VulnHub Raven 1: Capturing Flags and Obtaining Root

Shane Irons

CIS5627 Intro to Offensive Computer Security – FSU Cybersecurity

Dr. Xiuwen Liu

12/9/2022

I. Introduction

This project consists of exploiting and completing Raven 1 from VulnHub. This is a virtual machine that was built to be exploited and runs using Debian 8. For the "hacking", I will be using a default Kali Linux install, which is powerful and comes with many tools that I will be utilizing to complete this challenge. This project is my first Capture the Flag, and it took between eight to ten hours total to configure and complete the Raven 1 challenge.

II. Problem Description

Raven is a beginner/intermediate CTF problem that can be found and downloaded at VulnHub. I had initial issues (shown in demonstration) with configuring Raven as a VM. Initially, I wanted to have it running in VirtualBox on my Kali Linux machine, however, there were issues with this. As a solution, Raven was run in VMWare Workstation Pro alongside Kali. This had no issues and worked fine through the rest of the project. Configuring Kali was simple as well, as after the install the only things needed to do were basic *sudo apt update* and *sudo apt upgrade* commands.

The challenge itself involves using an attacker host (Kali in this case) to gain information on the target (Raven) and obtain flags within as well as root access if possible. Raven contains four flags at unknown locations with initially unknown key areas.

III. Exploits and Techniques

Throughout the project, many Kali Linux tools are used to complete Raven: Netdiscover [1], Nmap [2], Dirbuster [3], Nikto [4], Wpscan [5], Ssh [6], John [7], and Hash-identifier [8]. There are two major vulnerabilities that are exploited in this project: WordPress and Python scripting from the terminal.

1. WordPress

   The major exploit in this challenge resides with WordPress. During the demonstration, the IP address can be found of the Raven machine and when visited, a website is shown. At first, this website does not seem to have any glaring vulnerabilities, however, it is eventually found that it has a WordPress installation. This is a big find because Kali has the wpscanner tool [5] which allowed me to initially find the users "steven" and "michael". With this information, an attacker can try and guess the password for a user (which I did for Michael) or at the least have more information they can use to search files and find a password that way. Running a tool like Hydra on the target can be very time consuming, but also give the hacker the same information.

2. Python Script

   After decoding the hashed password for steven and logging in as that user, it was seen that they had access to Python with sudo level. Knowing this, a python script can be crafted from the terminal window to spawn a root shell. After investigating online, [9] proved to be a helpful resource as it showed a pty import for python that has a shell spawning function. From this document, and after some trial and error, a python script was crafted to spawn a root shell from the user steven. This script can be seen in the demonstration at section IV of this document.

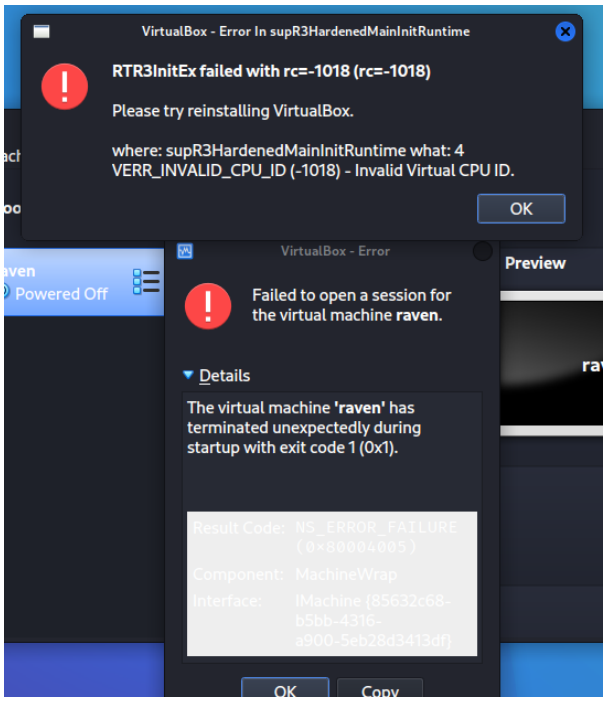
These two exploits lead to the ultimate completion of this challenge. While the WordPress exploit was used in the capture of the four flags, the python script exploit was solely used to

gain root access.

Capturing the four flags involved investigating interesting directories while logged in as the user “michael”. Looking through various files and directories for some time, the directory */var/www* was major progress as two flags were found here. One in the name of a text file simply called *flag2.txt* and the other inside the file at */var/www/html/service.html*. The final two flags were found after I gained access to mysql as user “michael”. Snooping through the various tables in the *WordPress* database, the *wp_posts* table showed the final two flags needed. Although a handful of tools and exploits were used initially, brute force investigation on the user files and database was how all the flags were eventually found. All these points, as well as the entire process from start to finish is documented in the demonstration section (IV).

IV. Demonstration

The following screenshots and explanations document my progress and ultimate completion of the Raven 1 CTF challenge. The first step, installing Kali Linux on VMWare Workshop Pro. This was a default installation and I feel did not require screenshot documentation. Once the VM files were downloaded, I used VMWare to host this machine and booted it. It started with no error and brought me to the desktop, where I ran *sudo apt update* and *sudo apt upgrade* to get everything updated. After configurating Kali, installation of virtual box on Kali Linux was attempted to host Raven. To install virtual box, I first had to import the repository key using curl commands [10]. Then, another update command is given and the installation of kernel modules. This is in the form of the command *sudo apt install -y dkms* [10]. After these steps, virtual box can be installed using *sudo apt install…* [10] and ran from the terminal. After the configuration and download of virtual box, Raven was downloaded and installed into it:

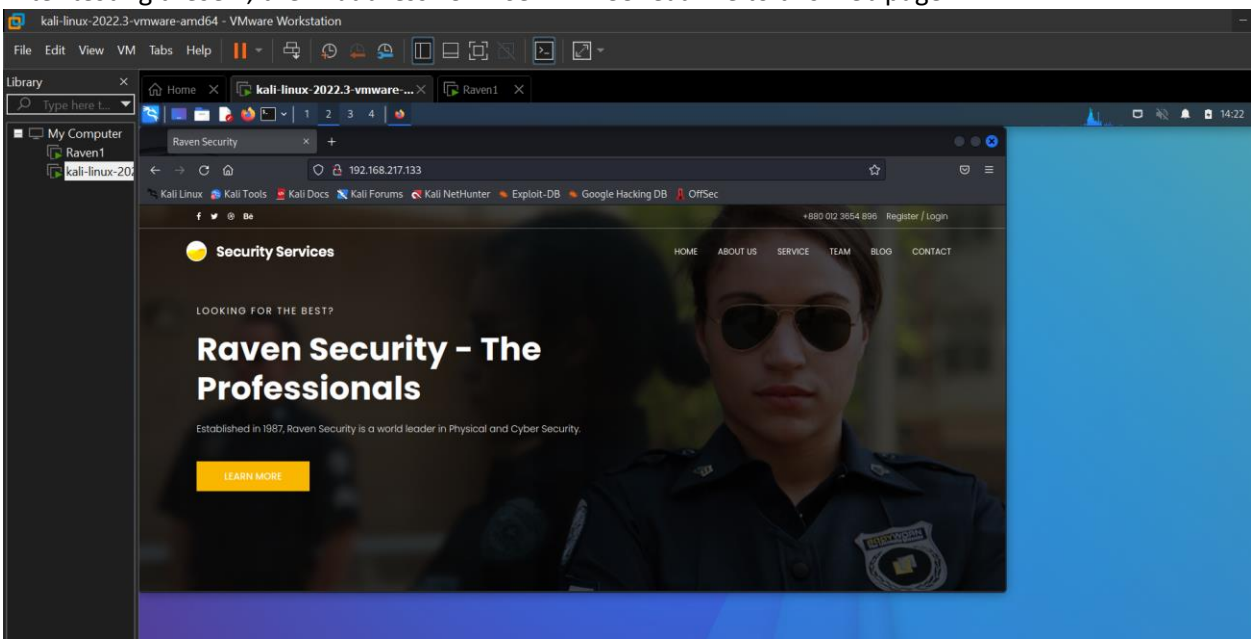


Raven was not working inside of the Kali Linux VM. After searching online for fixes, I decided to run Raven 1 in VMware workstation pro alongside Kali. This was a simple setup by clicking File>Open…>Raven.ova in VMWare. This is the end of the configuration; the following screenshots document the important and relevant steps to capturing the flags and attaining root.

