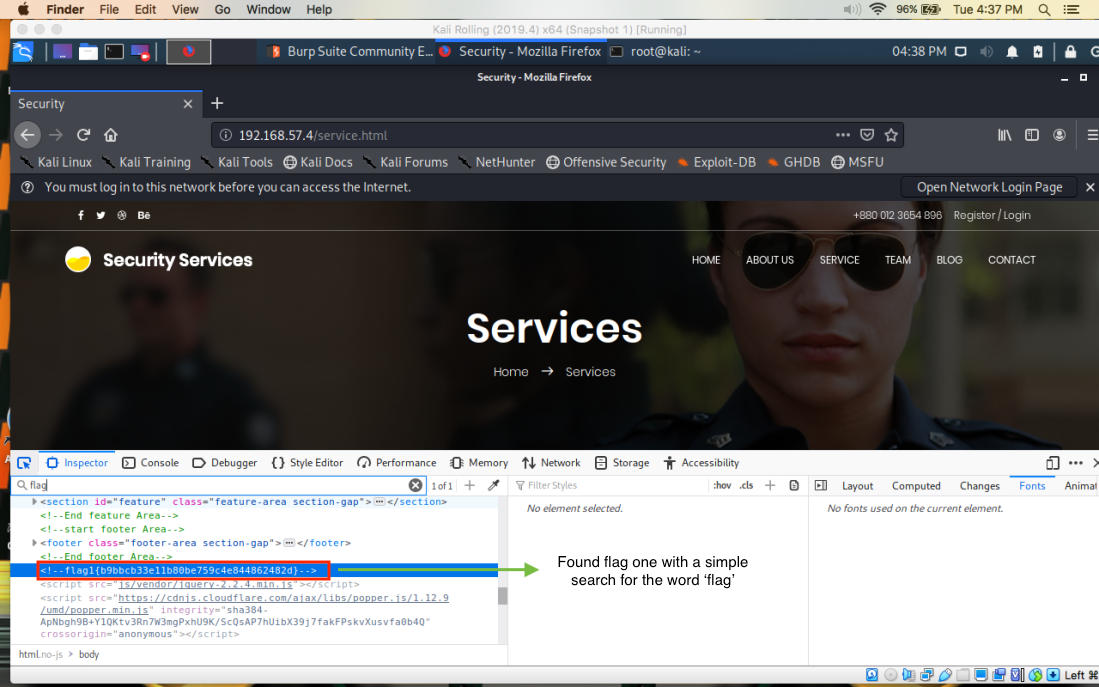
The method that we used was to brute force the user for their password. The quickest result was from michael. Whos password happened to be the same as his user ID

****

Now that we have done some basic reconnaissance let's take a look at this information that we have gathered.

|  |  |  |
| --- | --- | --- |
| **IP Address** | **Vulnerabilities** | **Users/Passwords** |
| 192.168.\*\*\*.\*\*\*  (results may vary) | ssh/ wordpress  (High risk) | **steven**: (not discovered yet)  **michael**:michael |

