Binghamton University, Watson School of Engineering

Project 1:
SQL Injection

Name: Rahul Verma

Science of Cyber Security – CS 559-01

Prof. Guanhua Yan

October 03, 2022

Contents

SQL INJECTION

1. Intro to SQLI:

There are weaknesses in almost every web application and we make use of this weakness in SQL injection. In order to access the database, we use the code injection technique known as SQL injection and although, the user shouldn't have access to the database, this strategy allows them to do so. The most prominent and well-known attack used to alter already-existing data and the most common web hacking method used is SQL injection. By getting access and impersonating a specific user, one can also destroy the data for their own personal advantage. Additionally, the entire database can be accessed as the attacker gains admin privileges, which leaves it open to attack which the attacker can use to get a persistent backdoor in any organization's systems.

2. Vulnerability:

On servers that don't adequately validate their user input, SQL injection can be simply implemented. There are many different sorts of vulnerabilities present, which might allow the program to accept any data from an unreliable source. The input can be constructed in such a manner that it purposefully matches the SQL query that the web application needs to run in order to authenticate its users and gain access to their data. Consider the following query, which was created by concatenating the input string that the user is said to have entered: "SELECT * FROM users WHERE username = ' +username + ' AND password = ' +password+ ' ", would allow the user access if the password contained a single quoted character. The SQL command that may be created by altering the input in this example serves as the vulnerability, and the code used to send the SQL instructions to the program serves as the exploit.

3. Attack Surface:

The SQL Injection attack surface consists of the login page and database. In this case, SQL injection, which includes the execution of queries, aids in gaining unauthorized access to the user data. The login page contributes to the attack surface since it doesn't check user input. Additionally, database is used as the attack surface, because the attacker makes use of it as a means of access.

4. Attack Vector:

An attack vector is a way or a method for anyone to access the web server or the system. Here, SQL Injection itself is the attack vector, that uses malicious SQL code to manipulate backend databases and access data that was not meant to be shown. When the user enters username and password as (' or 1=1 --), then the SQL query becomes "SELECT * FROM userpass WHERE user = '$user' AND pass ='' or 1=1 –". The attacker gets access to the database since the logic statement "1=1" is always true.

5. How the attack occurs:

Bypassing the login procedure, the SQL Injection method is used to get access to the database. The functioning of a SQL Injection attack will be demonstrated using the example below. The user must input his login and password in this HTML website form in order to access his data. The user sees the message "Login Successful" if the username and password are accurate. In this example, we utilize HTTP POST but the request can be made using either the HTTP GET or HTTP POST protocols.

```
<form action="/cgi-bin/login" method=post>
    Username: <input type=text name=username>
    Password: <input type=password name=password>
    <input type=submit value=Login>
```

The web browser sends the entered username and password as a string to the web server once the user enters the necessary username and password in the form mentioned above and clicks the login button. All the user information required for login is on the web server. The string is included in the request's body as follows:

```
Username = enteredUser & password = enteredPassword
```

The program doesn't check the login procedure before granting the user immediate access without checking and using additional security measures like executing a SQL query.

```
SELECT * FROM Users WHERE (username = "enteredUser" and password =
"enteredPassword");
```

The user's profile that contains the supplied username and password will be found using this SQL query. If the program does not employ any input validation techniques, it is susceptible to a SQL injection attack.

```
--Unauthorized Access Attempt Input:
    password = ' or 1=1 --

--SQL statement will become:
    select count(*) from users where username = 'user' AND password = '' or 1=1 --

(This will always check if the password is empty OR if 1=1 which will always
 be true and thus, give complete access to the database to the hacker.)
```

So, the fundamental concept is to provide a certain input that will modify the logic of SQL statements and therefore remove the password verification. The SQL query will treat the password field as commented out if it contains an empty string, skipping the password check and granting access of the database to the attacker.

6. Working of the attack:

SQL attack involves a query that serves as the attacker's entry point to the database. The SQL command grants the attacker access to the data which is susceptible. We do not need any extra tools for this kind of attack as the browser is sufficient for it. There are two possible attack scenarios: first, where the user must complete a php form before sending it to the server using the HTTP POST method and second, where the user may access the database by just entering their login and password.

