



NARFACE — INFORME TÉCNICO

Máquina Pickle Rick



Descargo de responsabilidad

Este informe contiene información confidencial y sensible destinada únicamente para el uso interno de la entidad. Cualquier acceso no autorizado despertará la furia del Pickle Rick xD.

25 de diciembre de 2023

Web: narface.github.io

Índice

1. Introducción	2
2. Objetivos	2
2.1. Consideraciones	2
3. Reconocimiento inicial del sistema	4
3.1. Escaneo y detección de vulnerabilidades	4
3.2. Resultados de Nikto	7
3.3. Resultados de Dirb	8
3.4. Gobuster	8
3.5. Resultados	9
3.5.1. Extracción de Información de Directorios	9
3.6. Análisis de Directorios	10
3.7. Resumen de Hallazgos	11
4. Explotación de vulnerabilidades	13
4.1. Prueba: Acceso al Portal	13
4.2. Primer ingrediente	17
4.3. Segundo ingrediente	18
4.4. Tercer ingrediente	21

1. Introducción

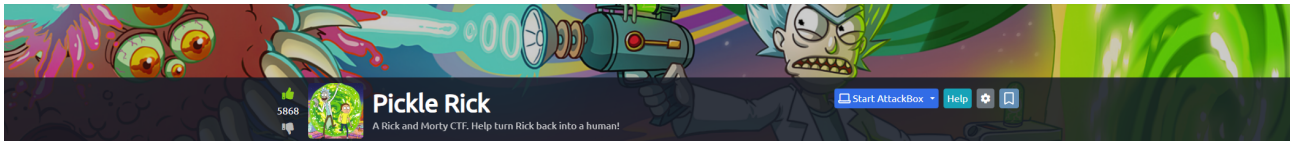


Figura 1: Cabecera de la máquina.

En este documento vamos a explicar cómo hemos resuelto el reto de la máquina **Pickle Rick** de la plataforma **Tryhackme**. Se trata de un reto temático basado en la serie de animación Rick y Morty, en el que tenemos que ayudar a Rick a volver a su forma humana después de que se haya convertido en un pepinillo. Para ello, tenemos que acceder a un servidor web vulnerable, encontrar tres ingredientes secretos que forman parte de la poción de Rick y escapar de la guarida del malvado científico. El reto es de nivel fácil y está pensado para principiantes en el hacking ético. Las herramientas y los recursos que he utilizado para resolverlo son los siguientes:

- Herramientas de escaneo:
 1. Nmap: un escáner de puertos y servicios de red.
 2. Nikto: un escáner de vulnerabilidades web.
- Herramientas de enumeración:
 1. Dirb: un buscador de directorios y archivos web.
 2. Gobuster: una herramienta de enumeración que busca directorios y archivos ocultos en servidores web.
- Herramientas de visualización:
 1. Less: una herramienta para visualizar y navegar a través de grandes archivos de texto en la terminal de forma eficiente.

Dirección URL

<https://tryhackme.com/room/picklerick>

2. Objetivos

El objetivo general es conocer el estado de seguridad actual del servidor **Pickle Rick**, que presenta varias vulnerabilidades que permiten el acceso no autorizado y la ejecución de comandos. enumerando posibles vectores de explotación y determinando el alcance e impacto que un atacante podría ocasionar sobre el sistema.

2.1. Consideraciones

Las limitaciones encontradas durante el desarrollo de este desafío incluyeron restricciones de tiempo, alcance y herramientas disponibles. Se abordó la resolución dentro de un límite temporal específico, lo que limitó la profundización en otras posibles vías de ataque o escalada de privilegios. El enfoque se centró en vulnerabilidades más evidentes y simples, omitiendo la exploración de métodos más complejos o sutiles. Además, se optó por el uso de herramientas manuales y básicas en lugar de recursos automatizados o scripts predefinidos.

- Nota:
 1. Se debe tener en cuenta que durante el desarrollo de la actividad, se experimentó una caída del sistema, lo que resultó en un reinicio. Por ende, algunas capturas de pantalla o referencias pueden mostrar una IP diferente a la originalmente asignada.

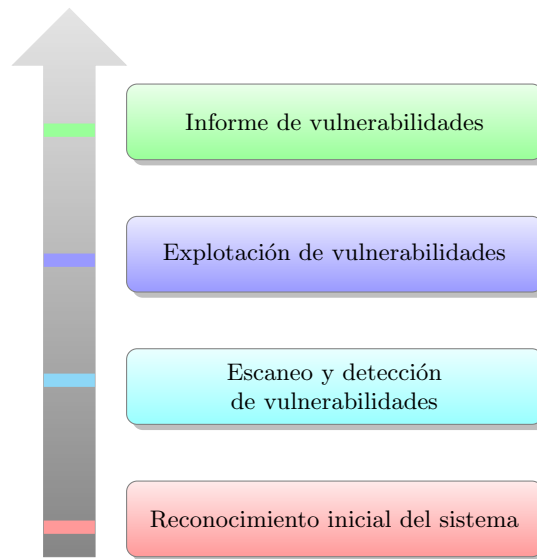


Figura 2: Flujo de trabajo para el análisis de vulnerabilidades

3. Reconocimiento inicial del sistema

El análisis de vulnerabilidades es el proceso de identificar y evaluar las debilidades o fallos de seguridad que presenta un sistema informático, con el fin de prevenir o mitigar posibles ataques. Para realizar este análisis, se ha seguido una metodología basada en las fases de reconocimiento, escaneo, explotación y reporte.

3.1. Escaneo y detección de vulnerabilidades

El objetivo de esta fase es obtener una visión general del sistema y detectar posibles puntos de entrada o vectores de ataque.

Se realizó un escaneo a través de la herramienta **nmap** para la detección de puertos abiertos:

```
root@ip-10-10-35-212:~# nmap -sV -sC -oA nmap_test 10.10.214.221

Starting Nmap 7.60 ( https://nmap.org ) at 2023-12-23 23:19 GMT
Nmap scan report for ip-10-10-214-221.eu-west-1.compute.internal (10.10.214.221)
Host is up (0.047s 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 d2:a1:50:71:d2:4d:f9:98:76:54:c0:ab:be:f8:94:10 (RSA)
|_  256 47:85:c0:2a:b5:3b:8a:52:e9:d7:04:32:89:a0:c5:fb (ECDSA)
|_  256 42:79:45:52:c9:e5:cc:25:d1:cc:0d:bb:e9:e4:6a:f1 (EdDSA)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Rick is sup4r cool
MAC Address: 02:3D:73:A3:A9:9D (Unknown)
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 10.98 seconds
```

Figura 3: Reconocimiento con nmap.

Nmap Scan Report - Scanned at Sat Dec 23 23:19:48 2023

Scan Summary | [ip-10-10-214-221.eu-west-1.compute.internal \(10.10.214.221\)](#)

Scan Summary

Nmap 7.60 was initiated at Sat Dec 23 23:19:48 2023 with these arguments:
nmap -sV -sC -oA nmap_test 10.10.214.221

Verbosity: 0; Debug level 0

Nmap done at Sat Dec 23 23:19:59 2023; 1 IP address (1 host up) scanned in 10.98 seconds

10.10.214.221 / ip-10-10-214-221.eu-west-1.compute.internal

Address

- 10.10.214.221 (ipv4)
- 02:3D:73:A3:A9:9D (mac)

Hostnames

- ip-10-10-214-221.eu-west-1.compute.internal (PTR)

Ports

The 998 ports scanned but not shown below are in state: **closed**

- 998 ports replied with: **resets**

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra Info
22/tcp	open	ssh	syn-ack	OpenSSH	7.2p2 Ubuntu 4ubuntu2.6	Ubuntu Linux; protocol 2.0
ssh-hostkey						2048 d2:a1:50:71:d2:4d:f9:98:76:54:c0:ab:be:f8:94:10 (RSA)
						256 47:85:c0:2a:b5:3b:8a:52:e9:d7:04:32:89:a0:c5:fb (ECDSA)
						256 42:79:45:52:c9:e5:cc:25:d1:cc:0d:bb:e9:e4:6a:f1 (EdDSA)
80/tcp	open	http	syn-ack	Apache httpd	2.4.18	(Ubuntu)
http-server-header						Apache/2.4.18 (Ubuntu)
http-title						Rick is sup4r cool

Misc Metrics (click to expand)

Metric	Value
Ping Results	arp-response

Figura 4: Resultados nmap en formato HTML.

```
nmap -sV -sC -oA nmap_test 10.10.214.221
```

Estos resultados indican que la máquina **Pickle Rick** tiene dos puertos abiertos:

TCP
Puertos
22, 80

El puerto 22 corresponde al servicio de SSH, que permite el acceso remoto al sistema mediante un protocolo seguro. El puerto 80 corresponde al servicio de HTTP, que permite el acceso a una página web mediante un protocolo de transferencia de hipertexto. Ambos servicios están ejecutando versiones de software que podrían tener vulnerabilidades conocidas, se tratan de las versiones OpenSSH 7.2p2 y Apache httpd 2.4.18. Asimismo, la página web tiene un título que hace referencia a Rick, el personaje de la serie de animación Rick y Morty.

