**PortSwigger SQL injection lab**

**Intern id:** 195

**Lab :** SQL Injection Login Bypass

**Environment :** Linux

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective**

Demonstrate that the login form is vulnerable to SQL Injection, allowing an attacker to bypass authentication without valid credentials.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Target**

Application: PortSwigger SQLi Lab – Login page

Vector: Username / password input fields

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Vulnerability Description**

The login function fails to properly sanitize user input before building the SQL query. This enables attackers to manipulate the query logic and gain access without knowing valid credentials.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Steps to Reproduce**

1. Navigate to the login page.

2. In the username field, enter:

administrator’--’ OR 1=1--

3. In the password field, enter anything (e.g., test).

4. Submit the form.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Technical Explanation**

The backend likely uses a query similar to:

SELECT \* FROM users WHERE username = '<input>' AND password = '<input>';

When injecting:

username = ' OR 1=1--

password = test

The resulting query becomes:

SELECT \* FROM users WHERE username = administrator’--’OR 1=1--' AND password = 'test';

adminstrator’--’ OR 1=1 always evaluates to true.

-- comments out the rest of the query (including the password check).

The query returns the first user in the database (often administrator).

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Impact**

Authentication bypass achieved.

Attacker gains access as another user (possibly admin).

Further exploitation may expose sensitive data.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mitigation**

Use parameterized queries / prepared statements.

Employ input validation and proper escaping.

Implement least-privilege accounts for DB connections.

**\_\_\_\_\_\_\_\_\_\_\_ THANK YOU \_\_\_\_\_\_\_\_\_\_\_**