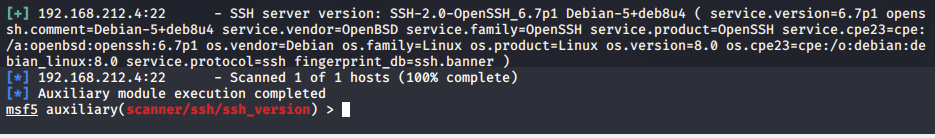
**Web-Server Attack**

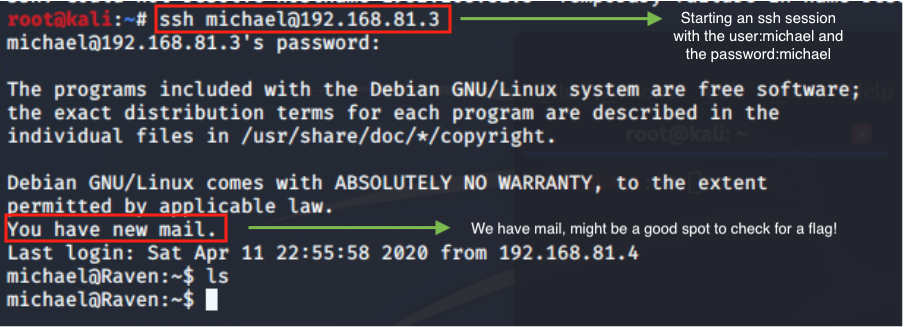


Once we have some information, we then navigate back to the host on a web browser to see what else we can find. Using Firefox’s developer tools (Inspector) we can navigate the website and look for anything hidden under the HTML. Here we have navigated to the services page and utilizing the Inspector tool, a simple search for “flag” displays flag 1 within the HTML.

**NETWORK ANALYSIS**

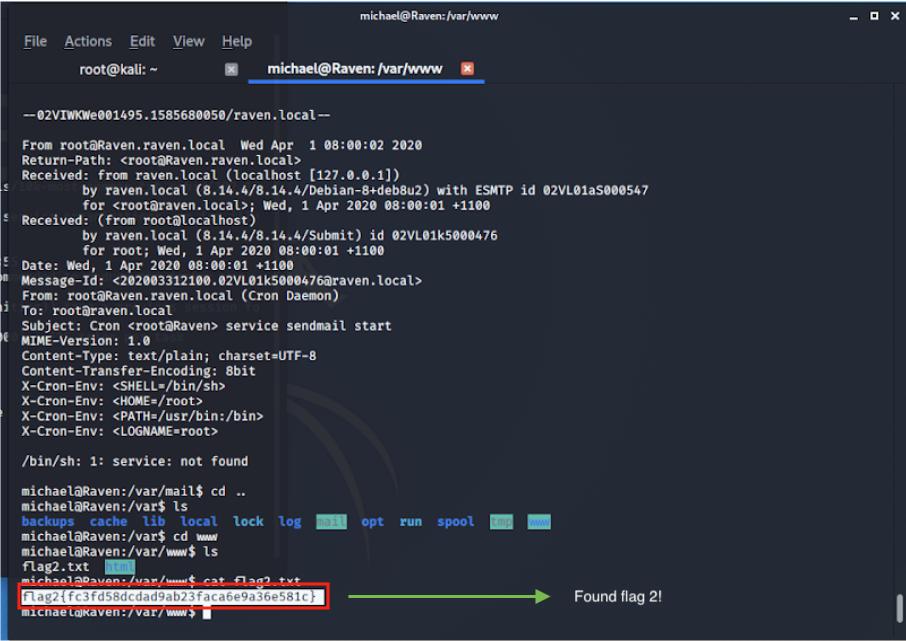
After completing the website inspection, the next objective was to gain access via ssh. Using metasploit we can determine the version of ssh being used on the server. There are also options to try to exploit ssh within metasploit but earlier we used hydra to learn the user “michael” password. With this information we used the ssh command below to gain access.