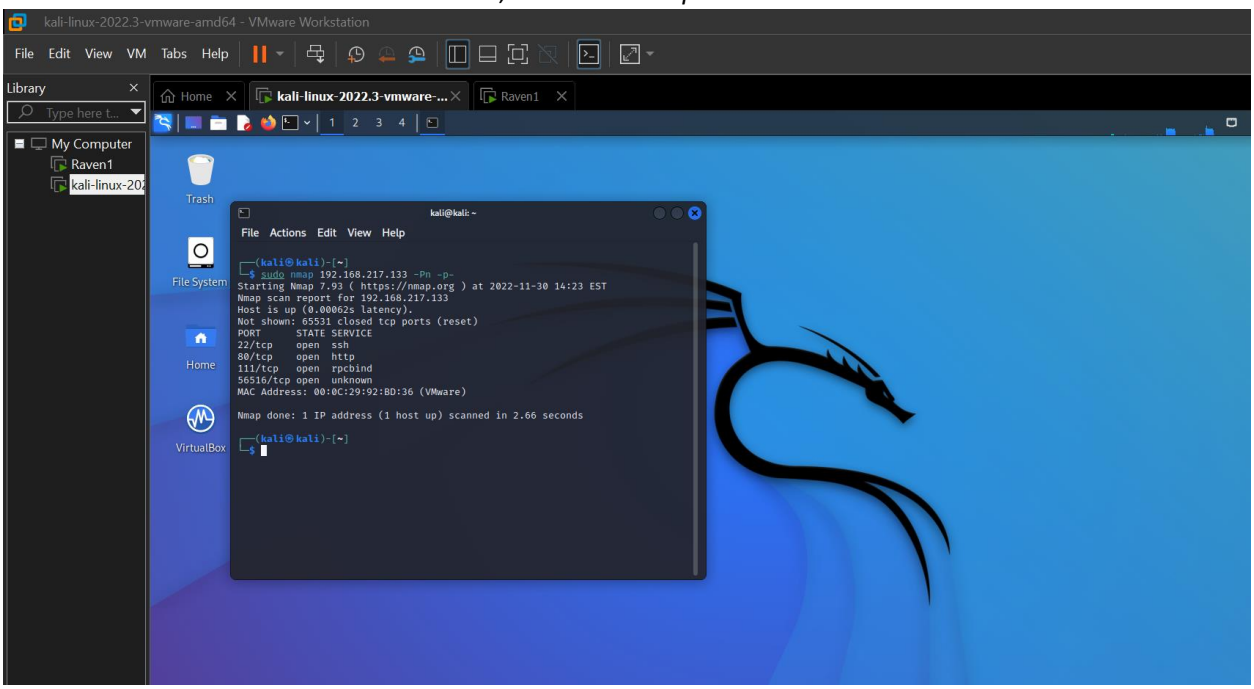
Running *netdiscover* command to find any IP:

```
 Currently scanning: 172.26.102.0/16   |   Screen View: Unique Hosts

 6 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 360
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 _____________________________________________________________________________
 192.168.217.1   00:50:56:c0:00:08      1      60  VMware, Inc.
 192.168.217.2   00:50:56:e3:71:7f      2     120  VMware, Inc.
 192.168.217.133 00:0c:29:92:bd:36      2     120  VMware, Inc.
 192.168.217.254 00:50:56:fe:d1:34      1      60  VMware, Inc.
```

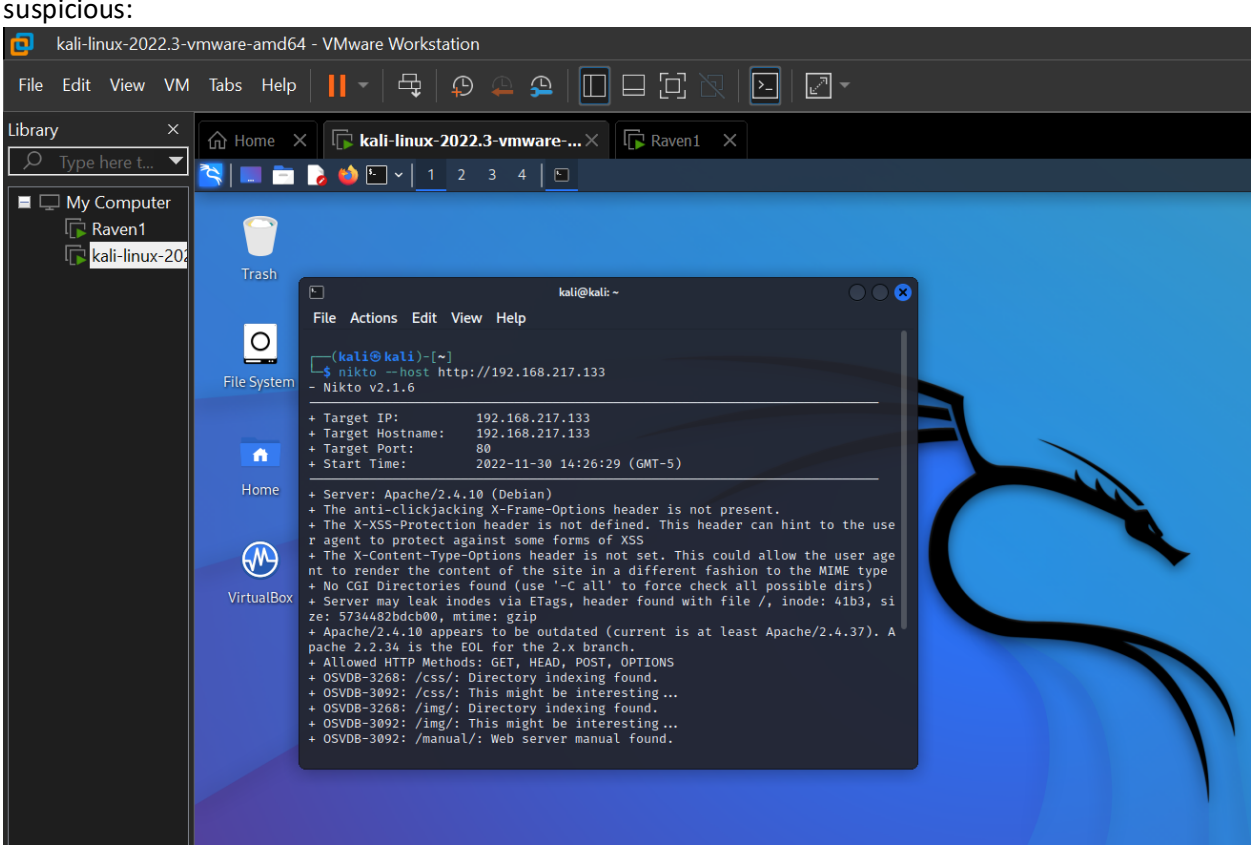
After testing these 4, the IP address *192.168.217.133* lead me to this web page:



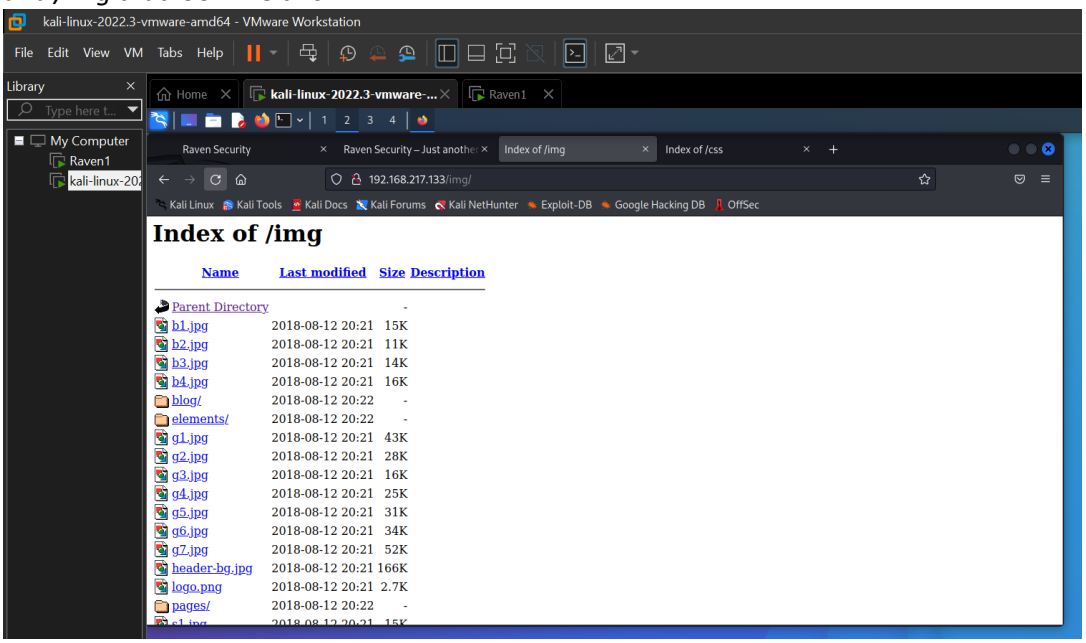
I want to find more information on this IP, so I run a *nmap* command:

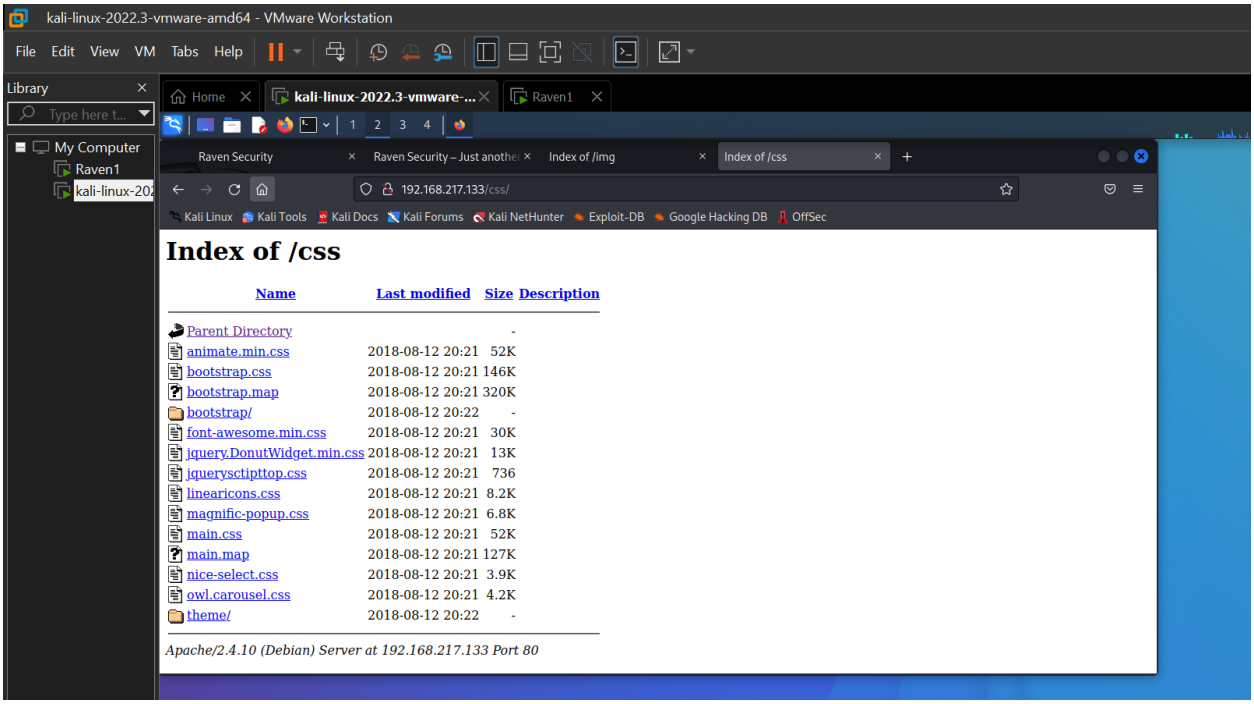


From the *nmap* command, I find four open ports. Beginning with the website (http port), I cannot find anything of value from the home page right away. I run a *nikto* scan on the IP to look for anything suspicious:

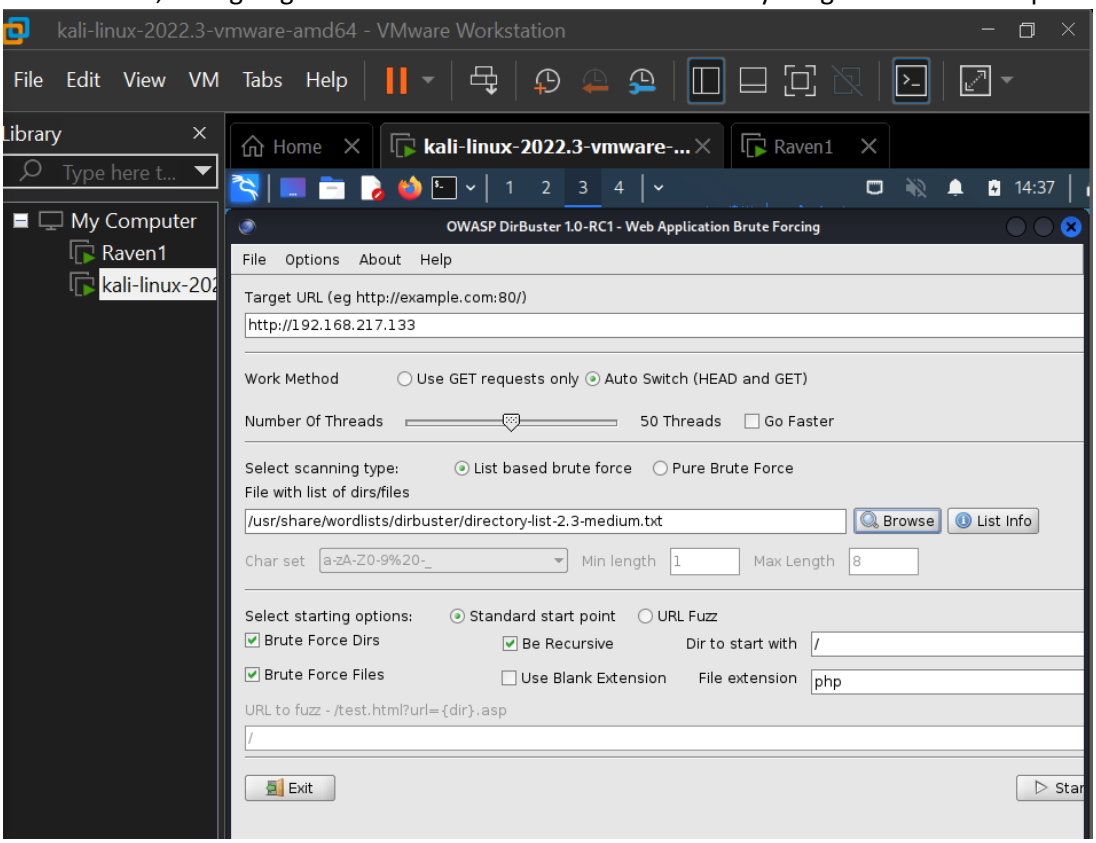


Nikto is a powerful tool that is an open-source web server and application scanner that looks for security threats. This scan provided no immediate results, however, there are some directories listed like */css* and */img* that look like this:

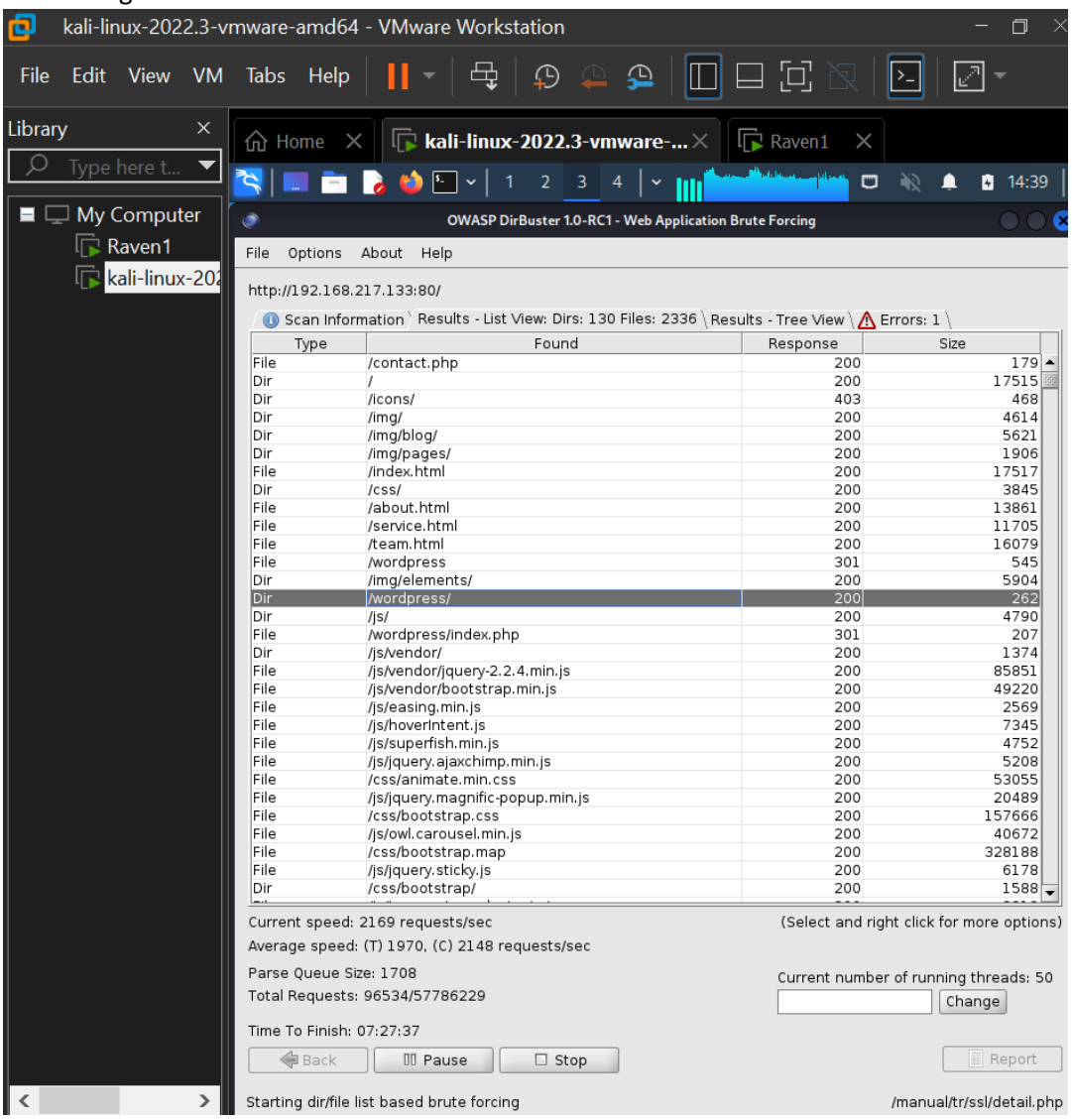




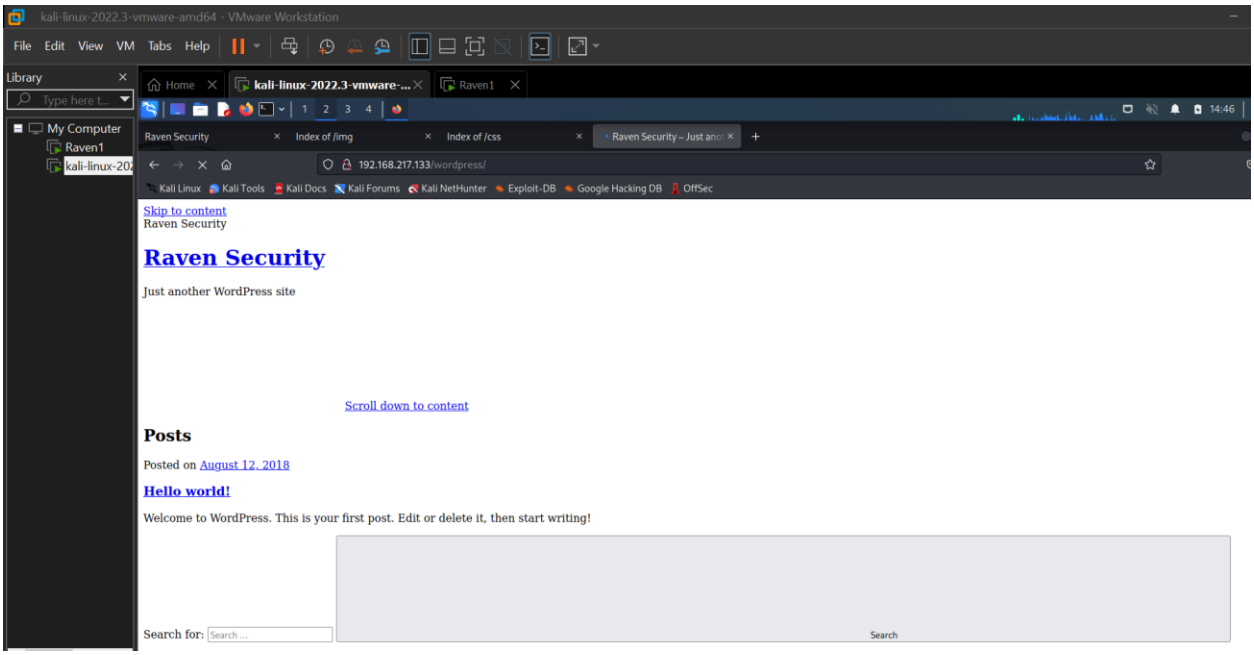
Furthermore, I am going to run a *dirbuster* on the IP to see if anything else can come up:



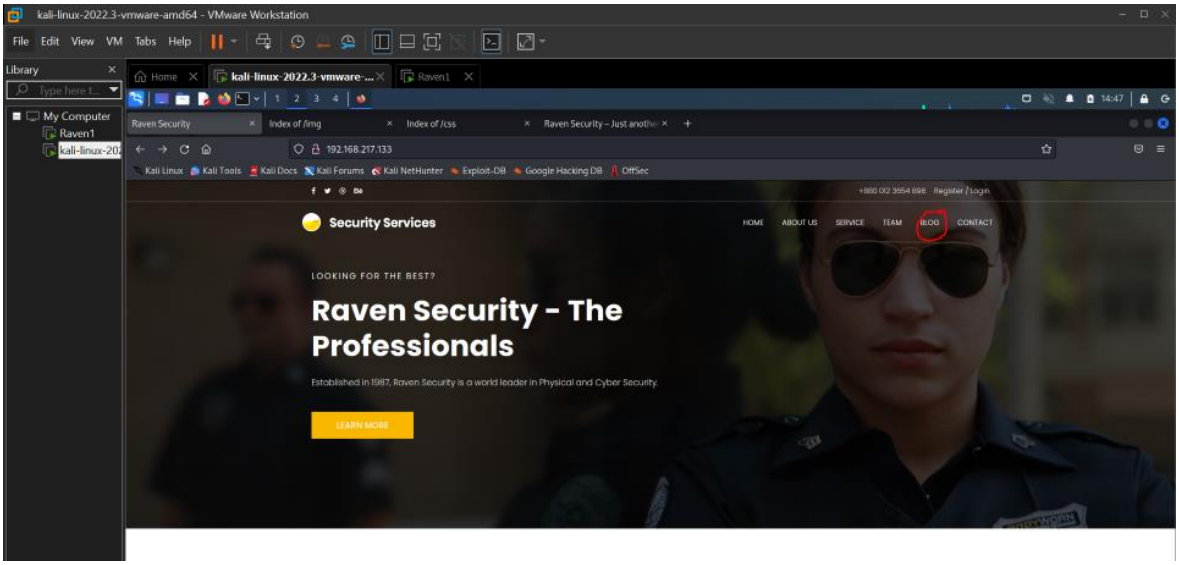
Interesting file found:



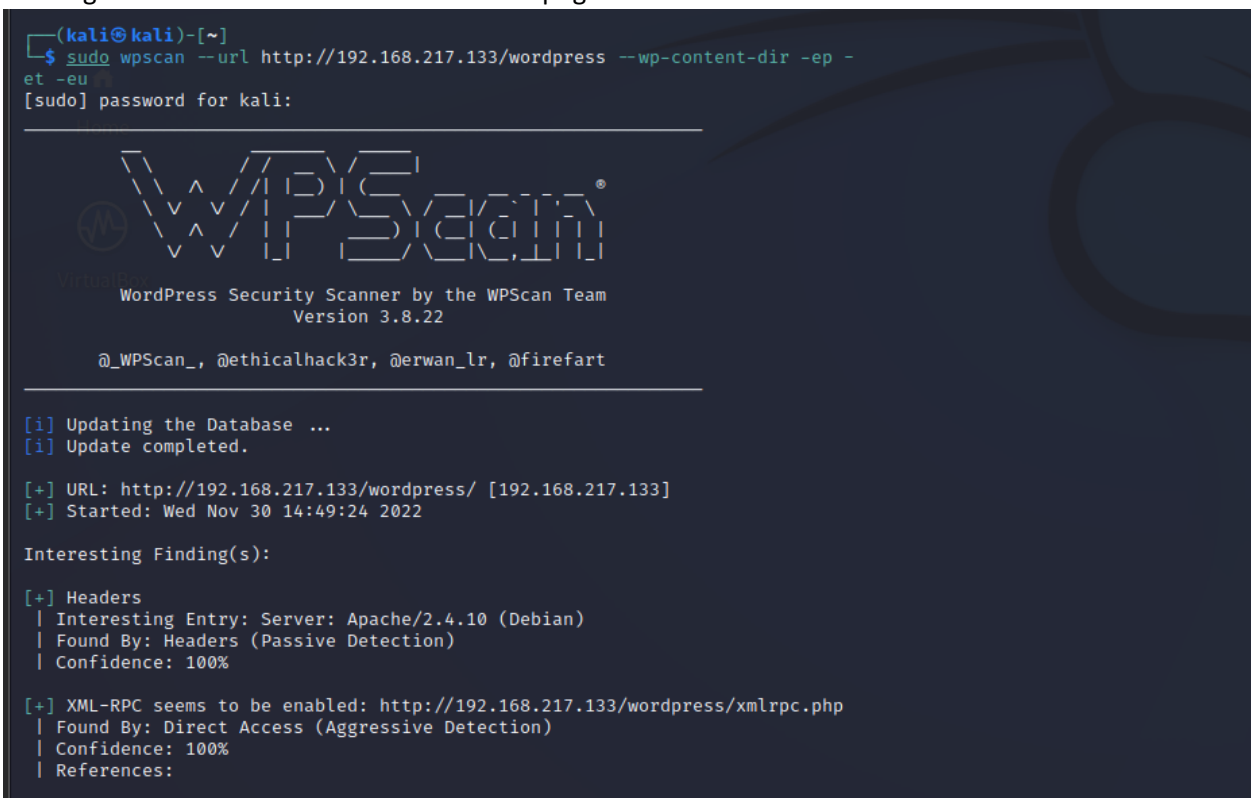
WordPress has now been found within the IP address for the Raven homepage. Knowing this, I try to access it:



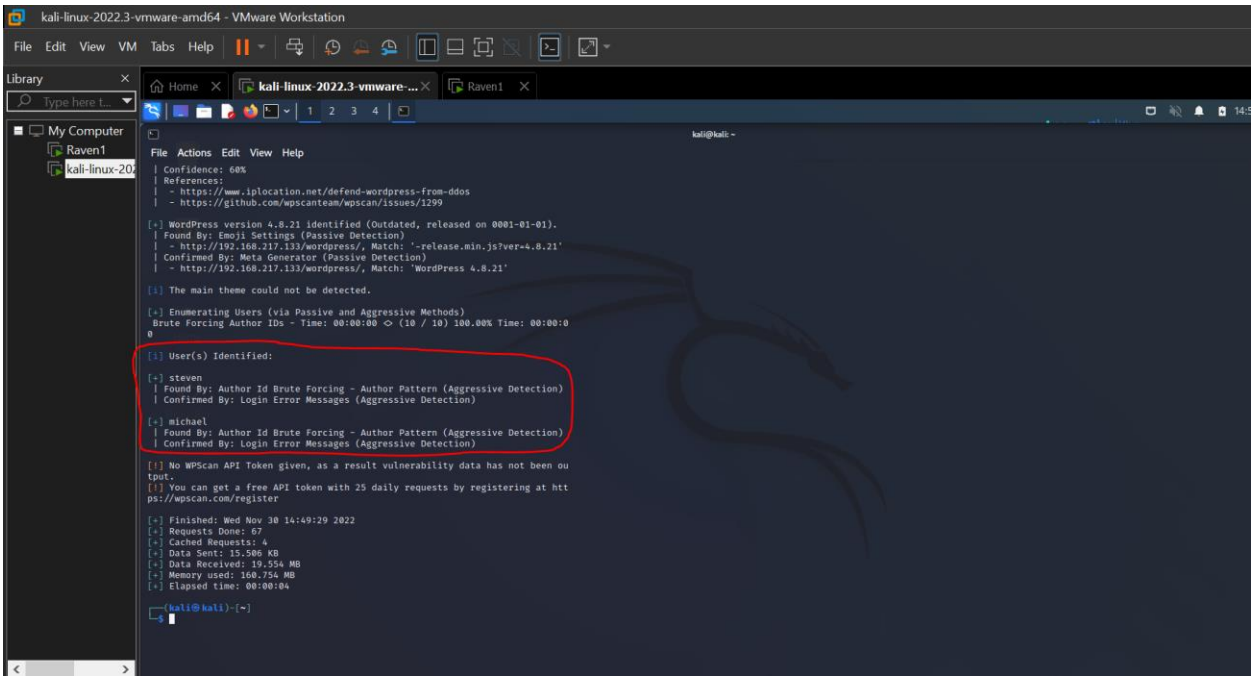
I find that the “Blog” button on the home page also leads to this WordPress page:



Starting a WordPress scan on the WordPress page:

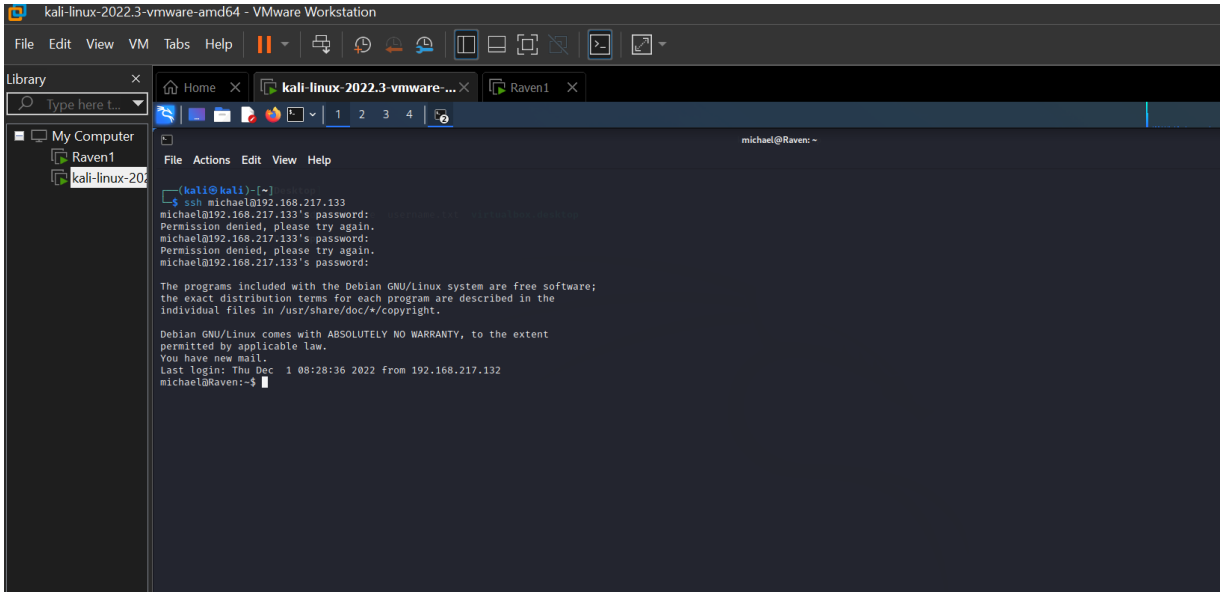


This scan should identify vulnerabilities in WordPress installations. After this scan, I find users steven and michael:

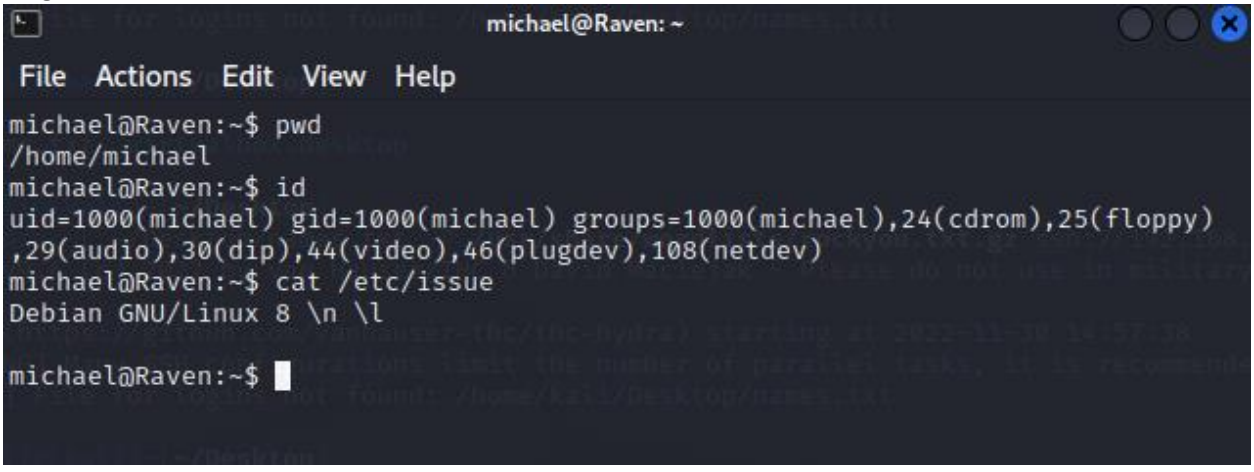


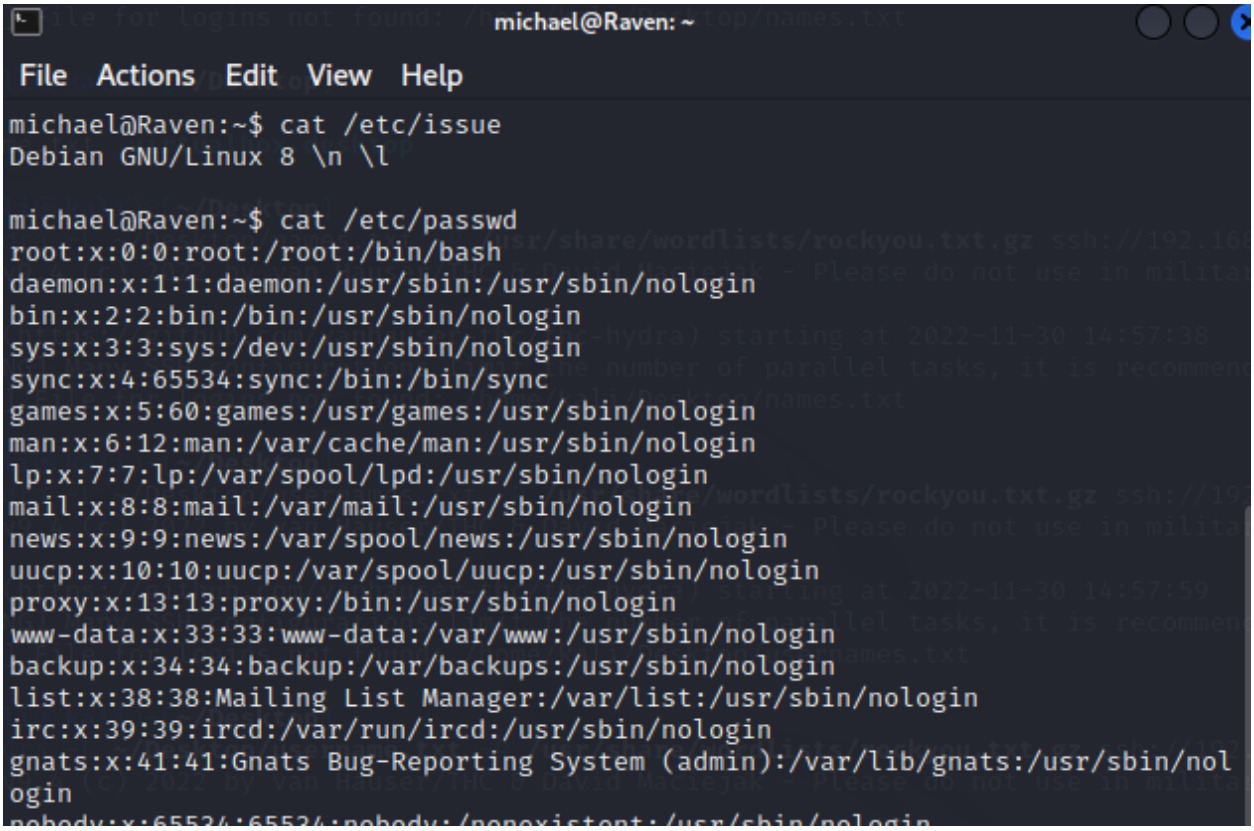
With this information, I am going to try and login to their accounts. The first step for this is to find a password:

From the wpscan, I am going to try guessing a few simple passwords to login to a user while the tool “Hydra” is running on the target. These were ‘password’, ‘123’, and same as username. This became time consuming, but I got lucky with the user Michael as this user password was the same as the username “michael”:

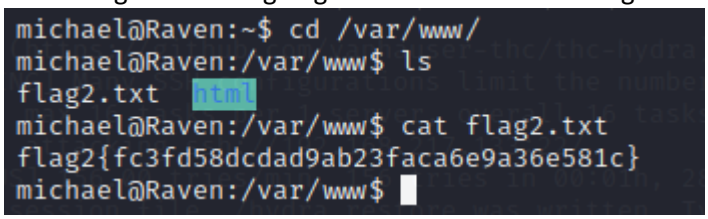


To gain some information, I use various commands and look into */etc/* files:





As Michael, I snoop around the directories like etc and var before finding a path */var/*www. This was interesting to me and going in here I find the first flag:

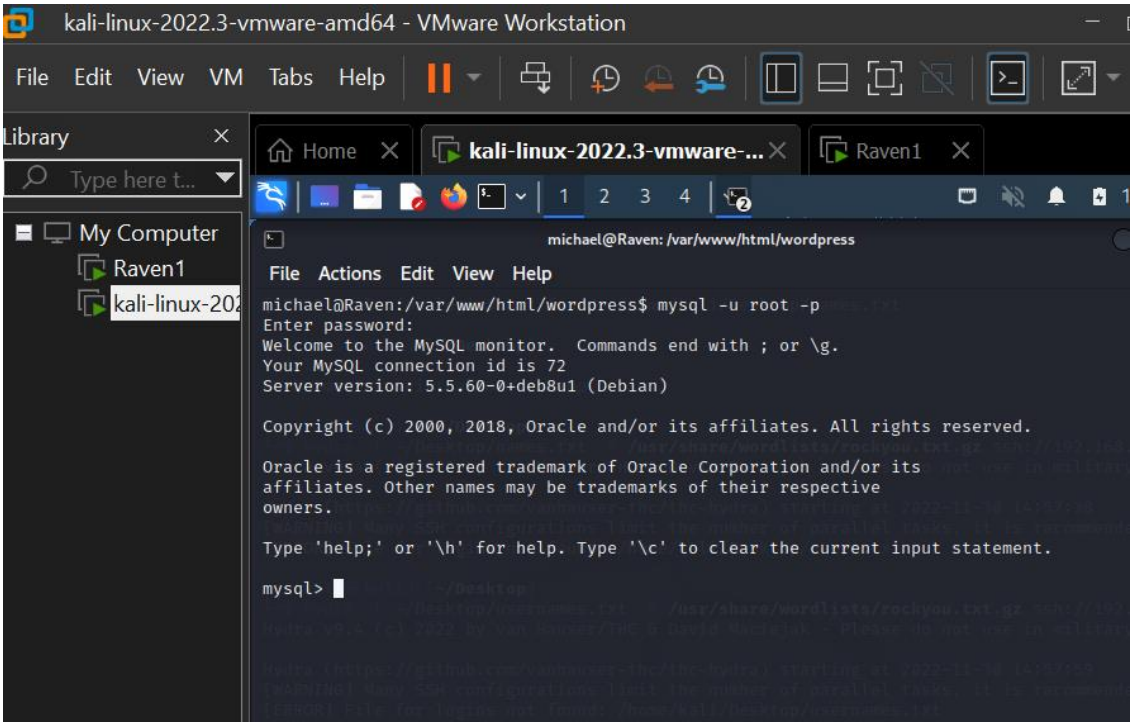


Once again, I begin snooping around in the directory */var/www* further to see if anything else can be found in here:

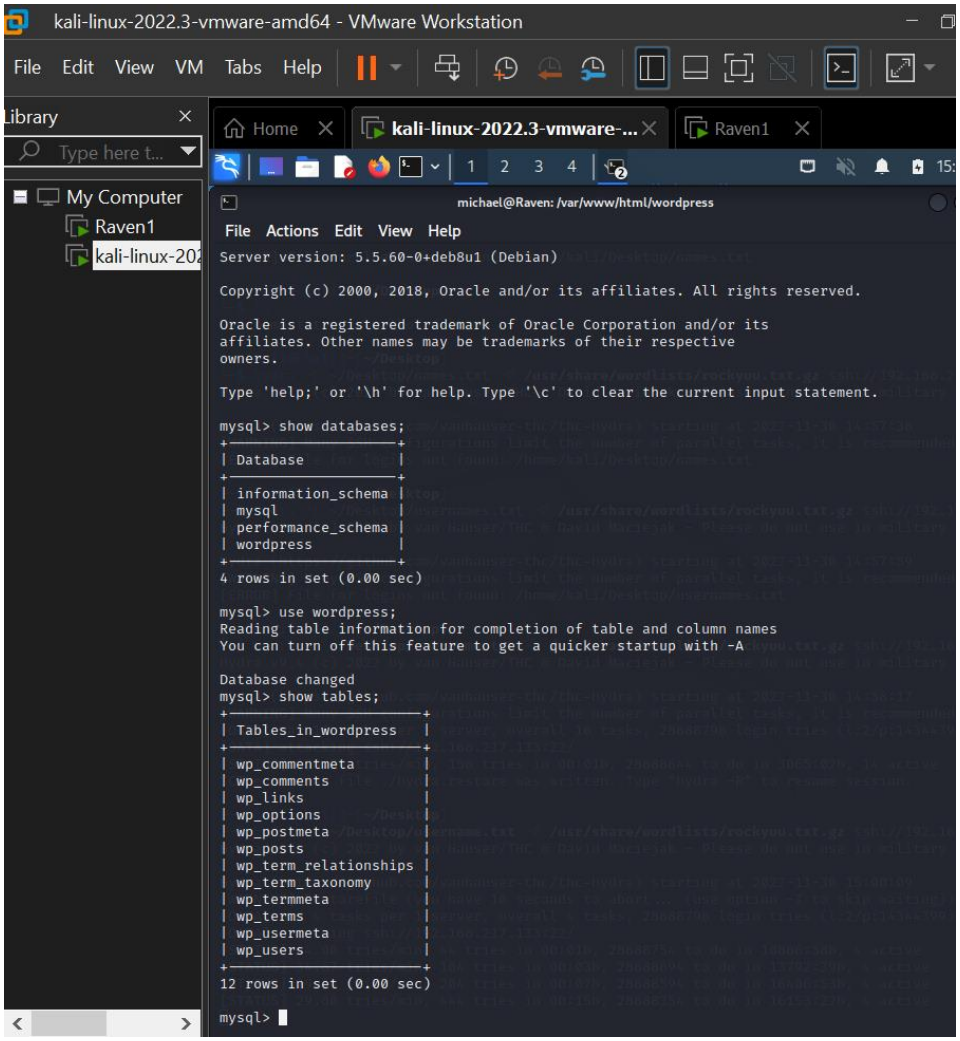
I find flag1 in the file *service.html*! Looking around in other files, I stumble across the username and password for the MySQL database in the directory */var/www/html/wordpress/wp-config.php*:

```
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
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** 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');
```