Se ha accedido a la web de la máquina **Pickle Rick**:

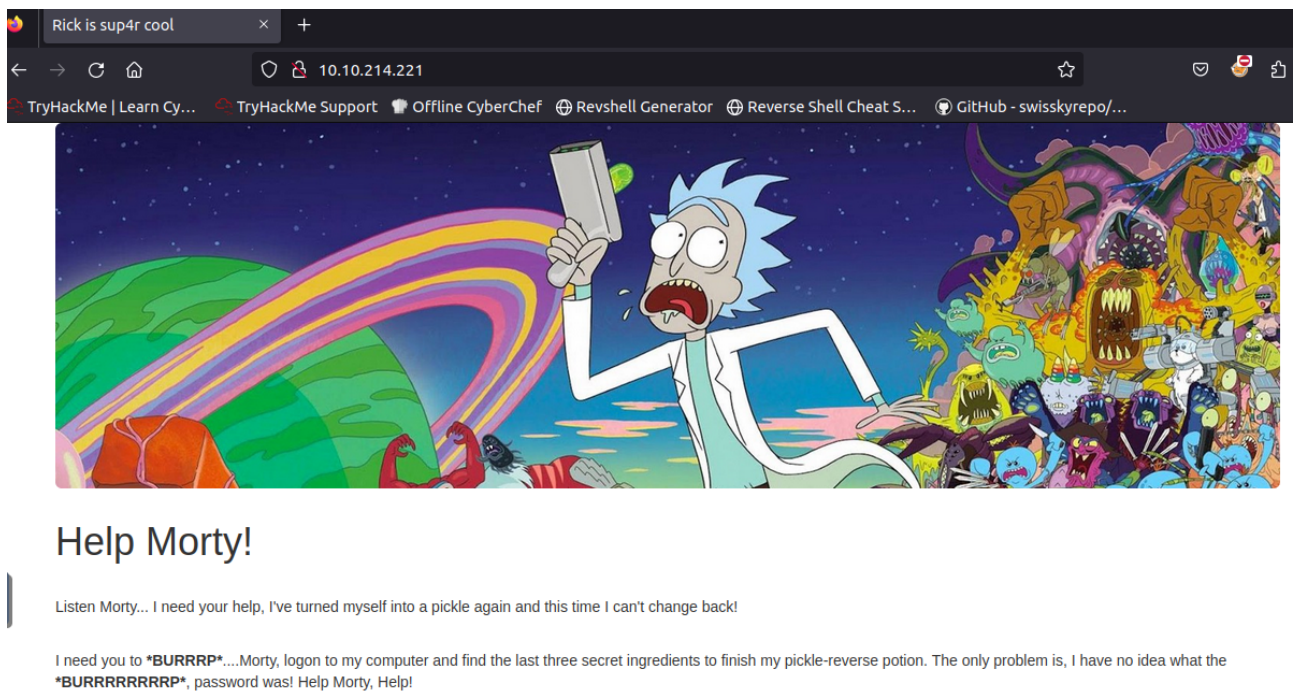
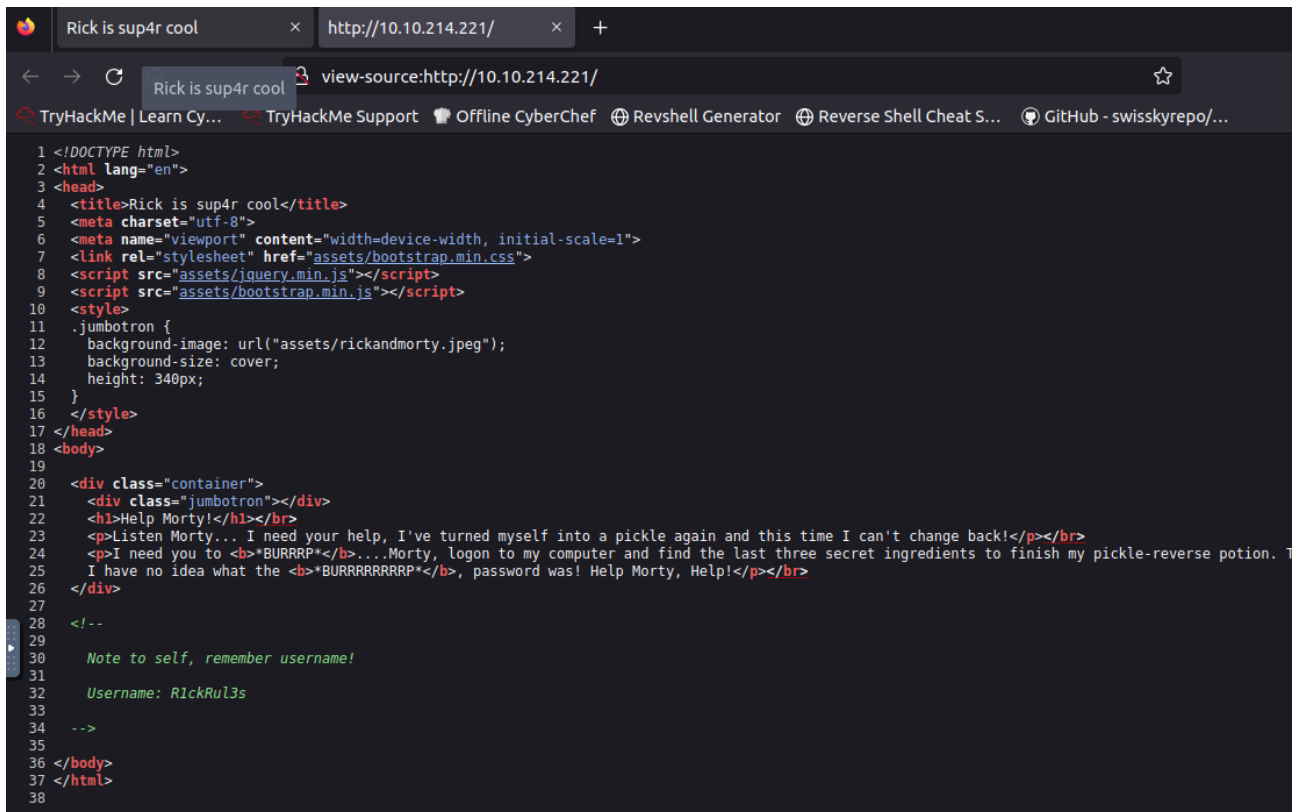


Figura 5: Web de la máquina **Pickle Rick**

Este mensaje es una parodia de la serie de animación Rick y Morty, en este capítulo, Rick se convierte en un pepinillo para evitar ir a terapia familiar con su nieto Morty. En el episodio "Pickle Rick", Rick necesita que Morty le ayude a recuperar una jeringa con una poción que le permita volver a su forma humana, pero Morty se va con su familia a la terapia y Rick tiene que enfrentarse a varios peligros como un pepinillo.

Tal y como se aprecia en la figura 6 de la página 6, se puede observar el mensaje que Rick le deja a Morty en la web de la máquina **Pickle Rick**. Para acceder al código fuente de la web, se ha usado la opción "Ver código fuente de la página" o CTRL + U del navegador.



```

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>BURRED*!</b>...Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion.
25     I have no idea what the <b>BURRED*!</b>, password was! Help Morty, Help!</p></br>
26   </div>
27
28   <!--
29     Note to self, remember username!
30     Username: RickRu13s
31   -->
32
33 </body>
34 </html>

```

Figura 6: Web de la máquina Pickle Rick

Se ha analizado el código de la web, identificando los elementos más relevantes de la Figura 6 y explicando qué hacen:

- La etiqueta `<title>` define el título de la página, que en este caso es "Rick is sup4r cool".
- La etiqueta `<meta>` define la información sobre la página, como el tipo de caracteres que usa (`utf-8`) o el tamaño de la ventana que se adapta al dispositivo (`width=device-width, initial-scale=1`).
- La etiqueta `<link>` enlaza la página con otros archivos, como el archivo "assets/bootstrap.min.css" que contiene el estilo de la página basado en el framework Bootstrap.
- La etiqueta `<script>` carga los archivos de código JavaScript que hacen que la página sea interactiva, como los archivos "assets/jquery.min.js" y "assets/bootstrap.min.js" que son librerías de JavaScript que facilitan el uso de Bootstrap.
- La etiqueta `<style>` define el estilo de algunos elementos de la página, como el elemento con la clase "jumbotron" que tiene una imagen de fondo ("assets/rickandmorty.jpeg") que ocupa todo el ancho y tiene una altura de 340 píxeles.
- La etiqueta `<div>` crea un contenedor para agrupar otros elementos, como el elemento con la clase "container" que contiene el mensaje de Rick y el elemento con la clase "jumbotron" que contiene la imagen de fondo.
- La etiqueta `<h1>` crea un encabezado de nivel 1, que en este caso es "Help Morty!".
- La etiqueta `<p>` crea un párrafo de texto, que en este caso contiene el mensaje de Rick dividido en dos partes.
- La etiqueta `` hace que el texto sea negrita, que en este caso se usa para resaltar los eructos de Rick.