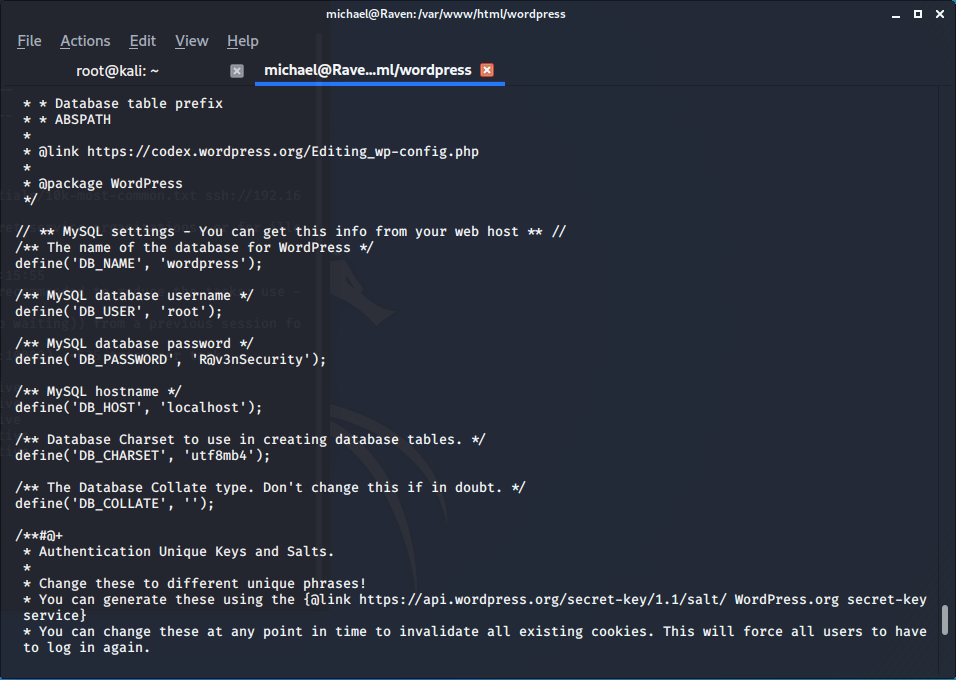
Upon logging into michael’s account we notice that he has a new message and we navigate to read its contents.

In the process we find flag 2

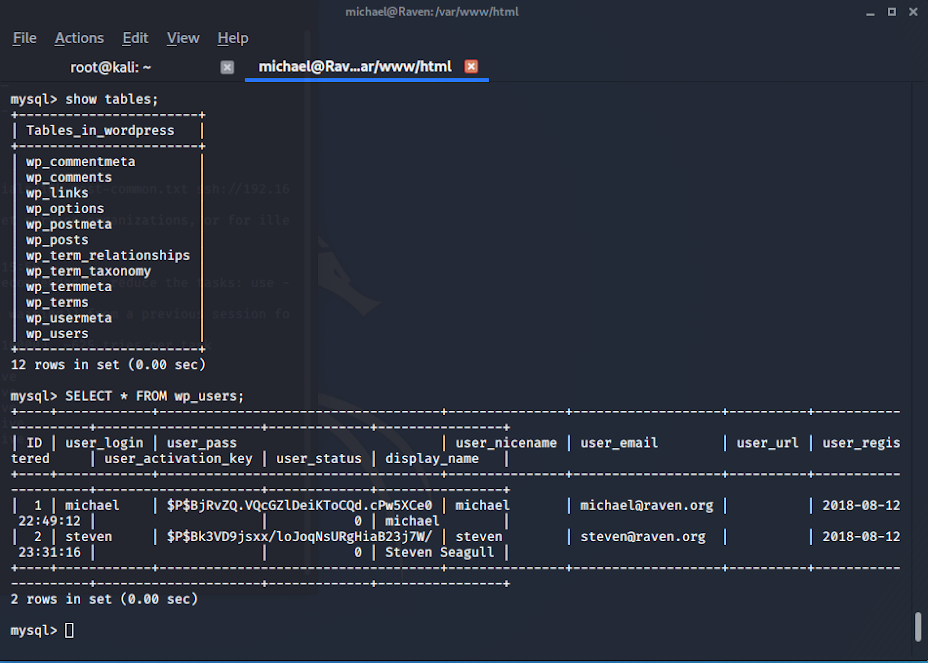


**DATABASE ANALYSIS**

After exploring as user “michael” and finding the second flag the next objective is to gain access to the mysql database. However “michael” does not have access to the sudo or mysql command. In order to circumvent this, we locate the wp-config.php file, the wordpress configuration file, to get the root privileges.