It appears that */var/*www was the jackpot. Knowing the username and password, I login to mysql:



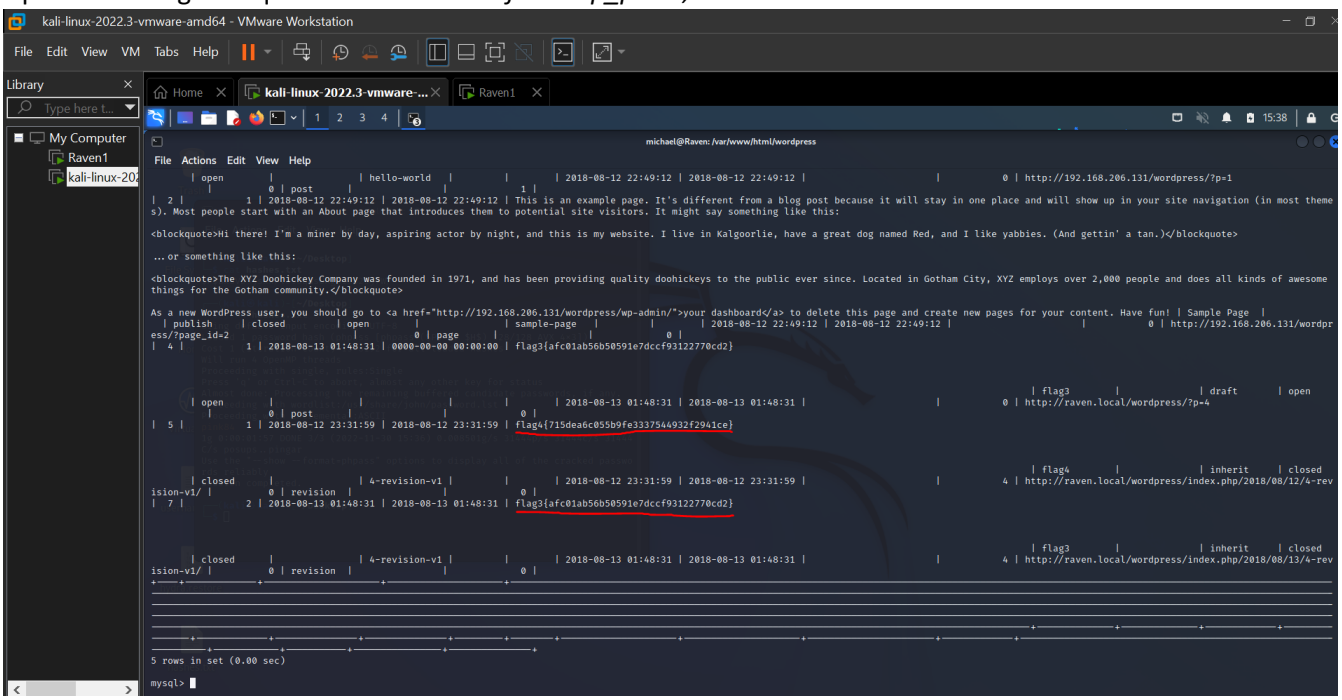
From here, I am looking for databases I can get information from. Using an SQL command, I find a wordpress database:



Inside of the wordpress database are a bunch of tables. My initial thought is to investigate *wp_users:*

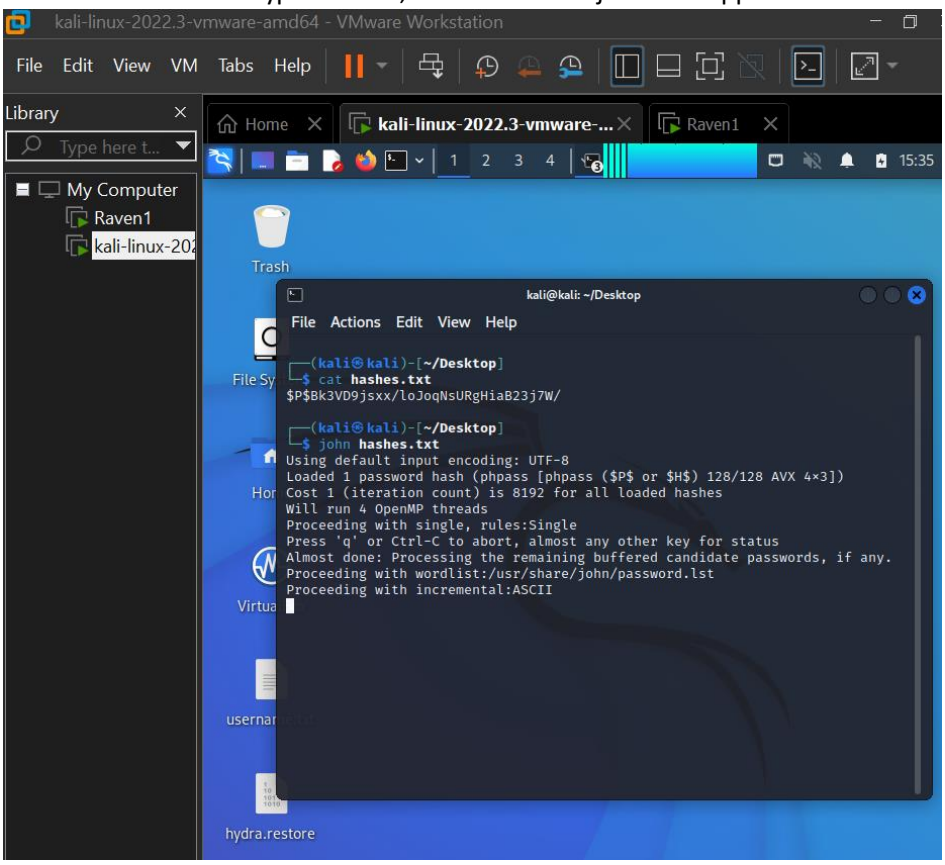
This is a good find, but I want to look at the other tables to see if there is anything else I can use since these are hashed. I copy and paste stevens hash (I already know michael's password) into a text document on the Desktop. Checking through the tables one by one, I finally come across *wp_posts*.

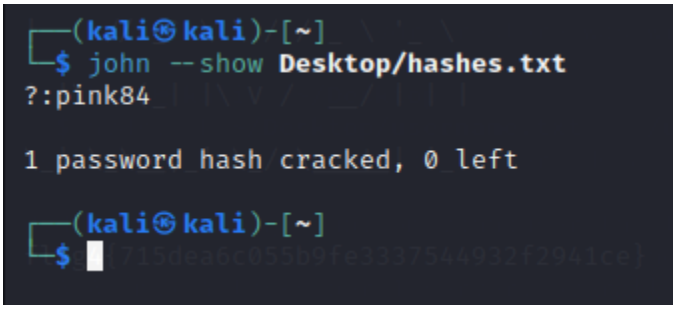
Upon executing the sql command *select * from wp_posts;* I come across this information:



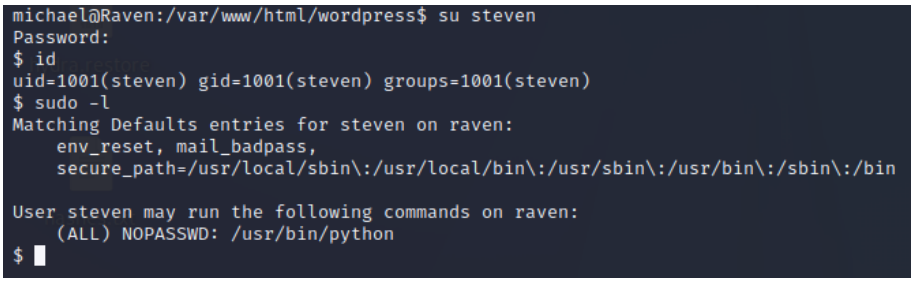
In the above screenshot, flag3 and flag4 can be seen from the *wp_posts* table. Now that I have all 4 flags, I need to think about getting root access to finish Raven. For this, I want to use the user steven. I put steven's password hash into hash-identifier and see that it is MD5:

Now that I know the type of hash, I use the tool "john the ripper" to crack the hash (this takes a while):



