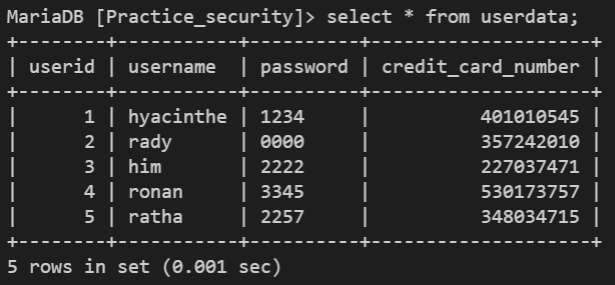
# C5-S2 – SQL INJECTION

# Create table userdata

Create database *Practice\_security*

Create table *userdata* in database *Practice\_security* and insert data from file *userdata.sql*



# Understand the project

* Run the project with PHP built-in server

php -S localhost:8080

* Try bad values for username and password:
  + What is the value of the variable *$query*?

Answer: We use $query to select data from table in database. So, value of the variable $query are data from database.

* + Which data is returned in *$list\_of\_users*?

Answer: Data that returned is empty array.

* + What error message do you see on the browser?

Answer: Wrong username or password!

* Try with correct values for username and password (for example username = “hyacinthe” and password = “1234”).
  + What is the value of the variable *$query*?

Answer: We use $query to select data from table in database. So, value of the variable $query are data from database

* + Which data is returned in *$list\_of\_users*?

Answer: Data that returned is a list of array associative.

* + Check that you can see on the browser the credit card number of this user (and this user only)

Answer: Yes, we can see the credit card number of this user.

# Perform SQL injection



Now let’s say that you are a hacker trying to steal credit card numbers from all users, without knowing their password.

To do this you can use well-chosen values for username and password:

Input these values in the form:

* username = “rady”
* password = “ ' OR True; ” (do not forget the first single quote in password value !)
* What is the value of the variable *$query*?

Answer: We use $query to select data from table in database. So, value of the variable $query are data from database.

* Which data is returned in *$list\_of\_users*?

Answer: The data is an array associative.

* Can you see the credit card number of all users?

Answer: Yes, I can.

* Why is this kind of attack called “SQL injection”?

Answer: Because this attack we use SQL statement to inject.

Now try with password = “ ' OR True; DELETE FROM userdata; ”

* What is the value of the variable *$query*? Why is SQL injection a very serious security issue?

Answer: Value of the variable $query is get data from table in database. SQL injection a very serious security issue because it can affect to your database that cause to lose your data.

# Free Images : antivirus, security, privacy, secured, safe, software, icon, clipart, os, yellow, green, laptop, computer, technology, communication, notebook pc, screen, equipment, electronics, keyboard, portable, connection, text, product, font, line ...Make the website safe!

## Use Prepared statements

You should never concatenate strings to build your query but use prepared statements instead!

After explanation from teacher, change the PHP code to use prepared statements

## Write cleaner code

You must always use prepared statements but if your code is clean, it is unlikely that an attacker can use SQL injection against your website

"SELECT username, credit\_card\_number FROM userdata WHERE username='$username' and password='$password';"

This query works but it is not a good way to check the password:

You should:

1. Get the password from database of the user whose username is *$username*
2. Check if the password is correct in PHP code, not in SQL query

Change the PHP code to use this better way to check the password

$list\_of\_users = $statement->fetchAll();

foreach ($list\_of\_users as $user) :

You expect only one record as result, so you should:

1. Fetch only one record, not All
2. Don’t loop, but display the only record

Change the PHP code to use this better way to display the results