- La etiqueta `<!-- -->` crea un comentario que no se muestra en la página, que en este caso contiene una nota de Rick para recordar su nombre de usuario, que es **"R1ckRul3s"**.

3.2. Resultados de Nikto

Nikto es una herramienta de escaneo de vulnerabilidades web utilizada para detectar problemas de seguridad en páginas web. Permite identificar archivos o directorios accesibles, configuraciones erróneas, versiones desactualizadas de software, entre otros. En el contexto de la máquina **Pickle Rick**, Nikto puede analizar la página web en busca de posibles puntos de entrada o vectores de ataque.

Para ejecutar Nikto, se requiere la dirección IP y el puerto del sitio web objetivo. El siguiente comando muestra cómo usarlo:

```
nikto -h 10.10.214.221 -p 80
```

Donde `-h` indica la dirección IP y `-p` el puerto.

```
root@ip-10-10-35-212:~# nikto -h 10.10.214.221 -p 80
- Nikto v2.1.5
-----
+ Target IP:      10.10.214.221
+ Target Hostname: ip-10-10-214-221.eu-west-1.compute.internal
+ Target Port:    80
+ Start Time:     2023-12-23 23:35:08 (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: GET, HEAD, POST, OPTIONS
+ 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:       2023-12-23 23:35:18 (GMT0) (10 seconds)
-----
+ 1 host(s) tested
```

Figura 7: Ejecución de Nikto.

Los hallazgos revelan debilidades de seguridad en la página web de la máquina **Pickle Rick**:

- Ausencia de cabeceras para prevenir ataques de clickjacking, XSS o inyección de contenido.
- Directorio `/assets/` permitido en `robots.txt`, posiblemente con información relevante.
- Archivo `README` en el directorio `/icons/`, el cual podría revelar detalles del servidor.
- Muestra de métodos HTTP permitidos: GET, HEAD, POST y OPTIONS, lo que podría facilitar la inyección de parámetros o datos en las solicitudes.

Hallazgos de Nikto

- **Servidor:** Apache/2.4.18 (Ubuntu).
- **Vulnerabilidad de ETags:** Inclusión de inodes a través de ETags.
- **X-Frame-Options ausente:** Riesgo de clickjacking.
- **robots.txt sin 'disallow':** Ausencia de restricciones en el archivo.
- **Archivos por defecto de Apache:**
 - Archivo `README` encontrado en `/icons/`.
 - Página de inicio de sesión de administrador en `/login.php`.
- **Métodos HTTP permitidos:** OPTIONS, GET, HEAD, POST.

3.3. Resultados de Dirb

Los resultados del escaneo realizado con Dirb se detallan a continuación:

Hallazgos de Dirb

```
root@ip-10-10-35-212:~# dirb http://10.10.214.221 /usr/share/wordlists/dirb/common.txt

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sat Dec 23 23:42:21 2023
URL_BASE: http://10.10.214.221/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.214.221/ ----
==> DIRECTORY: http://10.10.214.221/assets/
+ http://10.10.214.221/index.html (CODE:200|SIZE:1062)
+ http://10.10.214.221/robots.txt (CODE:200|SIZE:17)
+ http://10.10.214.221/server-status (CODE:403|SIZE:301)

---- Entering directory: http://10.10.214.221/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----

END_TIME: Sat Dec 23 23:42:24 2023
DOWNLOADED: 4612 - FOUND: 3
```

Resumen de hallazgos:

- Se confirma la presencia del directorio `/assets/` y del archivo `robots.txt`, previamente detectados por Nikto.
- Se identifica un archivo `index.html`, la página principal de la web.
- Se localiza un archivo `server-status` que está prohibido, posiblemente conteniendo detalles sobre el estado del servidor Apache.
- No se encuentran otros archivos o directorios ocultos o no listados en la web.

3.4. Gobuster

Gobuster es una herramienta de enumeración que permite buscar y enumerar directorios y ficheros presentes en un servidor web. Para realizar un escaneo de directorios en la máquina **Pickle Rick** con Gobuster, se utilizó el siguiente comando:

```
1 gobuster dir -u http://10.10.214.221 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x php,sh,txt,cgi,html,css,js,py
```

- `dir`: indica que se realizará un escaneo de directorios.
- `-u http://10.10.214.221`: especifica la URL del sitio web objetivo.
- `-w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt`: Especifica la lista de palabras que se utilizarán para realizar el escaneo. En este caso, se utiliza el archivo de listado de directorios `directory-list-lowercase-2.3-medium.txt`, que contiene una lista de palabras comúnmente utilizadas para nombres de directorios en minúsculas y de tamaño medio.
- `-x php,sh,txt,cgi,html,css,js,py`: especifica las extensiones de archivo que se incluirán en la búsqueda. En este caso, se buscan archivos con extensiones `php`, `sh`, `txt`, `cgi`, `html`, `css`, `js` y `p`.

```
root@ip-10-10-35-212:~# gobuster dir -u http://10.10.214.221 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x php,sh,txt,cgi,html,css,js,py
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:             http://10.10.214.221
[+] Threads:         10
[+] Wordlist:         /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Status codes:    200,204,301,302,307,401,403
[+] User Agent:       gobuster/3.0.1
[+] Extensions:     .php,.sh
[+] Timeout:         10s
=====
2023/12/23 23:11:57 Starting gobuster
=====
/login.php (Status: 200)
/assets (Status: 301)
/portal.php (Status: 302)
/denied.php (Status: 302)
/server-status (Status: 403)
=====
2023/12/23 23:23:14 Finished
=====
```

Figura 8: Captura de pantalla de los resultados de Gobuster.

3.5. Resultados

Los resultados obtenidos se encuentran detallados a continuación: Se ejecutó Gobuster para realizar un escaneo exhaustivo en busca de directorios y ficheros en la máquina **Pickle Rick**. A continuación se presentan los hallazgos identificados:

Cuadro 1: Resumen de hallazgos de Gobuster

Directorio	Estado
/assets	Status: 301
/denied.php	Status: 302
/index.html	Status: 200
/login.php	Status: 200
/portal.php	Status: 302
/robots.txt	Status: 200
/server-status	Status: 403

Los hallazgos obtenidos revelan una variedad de directorios y páginas con diferentes estados de respuesta del servidor. Estos descubrimientos proporcionan pistas potenciales para investigaciones posteriores y podrían ser puntos de entrada para la exploración y explotación de posibles vulnerabilidades en la máquina **Pickle Rick**.

3.5.1. Extracción de Información de Directorios

Durante el proceso de reconocimiento, se llevó a cabo un análisis exhaustivo de los directorios encontrados mediante el uso de diferentes herramientas de escaneo como Dirb, Gobuster y Nikto. Esta fase se centró en identificar y explorar posibles rutas y archivos disponibles en el sitio web objetivo para obtener información adicional sobre su estructura y contenido.

Dirb, al analizar la web, identificó una serie de directorios y archivos que incluían directorios permitidos y archivos predeterminados de Apache, lo cual proporcionó una visión inicial de la disposición de la página web.

Por su parte, Gobuster reveló una lista más detallada de directorios y archivos, mostrando rutas específicas y extensiones de archivos encontrados durante el escaneo, permitiendo una exploración más profunda de la estructura del sitio web.

Nikto, además de detectar vulnerabilidades, identificó rutas y directorios accesibles, destacando posibles áreas de interés y archivos permitidos, enriqueciendo así el conocimiento sobre la composición del sitio web objetivo.