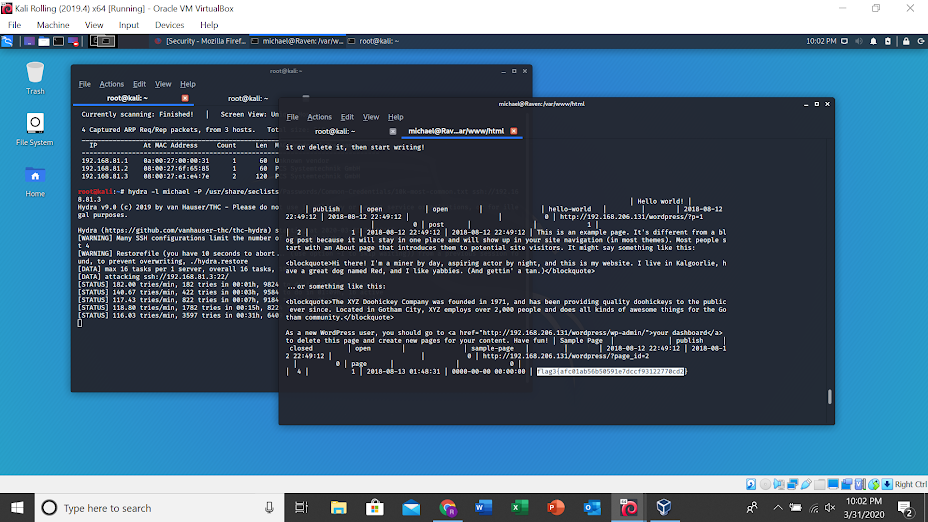


Once we have obtained the mysql root username and password, using the command “mysql -uroot -p” prompts for the password and upon entering the correct one it allows access to the database. Using the wordpress database we see the following tables. Within the wp\_users table we find the password hashes for michael as well as steven, we copy steven’s password for use in the future.



After obtaining the password hashes we explore the other tables in the database. In the wp\_posts table we

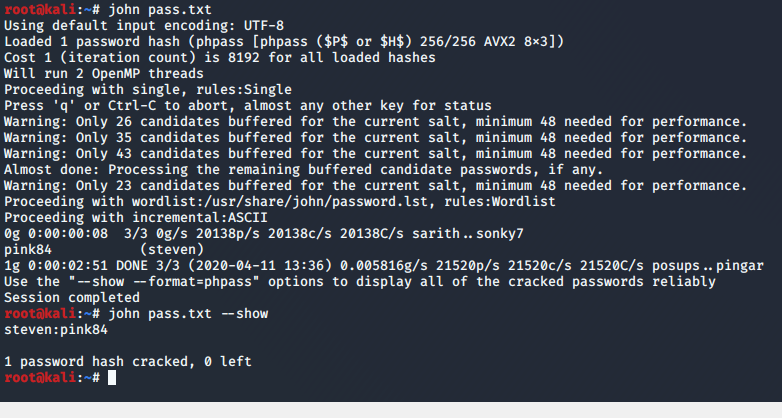
discover the flag 3 hash, which is highlighted below.