7. Code Explanation:

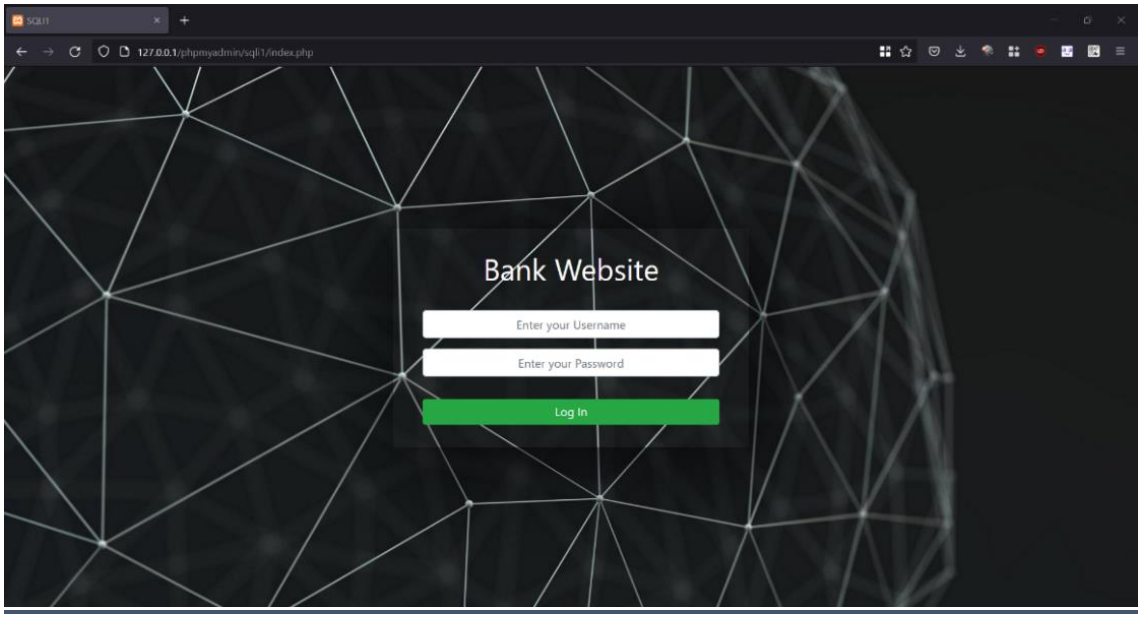
In "index.php" there are two links: one for bootstrap CSS and another for global CSS. These two files contain all of Bootstrap's stylesheets, so they're necessary in order to make the website look like an actual website instead of just some text on a white background There is a form container which consists of 3 input fields where user can enter their username and password and submit using the "Log In" button. Here, POST method is used to submit the data.

In "valid.php" when the entered username and password is correct, the user will be able to see all the Account Details, along with "Login Successful" message on top, which includes the balance in the account, but this page will be accessible to the hacker using SQL Injection. There is a "Log Out" button at the bottom which will redirect to "index.php"