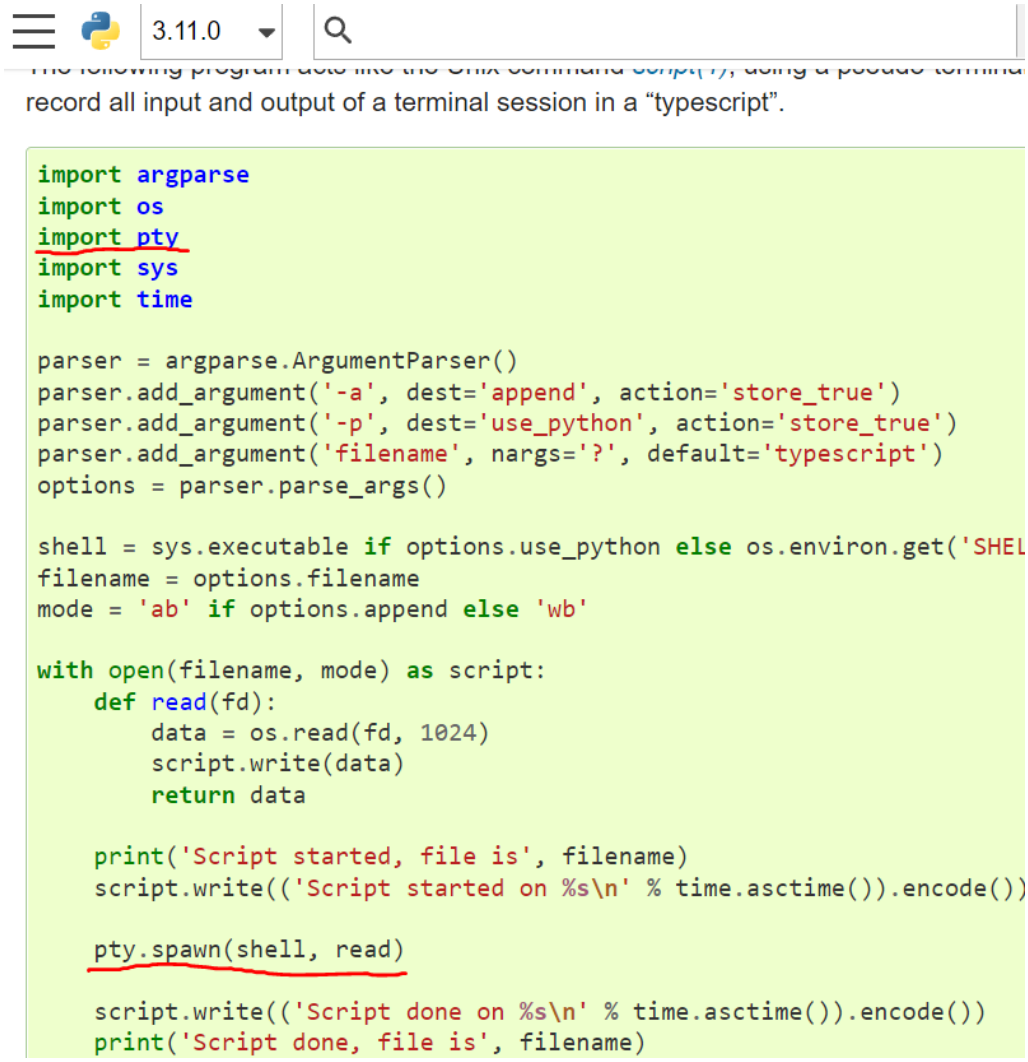


John successfully cracks stevens password and I login as this user. As steven, I check id and -/ to see if I can do anything useful.



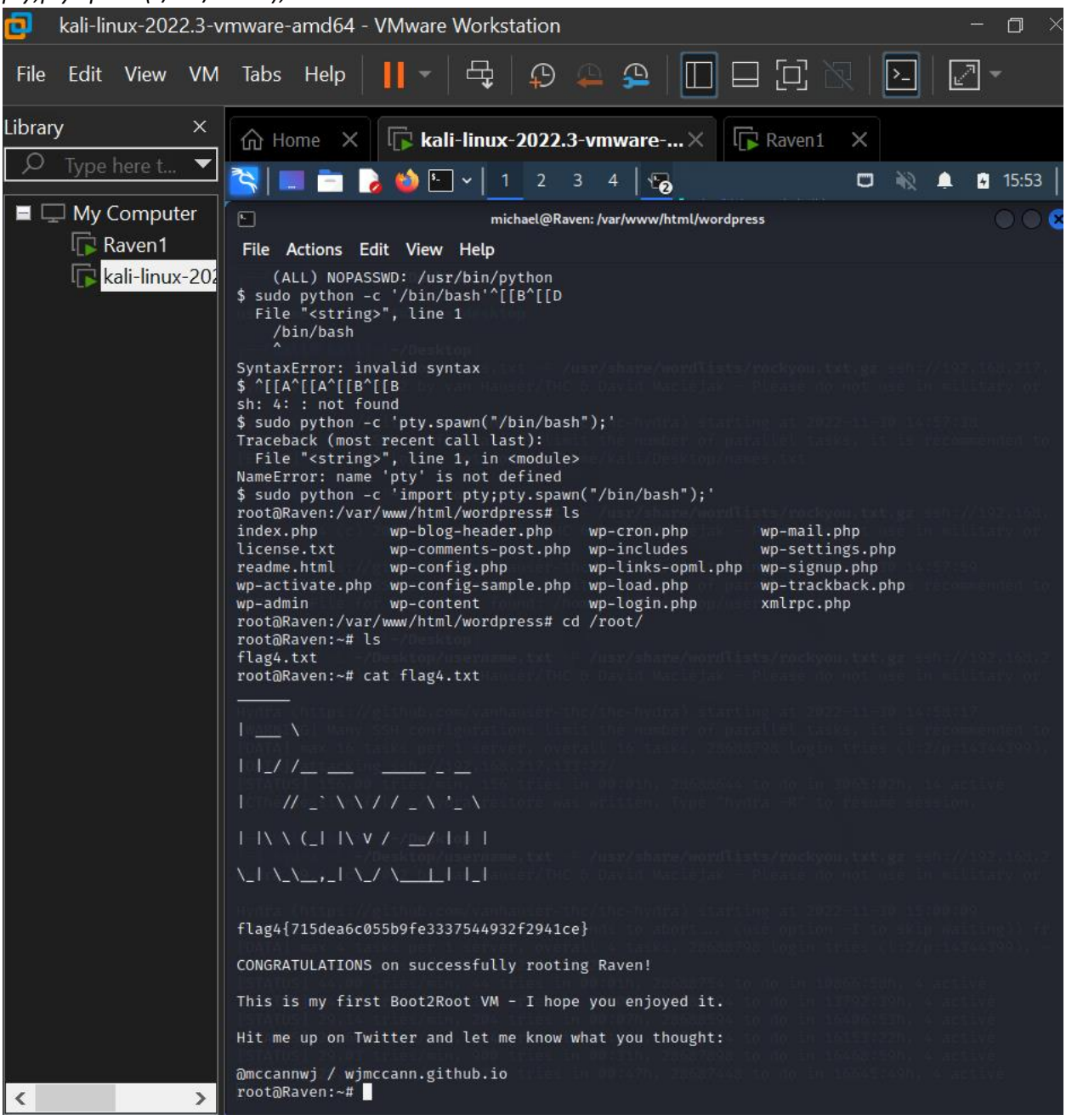
From these commands, I see that as steven, I can run python with sudo (NOPASSWD).

To come up with a python command to gain access, I am looking at documentation online. This took some attempts and a lot of web surfing. [9] from this website, I find some code snippets that stick out to me:



Pty.spawn immediately catches my eye as this looks like it spawns a shell using python. With this information, I try some commands before eventually finding the one that works *sudo python -c ‘import*

pty;pty.spawn("/bin/bash");'



```
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
root@Raven:~# id
uid=0(root) gid=0(root) groups=0(root)
root@Raven:~# sudo -l
Matching Defaults entries for root on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User root may run the following commands on raven:
    (ALL : ALL) ALL
root@Raven:~#
```

The final two screenshots here show that the python input as user steven was able to get a root shell in Raven. With this complete, the four flags have been obtained and root access has been found.

V. Programs and Scripts

Given in this section are the only documents providable in this project: flags.txt and hashes.txt. These files are text documents that include the four flags from Raven 1 as well as the hashes for the passwords of users “michael” and “steven” respectively.


flags.txt


hashes.txt

VI. References

1. [1] OffSec Services (2022). Netdiscover | Kali Linux Tools. https://www.kali.org/tools/netdiscover/
2. [2] JavaTpoint (2021). Nmap Commands in Kali Linux. https://www.javatpoint.com/nmap-commands-in-kali-linux#:~:text=In%20Kali%20Linux%2C%20Nmap%20means,schedules%2C%20host%20monitoring%2C%20etc.
3. [3] OffSec Services (2022). Dirbuster | Kali Linux Tools. https://www.kali.org/tools/dirbuster/
4. [4] OffSec Services (2022). Nikto | Kali Linux Tools. https://www.kali.org/tools/nikto/
5. [5] OffSec Services (2022). Wpscan | Kali Linux Tools. https://www.kali.org/tools/wpscan/

6. [6] OffSec Services (2022). Openssh | Kali Linux Tools. https://www.kali.org/tools/openssh/
7. [7] OffSec Services (2022). John | Kali Linux Tools. https://www.kali.org/tools/john/
8. [8] OffSec Services (2022). Hash-identifier | Kali Linux Tools. https://www.kali.org/tools/hash-identifier/
9. [9] Python Software Foundation (2022). Pty – Pseudo-terminal utilities. https://docs.python.org/3/library/pty.html
10. [10] OffSec Services (2022). Installing VirtualBox on Kali (Host). https://www.kali.org/docs/virtualization/install-virtualbox-host/