

PROJECT 6 CS460

TAN THO LE

SQL Injection

I choose to do SQL injection for this project. SQL injection is the placement of malicious codes into SQL statement so that an attacker can gain access to data.

The tools I use is the DB Browser for SQLite.

I created a sample database named **sample.db** that has two columns, username and password.

For a user to gain access to the information, they have to provide correct username and password for this basic query:

```
SELECT * FROM `users` WHERE username = 'username' AND password = 'password'
```

The query above returns information from table **users** if the *username* and *password* provided the user match those in the table.

It seems logically correct, but this query is prone to a very basic SQL injection technique.

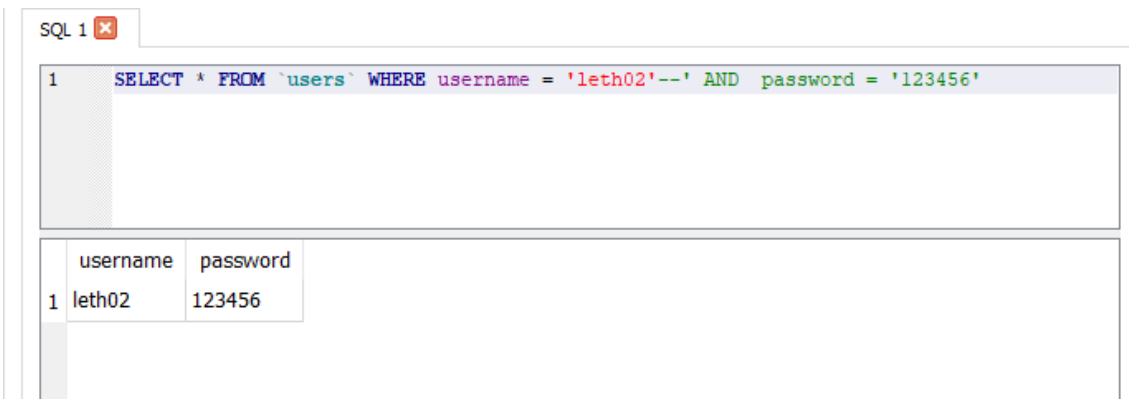
If an attacker insert '-- after a username, then the query will turn into:

```
SELECT * FROM `users` WHERE username = 'leth02'--' AND password = 'blabla'
```

When running this query, the part of the query starting from **AND** is commented out, which means that whatever the password is, a user can still get access to data if they add '-- after the username.

So, both of the above query and the following query return the record of user **leth02**, no matter what the password is for the SQL injection statement

```
SELECT * FROM `users` WHERE username = 'leth02'--' AND password = '123456'
```



SQL 1

```
1 SELECT * FROM `users` WHERE username = 'leth02'--' AND password = '123456'
```

	username	password
1	leth02	123456

One solution to prevent this attack is to not directly compare/use the information provided by user on the form of the client-side to execute SQL query.