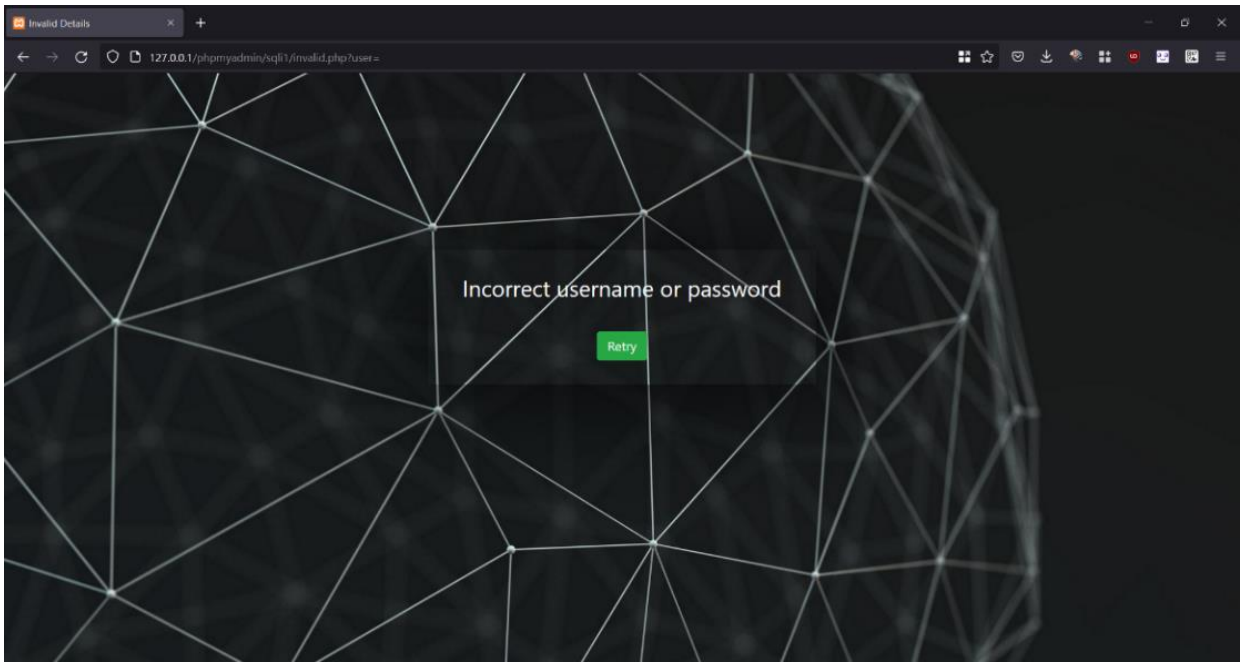
In “invalid.php” when the entered username and password is incorrect the page will show “Incorrect username or password” and a “Retry” button which redirects to “index.php” for the user to enter the correct username and password again.

The “login.php” file is used to check the inputs with the ones stored in the database. The created query has the following syntax: “SELECT * FROM userpass WHERE user=‘$user’ AND pass=‘$pass’ ”. The database is handled by the Xampp server. “user”, “pass” and “amount” are the respective columns in “userpass” table. The first row in the database is returned as the result when the user’s input is (1’ or ‘1’ = ‘1) on the website, which gives unauthorized access of user1’s data to the hacker, but we can see when entered in phpMyAdmin command line we get all the results including the first user. The row with the name user1 won’t be picked and the subsequent immediate user will be logged in if the input is of the type (1’ or ‘1’ = ‘1’ and user<>‘user1), we can see this in the phpMyAdmin screenshot where we are getting the output of all the users except the first user, we can modify this command such that it will return the third, fourth users and so on. This is the complete functioning of an SQL Injection Attack.

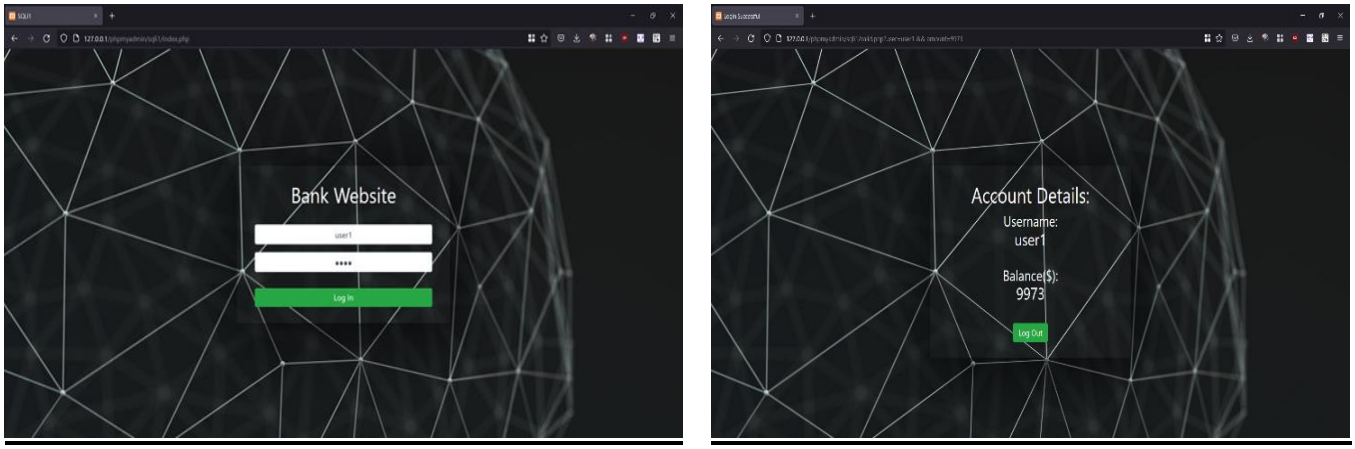
8. Screenshots:



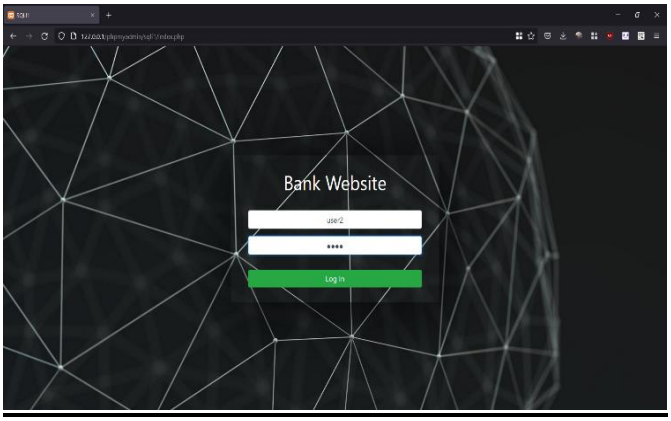
Login Page of the Website

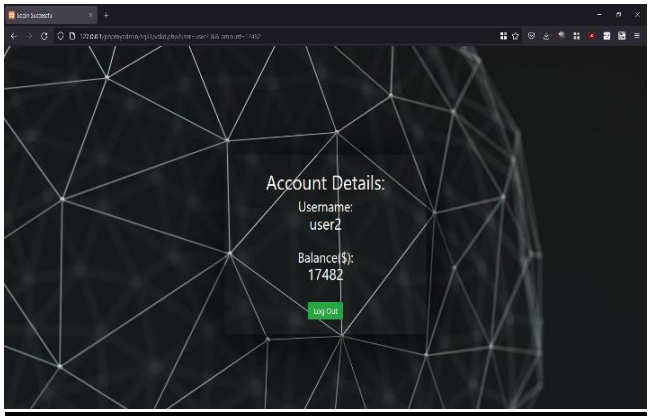


Incorrect username or password

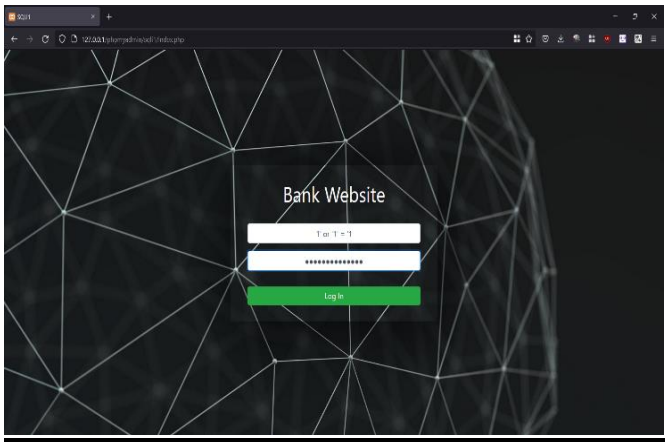


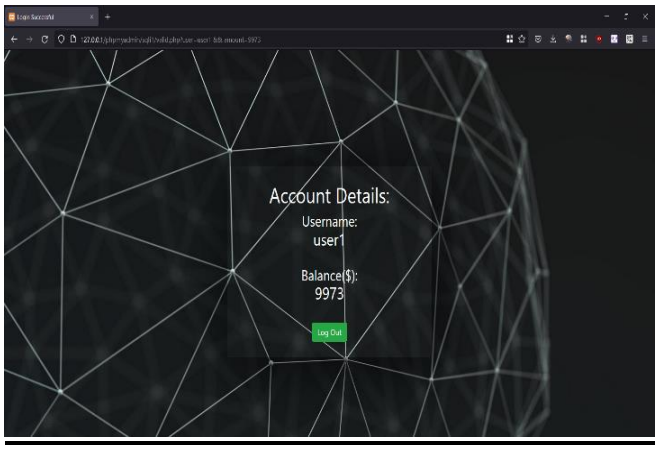