El análisis conjunto de los resultados de estas herramientas proporcionó una visión más amplia de los recursos disponibles en el servidor, permitiendo un análisis más profundo y una mejor comprensión de la arquitectura y la posible exposición de información sensible.

3.6. Análisis de Directorios

Al explorar el directorio `/assets`, se identificaron los siguientes elementos:

- `bootstrap.min.css`: Archivo CSS, 119 KB.
- `bootstrap.min.js`: Archivo JavaScript, 37 KB.
- `fail.gif`: Imagen GIF, 49 KB.
- `jquery.min.js`: Archivo JavaScript, 85 KB.
- `picklerick.gif`: Imagen GIF, 222 KB.
- `portal.jpg`: Imagen JPEG, 50 KB.
- `rickandmarty.jpeg`: Imagen JPEG, 488 KB.

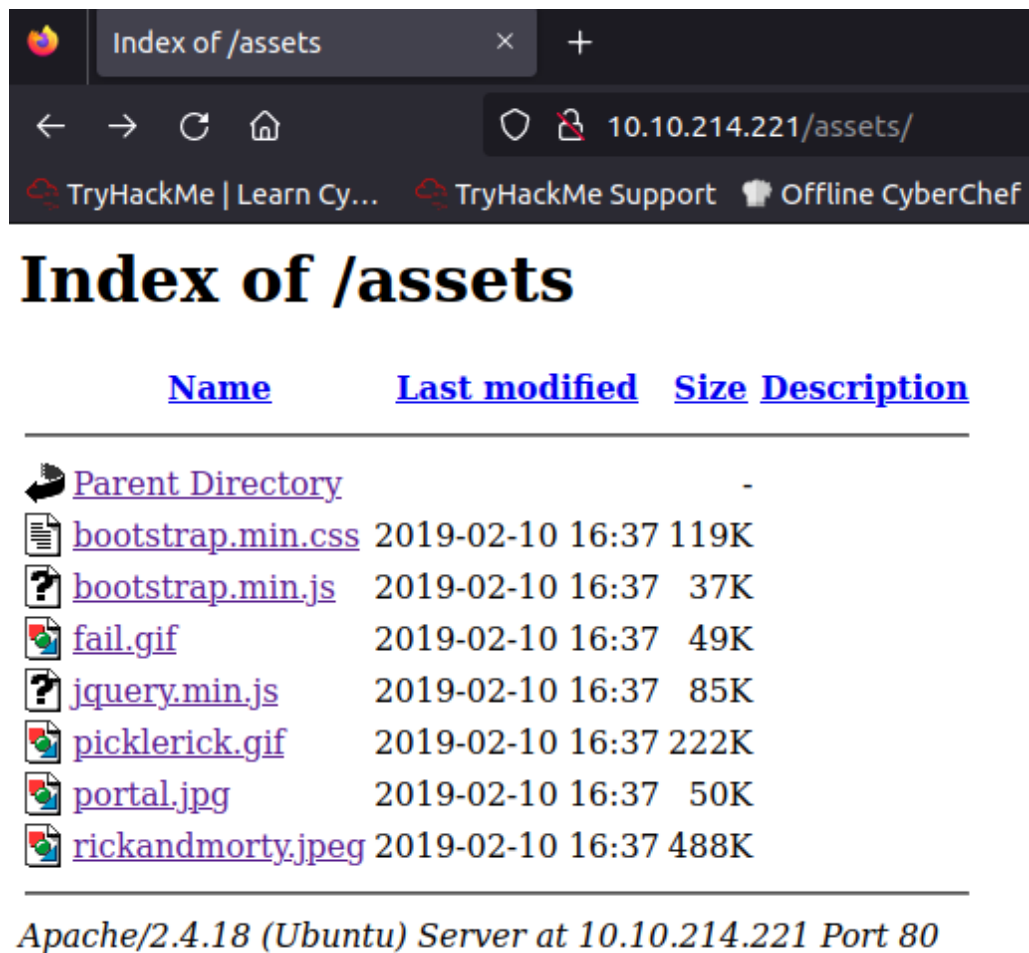


Figura 9: Captura de pantalla del directorio `/assets`

El directorio `/denied` redirecciona al login, donde se realizarán pruebas utilizando el usuario `R1ckRu13s` tanto para el acceso como para la conexión SSH, aunque parece restringido.

El acceso al directorio `/portal` también conduce al formulario de inicio de sesión de PHP.

En el archivo `/robots.txt`, se encontró la posible clave `Wubbalubbadubdub`. Por último, el acceso al directorio `/server-status` fue denegado.

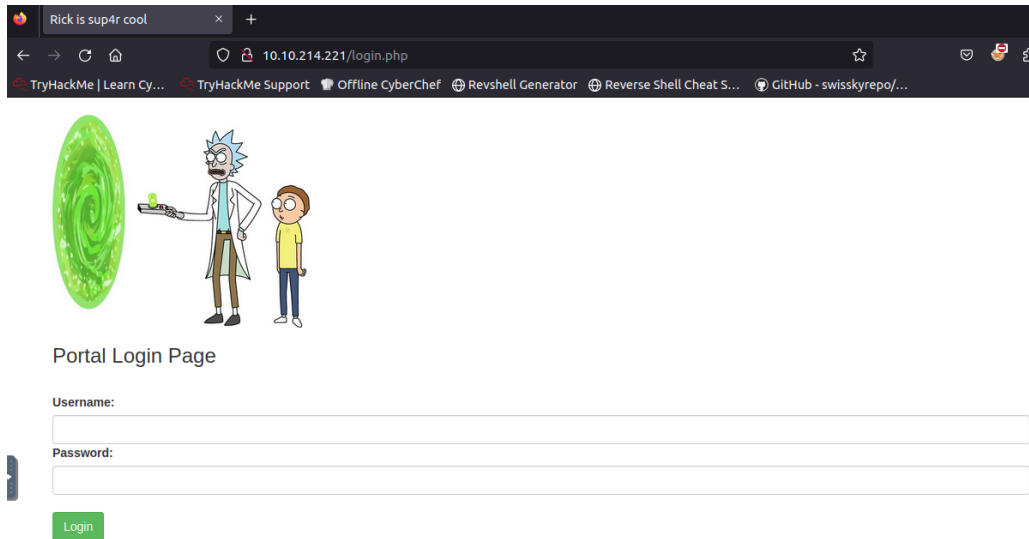


Figura 10: Captura de pantalla del acceso al portal

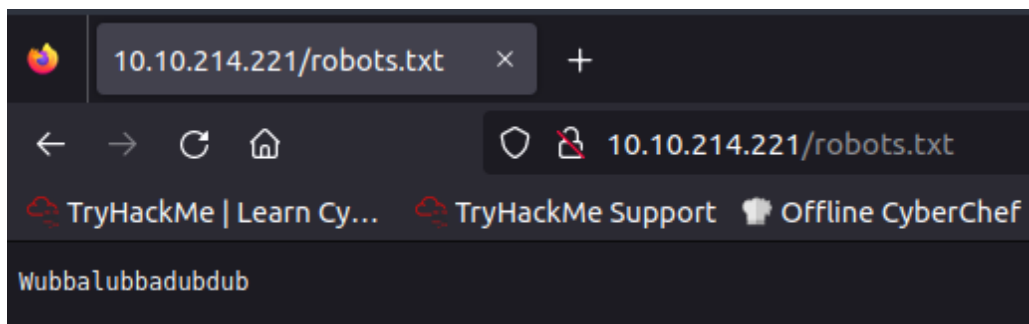


Figura 11: Captura de pantalla del directorio /portal

3.7. Resumen de Hallazgos

Se identificaron los siguientes elementos:

Directorio	Descripción
/assets	Contiene archivos y recursos variados.
/denied	Redirige al formulario de login.
/portal	Conduce al formulario de login PHP.
/robots.txt	Contiene la posible clave.
/server-status	Acceso denegado.

Cuadro 2: Resumen de Hallazgos en Directorios

Durante la exploración, se encontraron credenciales potencialmente sensibles que podrían comprometer la seguridad del sistema.

Usuario	Contraseña
R1ckRu13s	Wubbalubbadubdub

Cuadro 3: Credenciales encontradas

4. Explotación de vulnerabilidades

En esta etapa, se realizan pruebas controladas destinadas a obtener acceso no autorizado, recopilar información sensible o alcanzar los objetivos predefinidos dentro del escenario planteado.

4.1. Prueba: Acceso al Portal

Se llevó a cabo una prueba utilizando el usuario encontrado a través de la frase presente en el archivo `robots.txt`. Utilizando el identificador `Wubbalubbadubdub`, se accedió con éxito al portal de la página web.