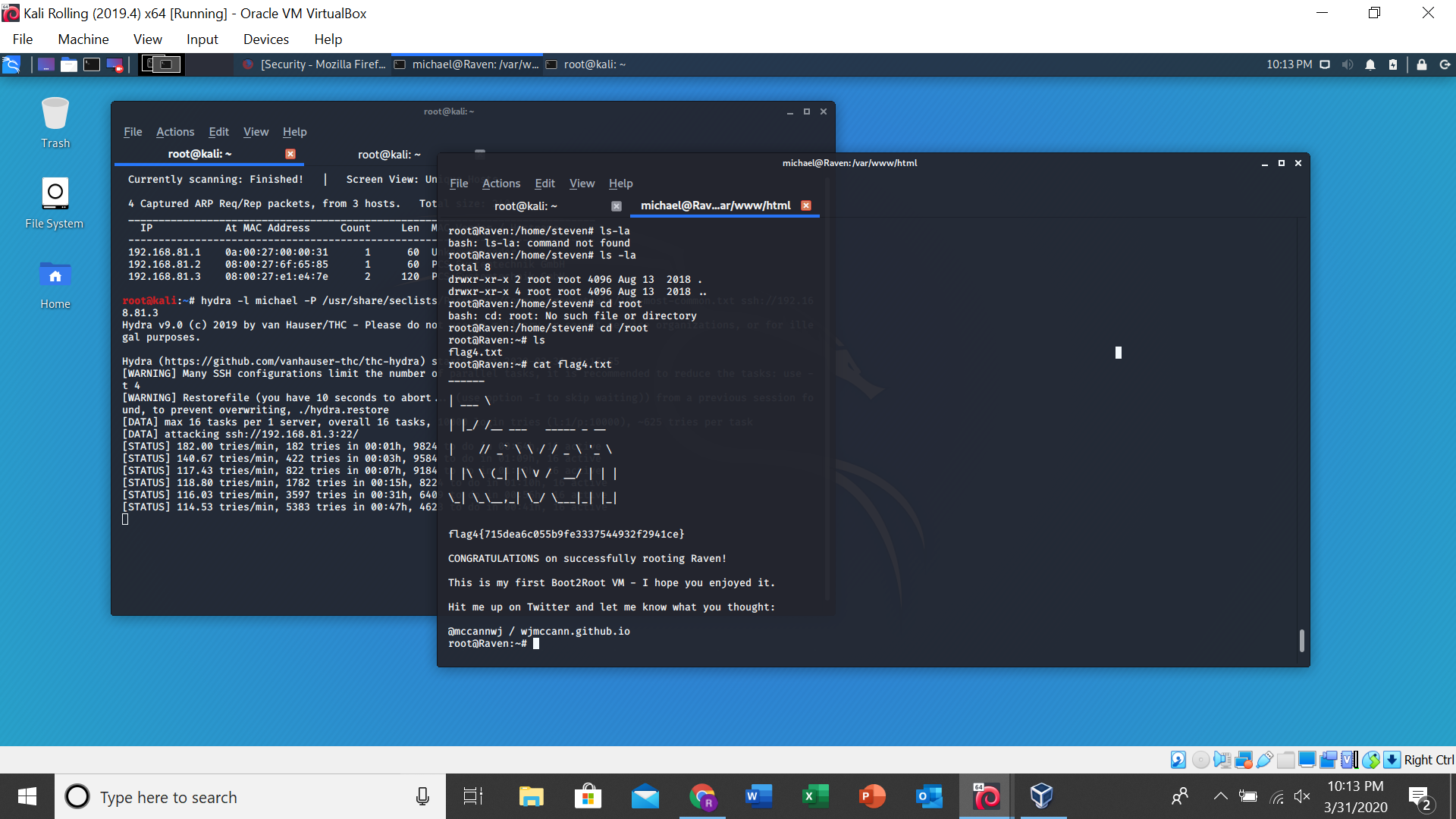
**PRIVILEGE ESCALATION**

After exploring as user “michael” it is time to explore as “steven”. Utilizing the password hash we found earlier we create a file with “steven:<password hash>” to run with john the ripper. Using the below command, john

tells us that user “steven” password is pink84.

****

Going back to our ssh session terminal we can switch to user “steven” and check sudo permissions using the command “sudo -l”. This tells us that “steven” can use “sudo python”. With this knowledge, as user “steven” we can use a python command sudo python -c 'import pty;pty.spawn("/bin/bash");' to gain access to the root user. Looking inside of the root directory we find the fourth and final flag.



**CONCLUSION & RECOMMENDATIONS**

Based on the results above we recommend raven security take the following steps to remediate the vulnerabilities identified.

**Web Server**

* Unique Passwords 8 characters long with symbols, uppercase and numbers
* 2FA for login attempts after 30 days from initial logins and logins from different locations
* Hide unnecessary HTML info on the webpage

**Network Services**

* Have IDS in place to monitor attacks
* Use IDP/ firewall for known vectors
* Set up an internal response team

**Hardening the Server**

* Should not be able to access mysql as root instead should be a separate admin account
* Limit what is shown inside of tables
* Limit who can view sensitive files such as the configuration file

Flags

|  |  |
| --- | --- |
| Flag 1 | *b9bbcb33e11b80be759c4e844862482d* |
| Flag 2 | *fc3fd58dcdad9ab23faca6e9a36e581c* |
| Flag 3 | *afc01ab56b50591e7dccf93122770cd2* |
| Flag 4 | *715dea6c055b9fe3337544932f2941ce* |