Web Application Penetration Testing Report

File Name: Web_Pentest_Report.pdf
Target: PortSwigger Academy Lab — SQLi Auth Bypass
Analyst: Sachin Sree D
Tools Used: Burp Suite
Date: 21-June-2025

1. Executive Summary

This report documents the findings from a security assessment conducted on a simulated web application from PortSwigger Web Security Academy. During testing, a critical vulnerability — SQL Injection in the login functionality — was identified and exploited to bypass authentication using a crafted SQL injection payload (administrator'--).

2. Scope of Testing

- **Target URL:** PortSwigger SQLi Lab (https://portswigger.net/web-security/sql-injection/lab-login-bypass)
- **Functionality Tested:** User login form (/login)
- **Tested For:**

1. SQL Injection (SQLi)
2. Authentication bypass
3. Basic error-based responses

3. Tools Used

✅ Burp Suite (Community Edition) — to intercept and modify login requests

✅ Browser (Chrome/Firefox) — for lab interaction

4. Methodology

1. Reconnaissance
   - Identified login form
   - Captured request using Burp Suite

2. Payload Injection
   - Modified username field in intercepted request
   - Tested `' OR 1=1--` and observed server response

3. Authentication Bypass Confirmed
   - App accepted input and responded with a 302 redirect (successful login)
   - Access granted without valid credentials

5. Vulnerability Details

🔴 SQL Injection in Login (Auth Bypass)

- Vulnerability: SQL Injection
- Endpoint: POST /login
- Parameter Affected: username
- Payload Used: administrator'--

- Proof of Concept:
  - Username: administrator'--
  - Password: anything
  - Result: logged in as administrator, bypassed authentication

- Impact:
  - Authentication completely bypassed
  - Full unauthorized access to the user dashboard

- Severity: Critical
- CWE ID: CWE-89

6. Recommendations

- ✅ Use parameterized queries (Prepared Statements)
- ✅ Sanitize and validate all user inputs
- ✅ Avoid dynamic SQL execution using string concatenation
- ✅ Implement proper error handling (no SQL error leakage)
- ✅ Enforce authentication logging and rate-limiting

7. Conclusion

The application is critically vulnerable to SQL injection in its login logic. This allows attackers to gain unauthorized access without valid credentials. Proper input sanitization and secure coding practices are essential to protect against this class of attack.

8. References

- OWASP SQL Injection Guide
 https://owasp.org/www-community/attacks/SQL_Injection

- CWE-89: SQL Injection (MITRE)
 https://cwe.mitre.org/data/definitions/89.html

- PortSwigger SQLi Labs Index
 https://portswigger.net/web-security/sql-injection