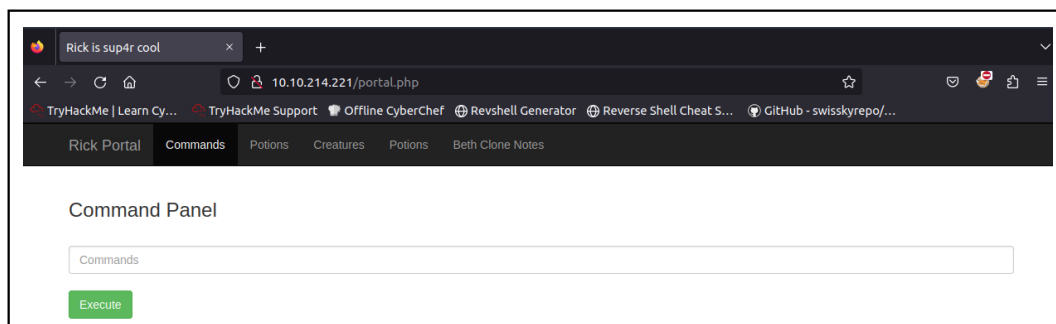


Figura 12: Captura de pantalla del acceso al portal

El panel de comandos es una interfaz web que permite ejecutar comandos en el sistema operativo de la máquina **Pickle Rick**, como si se usara una terminal o una consola. El panel de comandos se puede acceder desde la página web de la máquina.

Al ejecutar el comando `whoami` en el terminal del panel, se obtiene el resultado `www-data`. Este resultado indica el usuario con el que se está ejecutando el proceso del servidor web Apache. En sistemas basados en Unix o Linux, el usuario `www-data` se utiliza comúnmente como el usuario de ejecución para los servicios web. Este usuario está configurado para tener permisos limitados en el sistema, lo que mejora la seguridad al restringir el acceso a ciertos recursos.

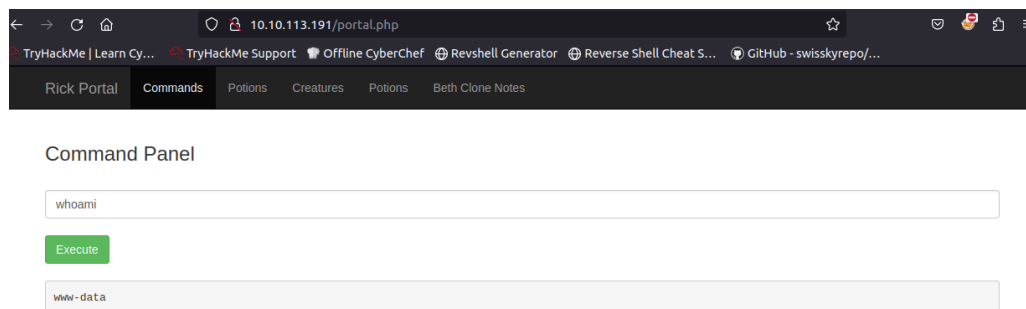


Figura 13: Captura de pantalla del comando `whoami`.

Por otro lado, al ejecutar el comando `pwd` en el terminal del panel, se muestra el directorio actual como resultado, que es `/var/www/html`. Este directorio es comúnmente utilizado como la ubicación principal para los archivos de un servidor web en sistemas basados en Unix o Linux. Contiene los archivos y recursos que son accesibles para el servidor web y se utiliza como el directorio raíz para servir contenido web.

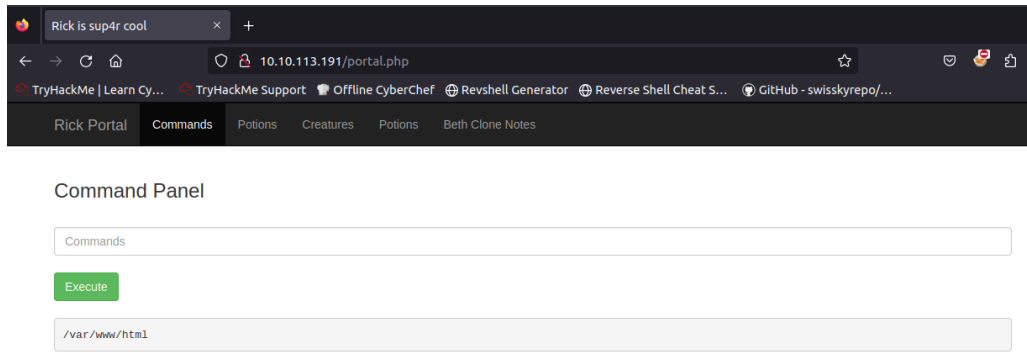


Figura 14: Captura de pantalla del comando pwd.

Al utilizar el comando ls en el terminal del panel, se listan los archivos y directorios del directorio actual. Esta acción permite visualizar el contenido del directorio en el cual se encuentra el usuario, mostrando los archivos disponibles para acceder o manipular.

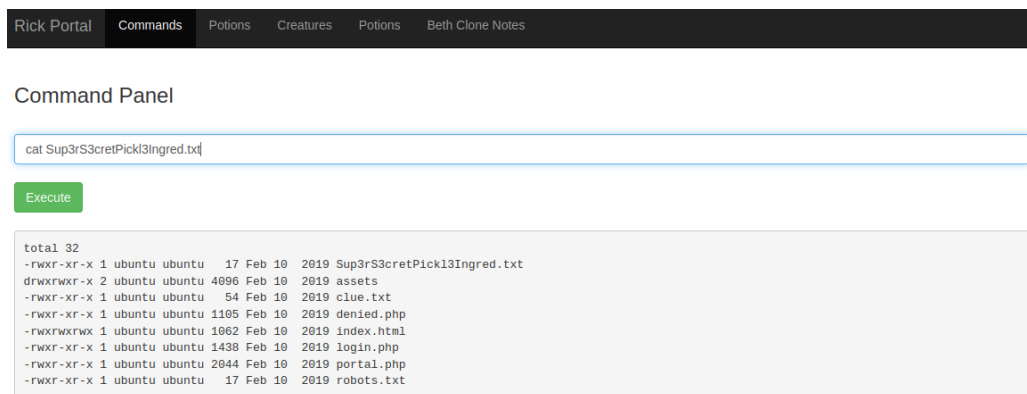


Figura 15: Captura de pantalla del comando ls.

Al intentar visualizar el contenido del archivo 'Sup3rS3cretPickl3Ingred.txt' utilizando el comando cat, se presenta un mensaje de error relacionado con permisos. Probablemente esto es debido a restricciones de permisos establecidas en el sistema objetivo.

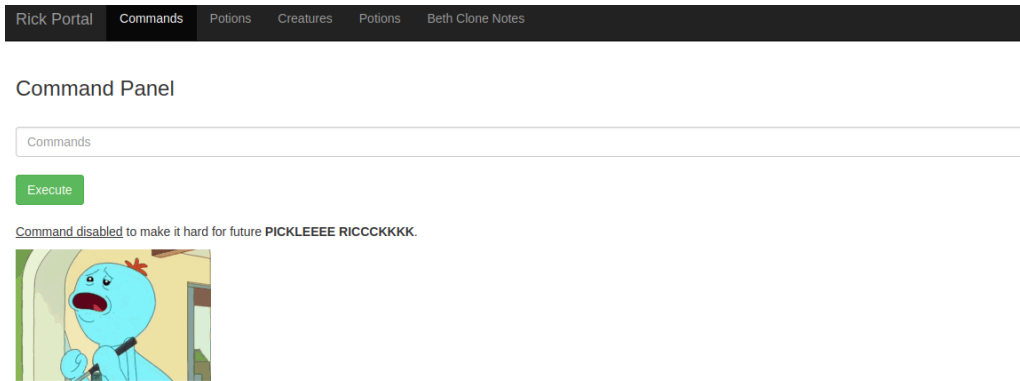


Figura 16: error al ejecutar cat.

Durante la inspección del código fuente de la página web en busca de pistas relevantes, se encontró un fragmento en base64 que despertó curiosidad. Al decodificarlo, reveló un mensaje aparentemente cifrado. La interpretación inicial de este mensaje sugirió la posibilidad de que se tratara de una pista relevante o un indicio para avanzar en el reto.