Accessing User1's account using correct username and password(username: user1 and password: 1111)



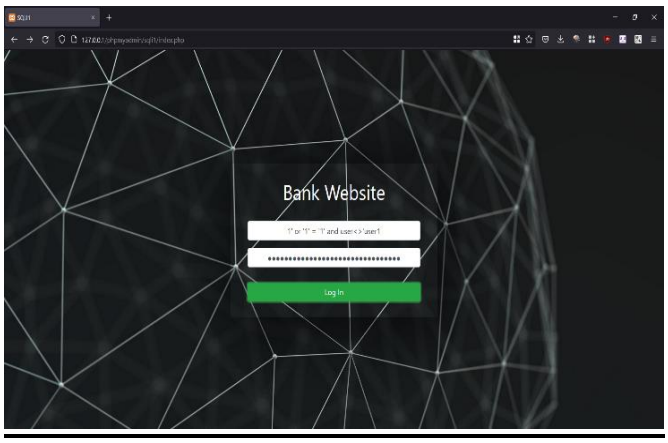


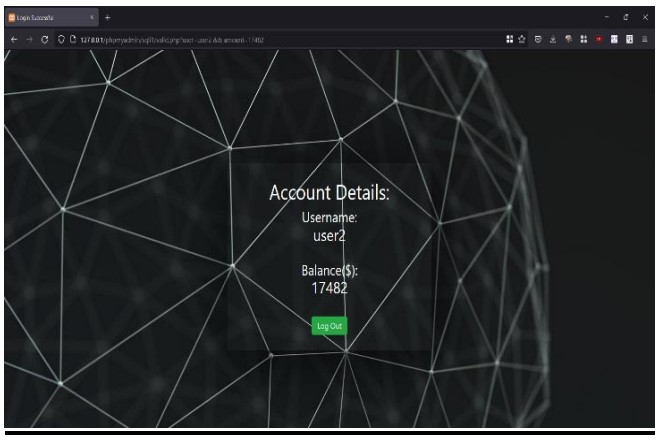
Accessing User2's account using correct username and password(username: user2 and password: 2222)



