**Project Description**

1. **SQL INJECTION ATTACK**

SQL Injection (SQLi) is a type of an [injection attack](https://www.acunetix.com/blog/articles/injection-attacks/) that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify, and delete records in the database.

1. Weak code and easy to make SQL Injection attack.

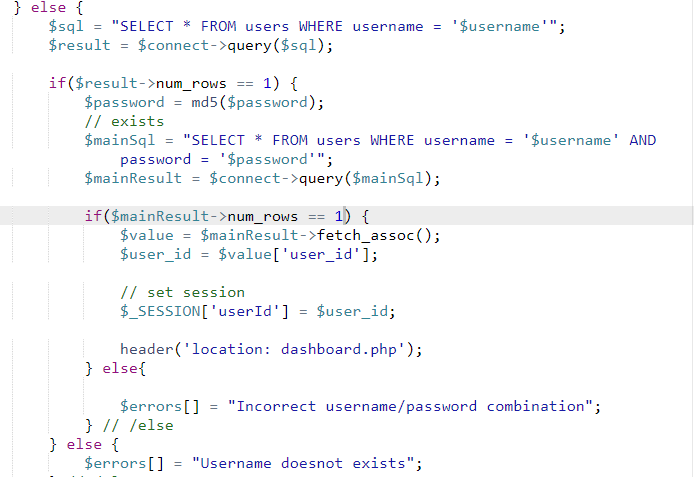
**Reason:**

This code is vulnerable because it is matching the username and password in the SQL query and it is very easy to inject any malicious SQL code to make the statement true for any condition.



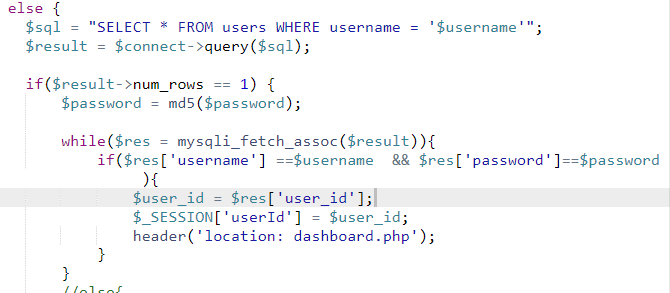
1. After making changes in the code but it is still vulnerable, if there is only one user exist in the users table.

**Reason:** This code is still vulnerable if there is only one user in the **users** table in the database. The reason behind this is that when the malicious SQL query runs if there is only one user the query fetches it and the condition becomes true and the unauthorized user is logged in successfully.



1. Now after making certain changes and doing the validations in the code and we finally made the code strong and not vulnerable for the SQL Injection.

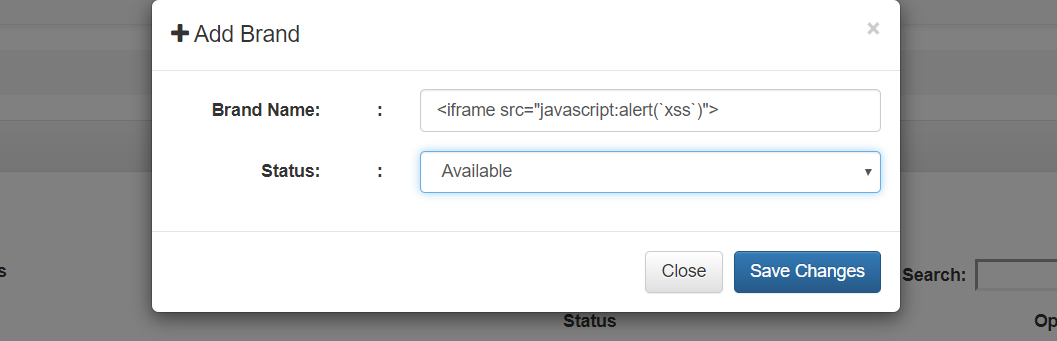
**Reason:** In this code the username is matched through the SQL query but after that username and password is again matched through the programming if condition in which any malicious SQL won’t work at all.



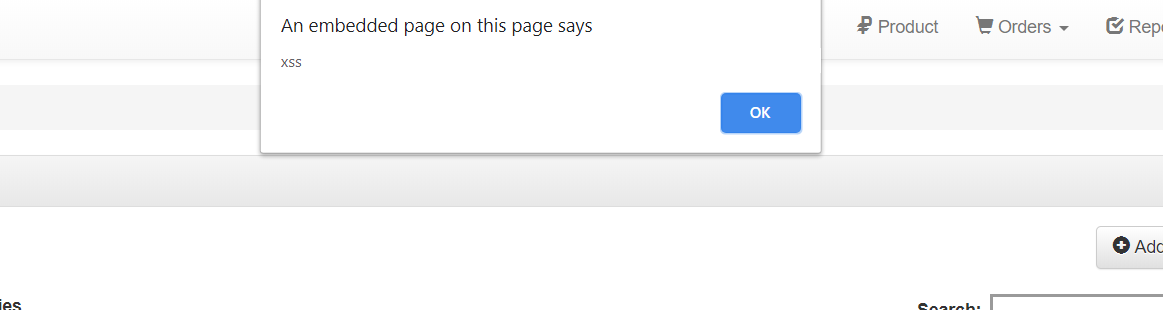
1. **Cross Site Scripting**

Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.

1. Injecting malicious JavaScript in the input fields.



When any user visits this page the malicious JavaScript code runs and it can steal any data of the user.



**Prevention of cross site scripting is to not allow any special character in any input fields. In php it is prevented by using a method htmlspecialchars().**

**Submitted By:**

**Cross Site Scripting Attack: 17SW20 Ali Sasoli**

**SQL Injection Attack: 17SW16 Haseeb Ansari**

**Front End Developer: 17SW42 Ahsan Methlo**

**Backend Developer : 17SW34 Waqas**

**File Upload Attack: 17SW38 Israr Ahmed**