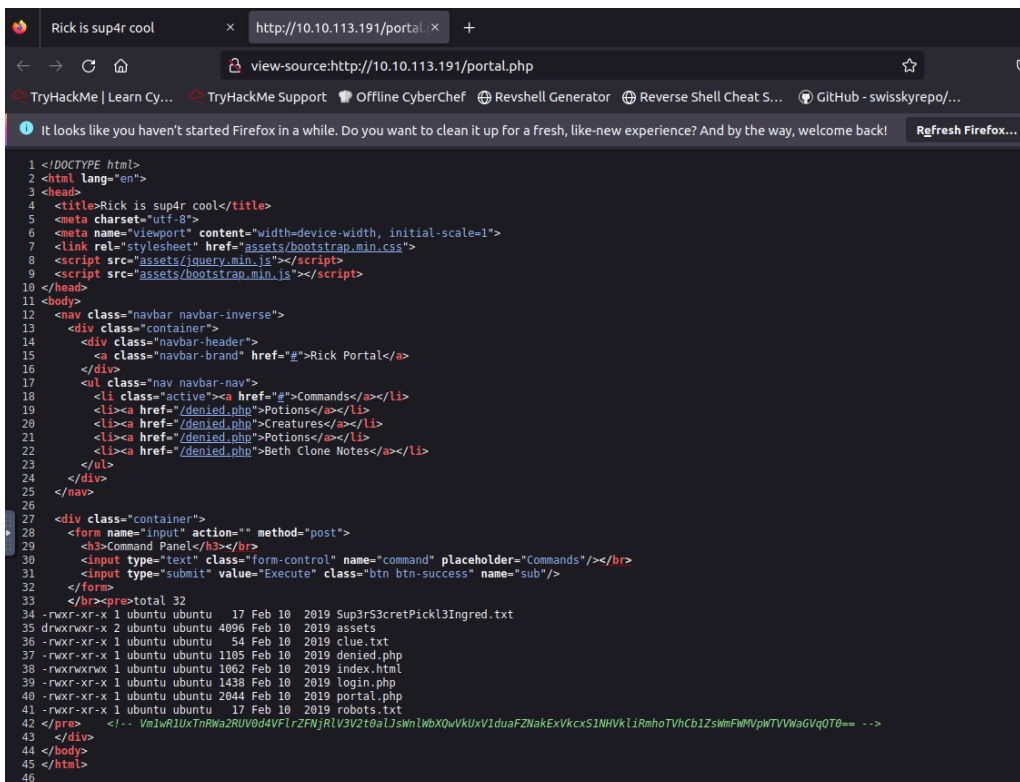


Figura 17: código portal web.

Esta revelación generó expectativas sobre la posible existencia de pistas adicionales o caminos no explorados anteriormente. Sin embargo, tras profundizar en el contenido decodificado, se descubrió que conducía a un camino sin salida, llevando la investigación hacia una distracción o un "rabbit hole".

Este descubrimiento resalta la importancia de analizar y evaluar detenidamente cada pista o indicio encontrado, así como mantener un enfoque crítico durante el proceso de resolución del desafío.

The "Base64 Decode Online" is a free decoder for decoding online Base64 to text or binary. In other words, it is a tool that converts Base64 to original data. This online decoder is as smart as it is simple. Its super encoding standard. Thanks to it, this converter allows you to "decrypt" some Base64 strings, even while other online or offline decoders are powerless and cannot decode them, because they support only the "mal process, check [Base64 encode](#).

Base64*

cmF1bnRlZSvbgU+

Base64 Standard

Auto detection (works like a charm, however sometimes may fail for short strings)

Strict Decoding

No (ignore invalid characters and force decoding value as Base64).

Character Encoding

Auto detection (an experimental feature that may fail for "exotic" encodings)

Decode Base64

Text

rebit hole

The result of Base64 decoding will appear here

Figura 18: caída en un rabbit hole.

Al explorar opciones adicionales fuera de la consola de comandos, se presentó una peculiaridad: al intentar acceder a ciertas áreas o funcionalidades, se desplegaba una alerta en forma de gif protagonizado por la versión Pepinillo de Rick. Este gif incluía un mensaje enfático que expresaba que solo "el real" tenía autorización para acceder a esa sección.

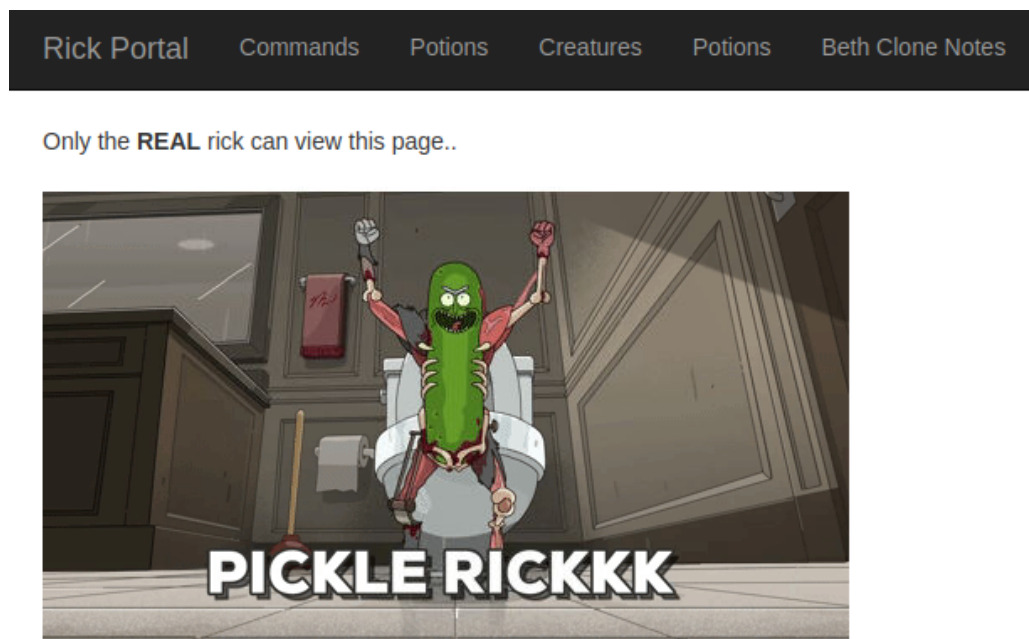


Figura 19: Pickle Rick nos reta.

Este inesperado comportamiento generó intriga sobre la posible relevancia de esta restricción. Sin embargo, tras varios intentos y exploración adicional, se confirmó que esta restricción era una medida implementada en el desafío, que no ofrecía pistas útiles para avanzar en el mismo.

Asimismo, se observó que el contenido de 'clue.txt' no contenía ninguno de los ingredientes necesarios para ayudar a Rick. Tal y como se aprecia en la figura 20 de la página 17.

Command Panel

Commands

Execute

Look around the file system for the other ingredient.

Figura 20: El fichero no contiene ningún ingrediente.

4.2. Primer ingrediente

Una vez dentro de la máquina, busqué los ingredientes que Rick necesita para hacer su poción. El primer ingrediente estaba en el mismo directorio donde me encontraba, así que lo abrí desde el navegador con la siguiente URL:

`http://10.10.113.191/Sup3rS3cretPickl3Ingred.txt`

El contenido del archivo era el siguiente:

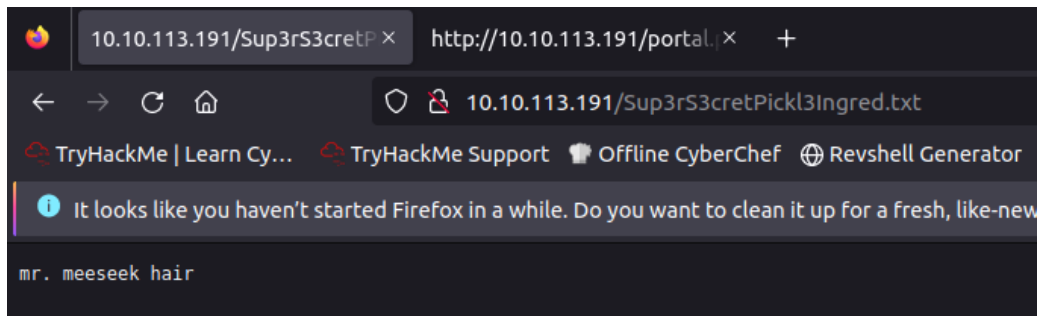


Figura 21: Primer ingrediente.