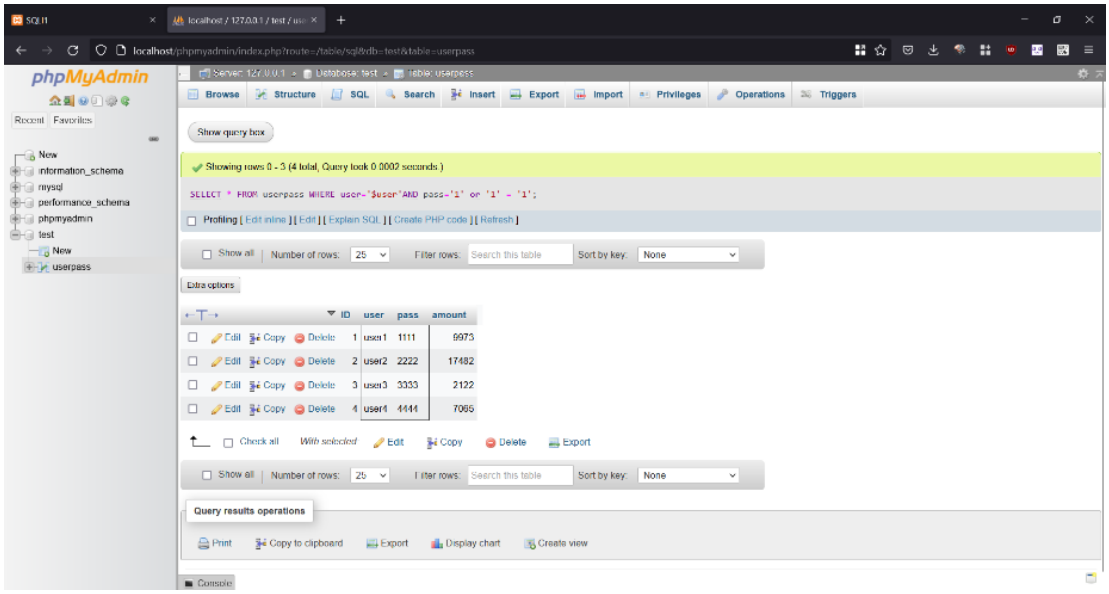


Unauthorized access to User1's account using SQL Injection(username and password: 1' or '1' = '1)

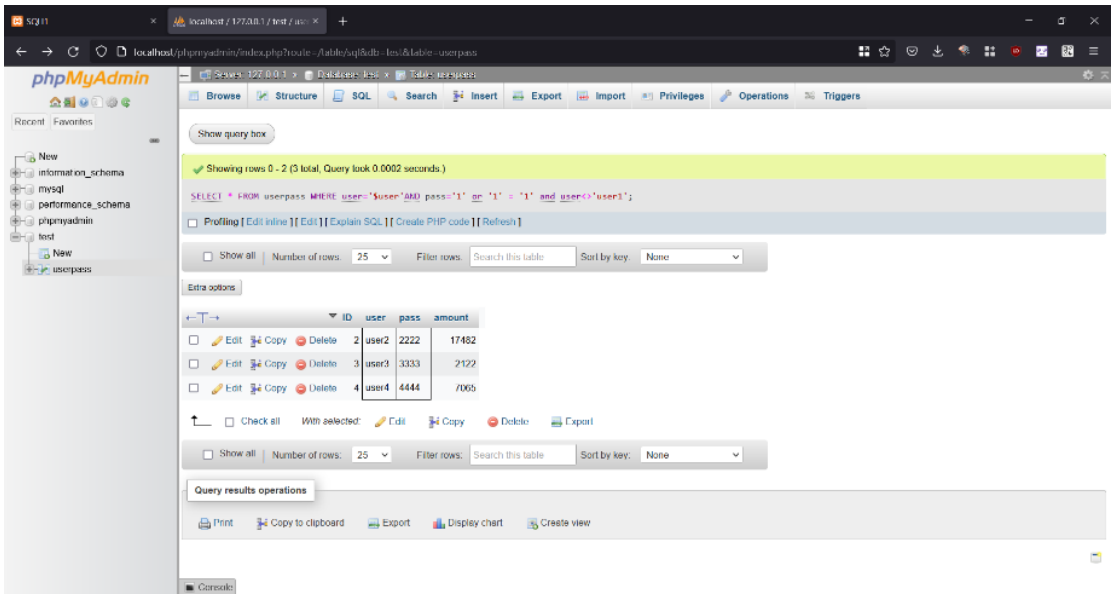




Unauthorized access to User2's account using SQL Injection(username and password: 1' or '1' = '1' and user<>'user1)



Using the SQLI query in phpMyAdmin returns all the users' data including user1



Using the SQLI query (<>user1) in phpMyAdmin returns all the users' data except user1

9. Bibliography:

- https://www.w3schools.com/sql/sql_injection.asp
- https://portswigger.net/web-security/sql-injection
- https://www.hacksplaining.com/exercises/sql-injection
- https://www.youtube.com/watch?v=ciNHn38EyRc
- https://www.youtube.com/watch?v=_jKylhJtPmI
- https://www.invicti.com/blog/web-security/sql-injection-cheat-sheet/
- https://en.wikipedia.org/wiki/SQL_injection
- http://testphp.vulnweb.com/ (sample SQLI vulnerable website)