<i>Primer ingrediente</i>	
<i>Flag</i>	
<i>mr.meeseekhair</i>	

4.3. Segundo ingrediente

Después de obtener el primer ingrediente, seguí buscando los demás en la máquina. Para ello, listé el directorio raíz con el comando `ls /` y obtuve el siguiente resultado:

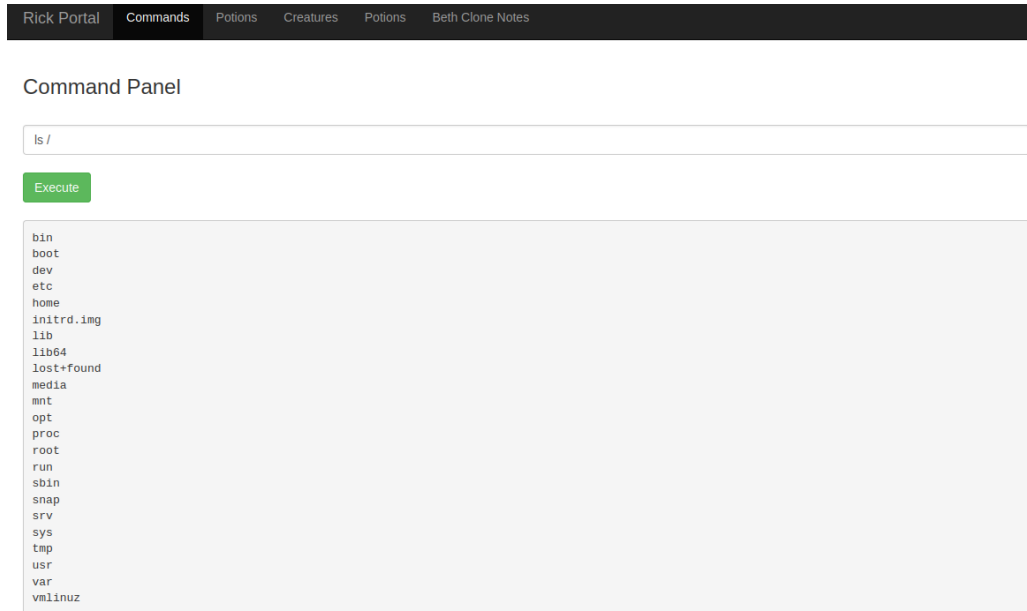


Figura 22: Listado del directorio raíz

Entre los directorios que aparecían, me llamó la atención el directorio `/home`, donde suelen estar los usuarios del sistema. Me dirigí a ese directorio y volví a listar con el comando `ls /home`, revelando los usuarios Rick y Ubuntu:

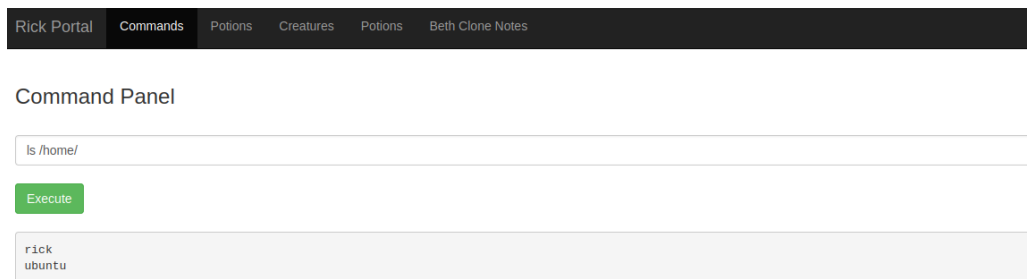


Figura 23: Listado del directorio home

Decidí entrar en el directorio de Rick, ya que era el nombre del dueño de la máquina, y listé su contenido con el comando `ls /home/rick`. Para mi sorpresa, me encontré con un archivo llamado **second ingredients**:

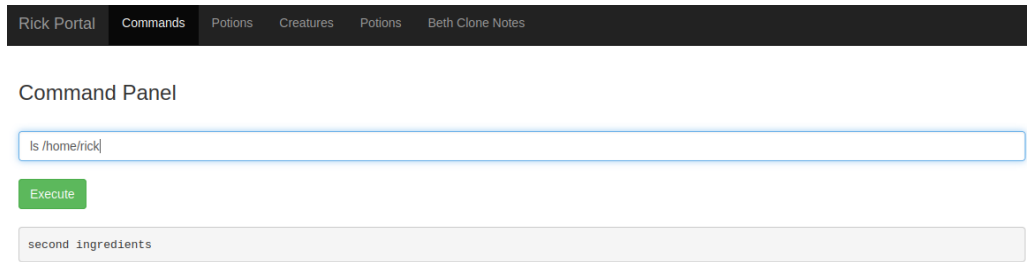


Figura 24: Listado del directorio de Rick

Este archivo parecía ser el segundo ingrediente que Rick necesitaba, pero no podía verlo en el navegador como el primero. Tampoco podía usar el comando `cat` para mostrar su contenido, ya que estaba restringido por el servidor web. Entonces, tuve que pensar cómo hacer para ver el contenido del archivo.

Lo primero que se me ocurrió fue investigar qué tipo de archivo era, para ver si podía usar alguna herramienta para abrirlo. Para ello, usé el comando `file second ingredients`, que analiza el contenido del archivo y muestra su tipo. El resultado fue el siguiente:

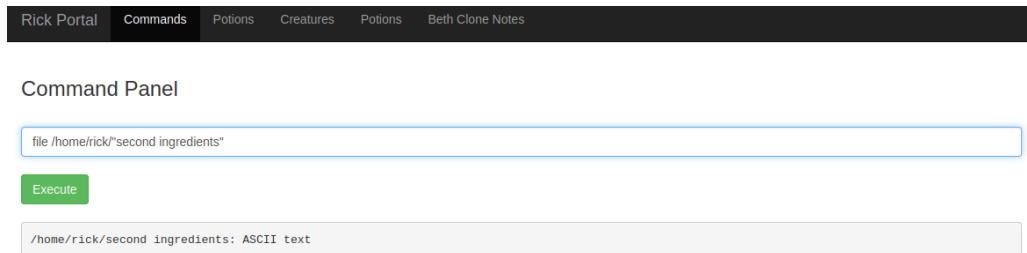


Figura 25: second ingredients: ASCII text

El archivo era un texto ASCII, que es un formato de texto simple. Esto me confirmó que podía usar algún editor o visor de texto para ver el archivo.

Intenté copiar el archivo a mi directorio actual, que era `/var/www/html`, donde sí tenía permiso de lectura y escritura. Para ello, usé el comando `cp /home/rick/"second ingredients" /var/www/html/segundo.txt`. Se debe destacar que, si no se entrecomilla el archivo, se producirá un error, ya que Linux lo interpretará como dos archivos distintos: `second` y `list`, cuando en realidad solo es uno. Por lo tanto, entrecomillar un archivo es una forma de evitar que Linux interprete los espacios como separadores de argumentos.

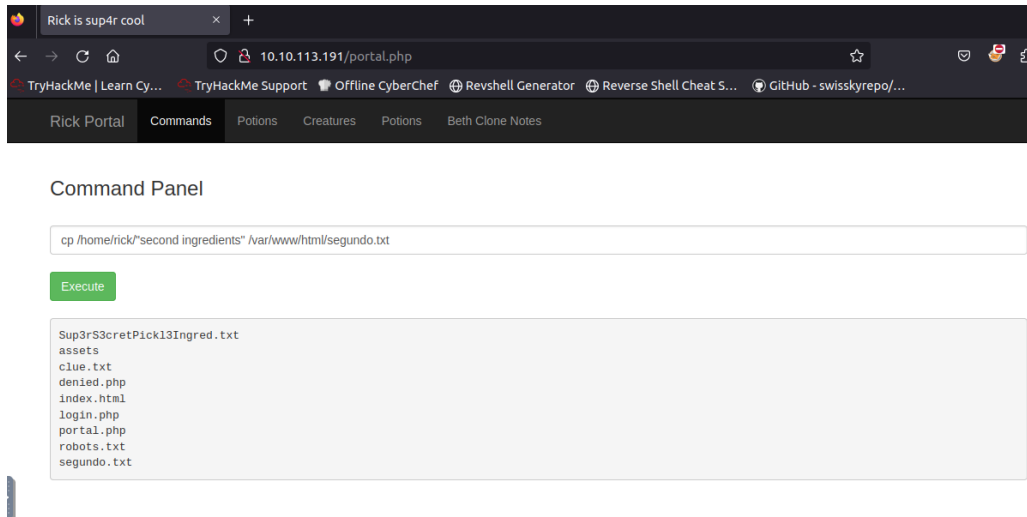


Figura 26: Copia del segundo ingrediente

Finalmente, opté por utilizar el comando `less second ingredients`. El comando `less` es un visualizador de archivos de texto en Unix y Linux. A diferencia de otros comandos como `cat`, `less` permite visualizar el contenido de un archivo en páginas, lo que facilita la lectura y exploración de archivos largos. Asimismo, posibilita desplazarse hacia adelante y hacia atrás en el texto, buscar términos específicos y navegar de manera eficiente a través del contenido. El archivo contenía el texto `1 jerry tear`, que era el segundo ingrediente que Rick necesitaba.

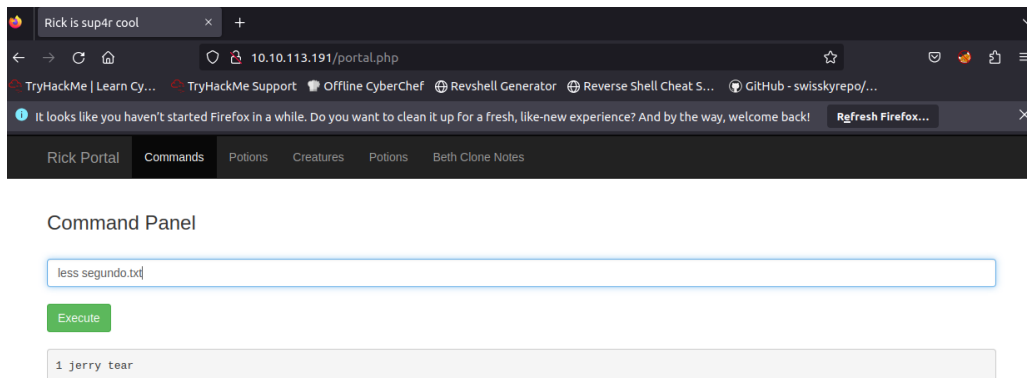


Figura 27: Flag segundo ingrediente

Segundoingrediente	
Flag	
1jerrytear	

4.4. Tercer ingrediente

Tras obtener el segundo ingrediente, me dispuse a buscar el tercero y último. Para ello, intenté probar si podía ejecutar permisos de sudo, que es un comando que permite ejecutar otros comandos como si fuera el usuario root. Para ver qué comandos podía ejecutar con sudo, usé el comando `sudo -l`, que muestra la lista de comandos permitidos.

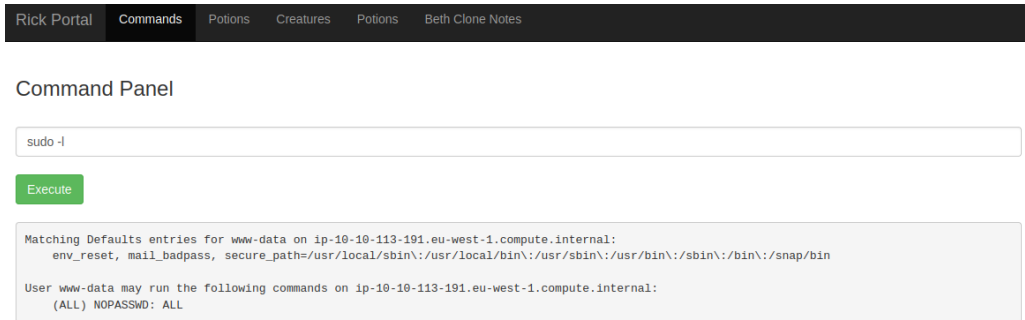


Figura 28: Resultado sudo -l

Esto significaba que podía ejecutar cualquier comando como root sin necesidad de contraseña, lo cual es una pésima práctica de seguridad, ya que facilita el acceso no autorizado al sistema. Así que aproveché esta oportunidad para escalar privilegios y acceder al directorio de root, donde suelen estar los archivos más importantes.

A continuación, listé el contenido del directorio de root con el comando `ls /root`, y encontré un archivo llamado `3rd.txt`:

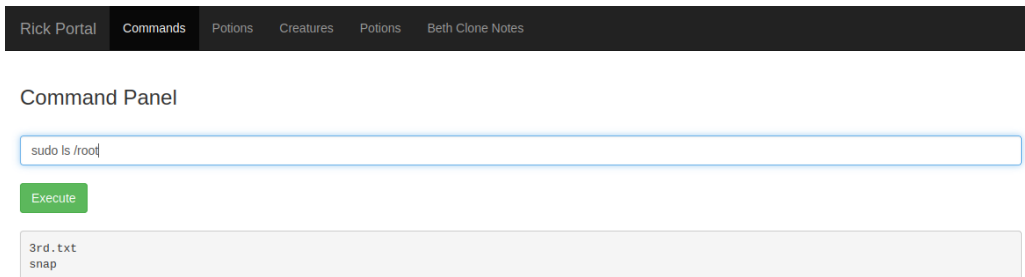


Figura 29: Listado del directorio de root

Este archivo parecía ser el tercer ingrediente que Rick necesitaba, así que lo abrí con el comando `less 3rd.txt`, que me mostró el siguiente contenido:

```

Tercer ingrediente _____
Flag
flee juice
  
```

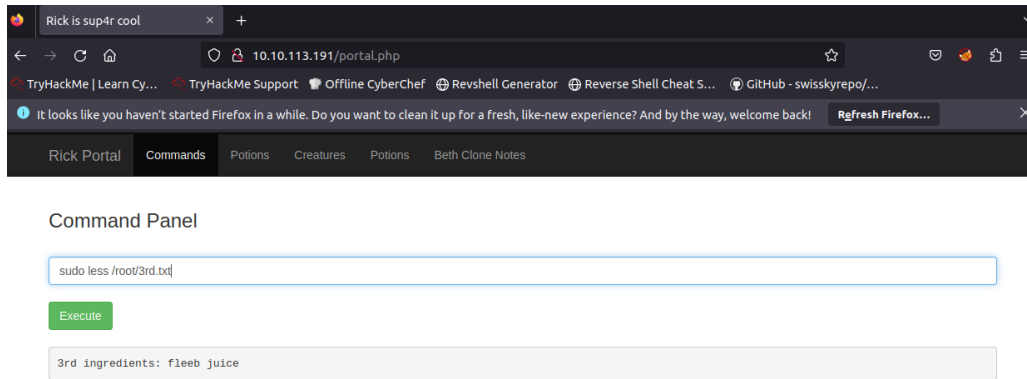


Figura 30: Resultado tercer ingrediente

Además de esta forma de obtener el tercer ingrediente, hay otras formas más complejas que implican el uso de una shell inversa (reverse shell), que es una técnica que permite establecer una conexión desde la máquina víctima a la máquina atacante, y así poder ejecutar comandos remotamente.

Este era el tercer y último ingrediente que Rick necesitaba. Con esto podemos concluir el writeup de la máquina rick pickle, que ha sido un reto interesante y divertido.

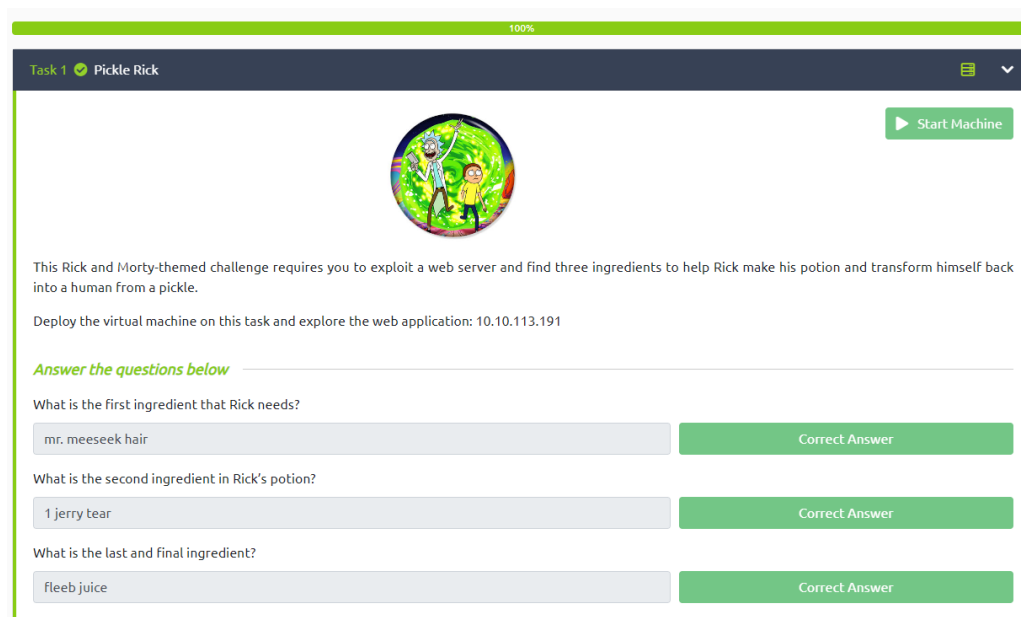


Figura 31: ¡Máquina Completa!